IMPROVEMENT OF THE DEMAND FORECASTING METHODS FOR VEHICLE PARTS AT AN INTERNATIONAL AUTOMOTIVE COMPANY

JOHN ANDERSON TORRES MOSQUERA

Universidad del Valle
Engineering Faculty
Industrial Engineering Program
Buga – Valle del Cauca
2018

IMPROVEMENT OF THE DEMAND FORECASTING METHODS FOR VEHICLE PARTS AT AN INTERNATIONAL AUTOMOTIVE COMPANY

JOHN ANDERSON TORRES MOSQUERA

Undergraduate final work to obtain the degree in Industrial Engineering

THESIS ADVISOR
Dr. Prof. Eng. Carlos Julio Vidal Holguín

THESIS CO-ADVISOR
Dr. Prof. Edwin Loaiza Acuña

COMPANY'S SUPERVISOR
Dr. Alexander Kressner

Universidad del Valle
Engineering Faculty
Industrial Engineering Program
Buga – Valle del Cauca
2018

Grade

JURY 1

JURY 2

Guadalajara de Buga, 2018

ACKNOWLEDGEMENT

I would first like to thank my thesis advisors Dr. Prof. Eng. Carlos Julio Vidal from the Industrial Engineering Faculty and Dr. Prof. Edwin Loaiza from the Science Faculty at the Universidad del Valle in Colombia. The doors to Prof. Vidal and Prof. Loaiza were always open whenever I ran into a trouble spot or had a question about my research or writing. They consistently allowed this paper to be my own work but steered me in the right direction whenever they thought I needed it.

I would also like to thank Dr. Alexander Kressner, my company’s thesis advisor, who also guided me through the whole process. Dr. Kressner not only provided me with excellent explanations of the company’s forecasting process, but he also guided me through the software development process, since he is the company’s forecasting software developer. He also directed me patiently on finding the company’s potential improvements points. He was not only a technical but also a personal support.

I would also like to thank Julia Schmitz who was my Boss at the company, and together with Dr. Kressner, gave me the opportunity of working for one of the most important automotive companies in the world on this research project.

I would also like to acknowledge Prof. Juan Carlos Osorio and Prof. Maria del Pilar Rivera from the Industrial Engineering Faculty at Universidad del Valle, I am gratefully indebted for their very valuable support.

I would also like to acknowledge Prof. Dr. Eng. Diego Fernando Manotas, Prof. Dr. Eng. Pablo César Manyoma from the Industrial Engineering School at the Universidad del Valle, whose valuable support allowed to reach this goal.

I would like to thank to all Professors in my time as a Business Administrator student, specially, Prof. Dorian Posada and Prof. Edwin Largo who supported me on the decision of studying Industrial Engineering.

I would also like to acknowledge Prof. Diego Peña, who also supported me on my first international semester.

Finally, I would also like to thank all other professors and colleagues who contributed to me on my university process. Thanks to you all.

DEDICATION

This thesis is dedicated, first of all, to God who has been my spiritual strength and guide through the whole process of studying this undergraduate Industrial Engineering program.

Secondly, I dedicated this thesis to my parents Luz Mery Mosquera and Diego León Torres who have always been a constant source of support and encouragement during the challenges of my whole life. Their unconditional love and good examples have taught me to work hard for the things that I aspire to achieve.

I would also like to dedicate this work to my brother, Juan Diego Torres, who has been a motivation for me, seeing him growing and spending time with him in any matter cannot be compared to anything else. He makes me feel that I have the best friend in the world. Also, to my brother, Christian, a great person, who has also been an impulse for me.

I must also express my very profound gratitude to my girlfriend, Alejandra Morales, for providing me with unfailing love and support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis.

This dedication is also to Dr. Prof Edwin who has been both a professional and personal support since I started the undergraduate program on Industrial Engineering. His outstanding lectures and explanations made me fall in love with mathematics, and his personal support through the whole university life, guided me to the right professional way.

To my Aunt Ana Milena Mosquera who is like my second mother, her prayers and unconditional help are an invaluable support to me.

To my friend Andrés Felipe Cano, who is one of my best friends, and was also a learning partner and true friend, who supported me on any situation. I would also like to thank to his mother Maria Eugenia Larrahondo, who also loved me like Andres' brother, and whose support I profoundly appreciate.

I would also like to thank to all friends and people who contributed to me in my 5-year university time. This accomplishment would not have been possible without them. Thank you all.

Finally, I would like to dedicate my thesis to the memory of my grandfather, Rafael Antonio Mosquera, who recently passed away. He was also a great support and example of life, who taught me that the most important value in life is to be always willing to help others.

ABSTRACT

This study aims to improve the forecasting accuracy for the monthly material flows of an area forwarding based inbound logistics network for an international automotive company. Due to human errors, short-term changes in material requirements or data bases desynchronization the Material Requirement Planning (MRP) cannot be directly derived from the Master Production Schedule (MPS). Therefore, the inbound logistics flows are forecast.

The current research extends the forecasting methods' scope already applied by the company namely, Naïve, ARIMA, Neural Networks, Exponential Smoothing and Ensemble Forecast (an average of the first four methods) by allowing the implementation of three new algorithms: The Prophet Algorithm, the Vector Autoregressive (Multivariate Time Series) and Automated Simple Moving Average, and two new data cleaning methods: Automated Outlier Detection and Linear Interpolation. All the methods are structured in a software using the programming language R.

The results show that as of April 2018, 80.1% of all material flows have a Mean Absolute Percentage Error (MAPE) of less than or equal to 20%, in comparison with the 58.6% of all material flows which had the same behavior in the original software in February 2018. Furthermore, the three new algorithms represent now 29% of all forecasts.

All the analysis realized in this research were made with actual data from the company, and the upgraded software was approved by the logistics analysts to make all future material flow forecasts.

Key Words: Forecast, Inbound Logistics, Prophet, Vector Autoregressive, Simple Moving Average, Outlier Detection, R.

TABLE OF CONTENTS

TABLES

FIGURES

1. INTRODUCTION

There are many business factors influencing a company's performance, among these accurate forecasts have the greatest impact on an organization's ability to satisfy customers and manage resources cost effectively (Syntetos, Babai, Boylan, Kolassa, & Nikolopoulos, 2016). A forecast is not simply a projection of future business; it is a request for product and resources that ultimately impacts almost every business decision the company makes across sales, finance, production, management, logistics and marketing (Logility, 2016). An improvement in forecast accuracy, even just one percent, can have a ripple effect across the business including reducing inventory buffers, obsolete products, expedited shipments, distribution center space, and non-value added work (Logility, 2017).

Typically, a variety of forecasting methods are applicable to any particular type of supply chain scenario. Exponential Smoothing Method is a best-fit statistical technique used when demand is trended but does not vary by the time of the year. The Holt-Winters variant is used when demand exhibits seasonality. The Moving Average Methods is best for products whose demand histories have random variations, including no seasonality or trend, or fairly flat demand (Logility, 2016). ARIMA models are the most suitable for stationary time series (Montgomery, 2016) and Neural Networks Forecasting is able to understand underlying pattern in the time series data (Hyndman & Athanasopoulos, 2014). The best tip is to pick the most effective and flexible models, blend their best feature, and shift between them as needed to keep forecast accuracy at its pick. Thence, allowing a forecasting software to choose for the best forecasting methods over time is the best approach.

The current research considers an International Automotive Company that produces vehicles in more than 23 assembly plants around the world and has 32 logistic service providers. The company has currently more than 4000 suppliers worldwide, which deliver different vehicles parts, components and finished goods to their corresponding consolidation center in its forwarding area. To be precise, the company has an area forwarding based inbound logistics network. Currently, there are 471 material flows considered in the forecasting software.

This software is able to produce 4-month-ahead forecast in aggregate units in tons, which are then upgraded monthly. The company has decided to forecast their material flows instead of using its Bill of Materials because it is currently undergoing an upgrading and synchronization process of all the logistics data bases. Moreover, since not all plants have the same MRP software, the information quality is also a main issue. This process causes the material flow observations to have many extreme values or outliers, as well as, missing values. When outliers and missing values are incorrectly handled they can certainly reduce the forecast accuracy (Chen & Liu, 1993) (Lepot, Aubin, & Clemens, 2017) (Vidal Holguín, 2010).

This Software has been developed in R, which is a Statistical Software Development Environment. Five forecasting methods have been implemented, namely, Naïve, ARIMA, Neural Network, Exponential Smoothing and Ensemble Forecast. The last one refers to the average of the forecasts delivered by the first four methods (Claeskens, Magnus, Vasnev, & Wang, 2016). These methods are based on the package *Forecast* on R (Hyndman & Khandakar, 2008).

As stated before, forecasts accuracy is a main issue in a company's performance. A 1% improvement in forecast accuracy leads to 3,2% reduction in transportation costs (Logility, 2017). Therefore, improving the forecast accuracy is the main issue regarding the material flow forecasts. Methods like the Moving Average are likely to bring good results, since many of the monthly material flows show either random variation or any trend. There are other new algorithms like Prophet Algorithm (Taylor & Letham, 2017) that is based on structural time series models (Harvey & Peters, 1990) which considers features further than time series autocorrelations, and therefore is proved to outperform ARIMA models in some cases.

Consequently, the current research is aimed to prove whether there are other Forecasting Methods than the currently used by the Forecasting Software, which are able to improve the forecast accuracy of the material flows.

The research's methodology starts at making a diagnosis of the current forecasting problem, as well as the software's forecasting approach. Then new algorithms are chosen based on the input data features and on the potential of the algorithms themselves. Finally, simulations with sample data are run, in order to test how the new algorithms perform, regarding subsets as well as with the whole data available.

This study is done as a research project at the company and all the analyses will be carried out using actual and current data.

This document is structured as follows: a description of the problem is made in Chapter 2. Chapter 3 states the Research Question, then Chapter 4 states the main and specific objectives of the research. Chapter 5 makes a theoretical description of the current forecasting methods used by the company along with the new algorithms which are likely to cause an improvement in the forecast accuracy. Furthermore, a time series outliers detection method proposed by (Chen & Liu, 1993) is explained in detail. Chapter 6 outlines the company's forecasting problem to every detail, from the demand patterns to the current forecast analysis approach. Moreover, the software routines, as well as the input and output data, are explained using a flow chart. Then, a diagnosis of the possible improvements to the overall forecasting approach is stated. Afterwards, the results delivered by the new algorithms are described in Chapter 7. Finally, Chapter 8 presents the Conclusions, Chapter 9 the References, and a general view of the Final Software's Routines, Input and Output Data can be found at the end of the document.

2. PROBLEM DESCRIPTION

Supply Chain Forecasting (SCF) goes beyond the operational task of extrapolating demand requirements at one echelon. It involves complex issues such as supply chain coordination and sharing information between multiple stakeholders. (Syntetos, et. al, 2016)

The final customer's demand sets the entire supply chain into motion. That is to say, this demand is the input for the Production Planning (Master Production Schedule - MPS) then this MPS is desegregated to get the Material Requirement Planning (MRP); for which the Materials and resources needed to meet the Production Plan are supplied. This would be the common process flow for material planning but when it comes to international companies with many factories around the world and different MRP Information Systems, the information flows are in most cases not as accurate as they should be.

The following problem concerns an international automotive company which produces vehicles in more than 23 assembly plants around the world and has 32 logistics service providers. The company has currently more than 4000 suppliers worldwide which deliver different vehicles parts to their corresponding consolidation center in its forwarding area. In other words, the company has an area forwarding based inbound logistics network. The material transportation is carried out via truck. Modes of truck shipments are milk runs, direct truck load between supplier and the plant and area forwarding.

The Area Forwarding Based Inbound Logistics Network consists of three major participants. The first one is an organization which must be supplied with goods. It can be seen as a customer, mathematically spoken a sink or, in a logistics term, the unloader in the network. Secondly, a set of suppliers provides the goods required by the unloader. The suppliers are the source of the network and are put together in groups, most likely based on their geographical location. Such a group is called area, and that is where the term Area Forwarding comes from. The third participant in the network is the logistics service provider, who organizes the transportation of goods between the suppliers and the unloader. In different areas, different logistics service providers can be hired by the unloader. The logistics service provider runs a consolidation center within the area. The logistics service provider collects goods from each supplier in his area and consolidates them in his consolidation center. The logistics service provider’s action is limited to a pure cross docking, no warehousing takes place in the consolidation center. The transportation step from the supplier to the consolidation center is called pre-leg or first leg. From this point, the goods from different suppliers in the area are transported together. This step is called main leg. If the load in the pre-leg exceeds the volume of one vehicle, the goods are transported directly to the unloader. This transportation is called full load (Schöneberg, Koberstein, & Suhl, 2010). A typical area forwarding based inbound logistics network is shown in Fig. 1.

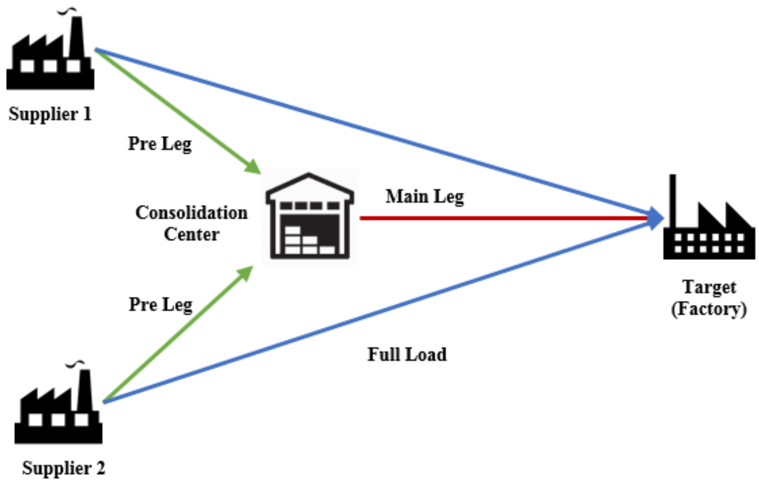


Figure 1. Area Forwarding based inbound logistics network

Currently there is a good information flow between the assembly plants and the suppliers, which allows the suppliers to plan their capacity and resources accordingly. However, the freight forwarders are not linked to these information flow yet, which results in imprecise transportation capacity planning. This causes additional transportation costs, which could be avoided. Added to that, the network dynamics is more and more complex.

The company has a main information system which allows to gather the material volume transported by all freight forwarders to the assembly plants in a big data base. However, due to human errors, short-term changes in material requirements or data bases desynchronization, the amount of material transported differs considerably from the actual material registered in the data bases. That is why, the company's first proposal was to develop mid-term forecasts to the material requirement flow for the main legs, with the following features:

- Forecast for the main legs.
- Forecast in ton/month.
- Future time frame of 3-6 months.
- Forecast for the Area Forwarding based inbound logistics Network in Europe.

This allowed to synchronize the Inbound Supply Chain capacities and reduce costs considerably. Four forecasting methods were implemented namely, Naïve, ARIMA, Neural Networks, and Exponential Smoothing, which are available in the Library *Forecast* for R (Hyndman & Khandakar, 2008). Moreover, a fifth method is used; this is a simple linear combination of the four forecasting methods with parameters set to 1; called Ensemble Forecasts, which delivers the average values predicted by the other models.

Due to the data quality problem already mentioned, there are flows that do not have more than 10 observations, which were not considered for the forecast. A simple outlier recognition method was implemented to clean the data before fitting the forecast models, namely values exceeding a threshold (5 times the Interquartile Range plus the 3rd quantile) were replaced by the median of the observations. After this, the data was split up into samples to make cross-validation, then the Mean Absolute Percentage Error (MAPE) was calculated for each validation test for a forecast of 4 periods in the future. After all these calculations, the mean for all the

MAPEs obtained in the cross-validation is calculated and taken as a decision variable to choose which of the four models better fits the observations, namely the lowest value. With the selected method then the forecast is carried out, which delivers a forecast report for 471 material flows in ton/month for the next 4 months.

This process is made for every main leg flows, i.e., the five methods are fitted to each time series, and then cross-validation tests are made with samples taken from each time series, these tests deliver a corresponding MAPE, which then is averaged with the other results. The decision variable to select the methods which fits the time series the best is the averaged MAPE from the cross-validation tests. After this process, the actual forecast with the best fitted method is performed.

Other important reports are also generated through this process, specifically the Forecast Errors Plots for every Material Flow forecasted, and the Monthly Material Volume Flow for each main leg.

Up to February 2018, the forecasts have the following MAPE performance, the frequency represents the number of main legs flows:

Table 1. *Forecast MAPE Distribution Feb. 2018*

MAPE Category	Frequency	Percentage
lower than 10%	61	13.0%
between 10% and 20%	215	45.6%
between 20% and 30%	106	22.5%
between 30% and 40%	37	7.9%
higher than 40%	52	11.0%
TOTAL	**471**	**100.0%**

The questions which naturally arise here are: (1) how the forecast performance measured by the MAPE can be improved? (2) Are there any other forecasting methods which could better perform as the current ones? (3) Is the outlier detection process good enough so that the current time series deliver reliable forecasts? (4) How can the forecasts variability be monitored, since the model that provides the best fit to historical data generally does not result in a forecasting method that produces the best forecast of new data (Montgomery, 2016)?

3. RESEARCH QUESTION

Which mathematical models are the best to improve the demand forecasting of vehicle parts at an international automotive company regarding their forecasting accuracy?

4. OBJECTIVES

4.1. Main Objective

To analyze and propose mathematical methods which can improve the demand forecast of vehicle parts at an international automotive company regarding their forecasting accuracy.

4.2. Specific Objectives

- To make a diagnosis of the current situation for the material flows' forecast at the company.
- To determine which mathematical models can improve the current forecast performance.
- To simulate forecasts with the selected models for sample data and select the ones with the best performance.

5. THEORETICAL FRAMEWORK

5.1 Forecasting Methods

In order to produce reliable forecasts, many mathematical methods have been developed. They can be classified as: subjective, time series, econometric, and other methods (Logility, 2016).

5.1.1 Subjective Methods

The Delphi method, The Market Research and the Historical Life-Cycle Analogy are examples of subjective methods. The **Delphi Method** is a structured communication method, originally developed as a systematic, interactive forecasting method which relies on a panel of experts whom are asked using questionnaires in two or three rounds; the final results are determined by the mean or median scores of the final round. On the other hand, the **Market Research** can also be used as a forecasting method, as it is defined as the process of assessing the viability of new goods and services through research guided directly with the consumer which allows the company to discover the target market and record opinions and other input from consumers concerning the interest in the product. Market Research can be supplemented by referring to the performance of a similar product or service, this is called the **Historical Life-Cycle Analogy**, using life cycle data from an ancestor of the product provides a starting point to focus the forecasting on the right way (Logility, 2016).

5.1.2 Time Series Methods

5.1.2.1 Definition

A time series is a collection of stochastic variables $x_1 \ldots x_t \ldots x_T$ indexed by an integer value *t*. The interpretation is that the series represent a vector of stochastic variables observed at equal-spaced time intervals. The series is also sometimes called a stochastic process. The distinguishing feature of time series is that of temporal dependence: the distribution of x_t conditional on previous values of the series depends on the outcome of those previous observations (Sorensen E., 2012).

A time series is called **stationary** (more precisely covariance stationary) if:

$$\begin{aligned} &E(x_t) = \mu \quad for\ t = 1,2,3 \ldots \\ &E[(x_t - \mu)^2] = \gamma(0) \\ &E[(x_t - \mu)(x_{t+k} - \mu)] = \gamma(k) = \gamma(-k) = E[(x_t - \mu)(x_{t-k} - \mu)]; k = 1,2 \ldots \end{aligned} \tag{1}$$

Where $\gamma(k); k = 0,1, \ldots$ are independent of *t* and finite.

There is a quite long tradition in time series to focus on only the first two moments of the process, rather than on the actual distribution of x_t. If the process is normally distributed, all information is contained in the first two moments and most of the statistical theory of time series estimators is asymptotic and more often than not only dependent on the first two moments of the process (Sorensen E., 2012). The $\gamma(k)'s$ for $k \neq 0$ are called **auto-covariances**, if these are divided by the variance then the **auto-correlations** are obtained, namely $\rho(k) = \gamma(k)/\gamma(0)$. These are the correlation of x_t with its own lagged value.

$\sum_T$ is the matrix of variances and covariances of $x_1, \ldots, x_t$ then:

$$\sum_{T} = \begin{pmatrix} \gamma(0) & \gamma(1) & \gamma(2) & \dots & \gamma(T-1) \\ \gamma(1) & \gamma(0) & \gamma(1) & \dots & \gamma(T-2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \gamma(T-2) & \dots & \dots & \dots & \gamma(1) \\ \gamma(T-1) & \gamma(T-2) & \dots & \gamma(1) & \gamma(0) \end{pmatrix} \quad (2)$$

Then let $\boldsymbol{\Omega_T}$ be the matrix of autocorrelations, i.e. $\sum_{\boldsymbol{T}} = \gamma(0)\boldsymbol{\Omega_T}$. It is also important to mention that, a stationary process e_t with mean 0 is called *white noise* if $\gamma(k) = 0 \; for \; k \neq 0$. Of course, this implies that the autocorrelation matrix is just an identity matrix, so that the standard OLS (Ordinary Least-Squares) assumptions on the error term can also be formulated as “the error term is assumed to be white noise”.

The Autoregressive (AR) model, moving average (MA) model, Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA) models, exponential smoothing and Multivariate Time Series (MTS) are examples of time series methods.

5.1.2.2 AR(p) Model

An **AR(*p*)** model (Autoregressive of order p) is a discrete time linear equation with noise, of the form:

$$X_t = \alpha_1 X_{t-1} + \cdots + \alpha_p X_{t-p} + \varepsilon_t \quad (3)$$

Here p is the order, $\alpha_1 \dots \alpha_p$ are the parameters or coefficients (real numbers), ε_t is an error term, usually a white noise with intensity σ^2. The model is considered either on integers $t \in \mathbb{Z}$, thus without initial conditions, or on the non-negative integers $t \in \mathbb{N}$. In this case, the relation above starts from $t = p$ and some initial condition $X_0 \dots X_{p-1}$ must be specified. The simplest case is an AR(1) model which can be written as:

$$X_t = \alpha_1 X_{t-1} + \varepsilon_t \quad (4)$$

With $|\alpha| < 1$ and $Var[X_t] < \sigma^2/(1-\alpha^2)$, it is a wide sense stationary process. Stationary means, in statistical terms, that the probability distribution of an arbitrary collection of X_t is time invariant (Tsay, 2014).

In order to model more general situations, it may be convenient to introduce models with non-zero average, namely of the form:

$$(X_t - \mu) = \alpha_1 (X_{t-1} - \mu) + \cdots + \alpha_p (X_{t-p} - \mu) + \varepsilon_t \quad (5)$$

When $= 0$, if the initial condition is taken as having zero average (in order to have a stationary series), then $E[X_t] = 0$ for all t. This situation can be avoided by taking $\neq 0$. The new process $Z_t = x_t - \mu$ has zero average and satisfies the usual equation:

$$Z_t = \alpha_1 Z_{t-1} + \cdots + \alpha_p Z_{t-p} + \varepsilon_t \quad (6)$$

But X_t satisfies:

$$X_t = \alpha_1 X_{t-1} + \cdots + \alpha_p X_{t-p} + \varepsilon_t + (\mu - \alpha_1\mu - \cdots - \alpha_p\mu) \tag{7}$$

$$X_t = \alpha_1 X_{t-1} + \cdots + \alpha_p X_{t-p} + \varepsilon_t + \tilde{\mu}$$

The **time lag operator** is widely used in the time series literature, before going any further is important to remember its use, because future equations are going to be simplified by using it.

Let S be the space of all sequences $(x_t)_{t\in\mathbb{Z}}$ of real numbers. Let define an operator $L: S \to S$, a map which transform sequences in sequences. It is defined as:

$$Lx_t = x_{t-1}, \quad for\ all\ t \in \mathbb{Z} \tag{8}$$

The equation should be written as $(Lx)_t = x_{t-1}$, with the meaning that, given a sequence $x = (x_t)_{t\in\mathbb{Z}} \in S$, a new sequence is introduced $Lx \in S$, that at time t is equal to the original sequence at time $t-1$, hence the notation $(Lx)_t = x_{t-1}$, but it is clear that L operates on the full sequence x, not on the single value x_t.

The map L is called time lag operator, or backward shift, because the result of L is a shift, a translation, of the sequence.

The time lag operator is a linear operator. The powers, positive and negative, of the lag operator are denoted by L^k :

$$L^k x_t = x_{t-k}, \quad for\ t \in \mathbb{Z} \tag{9}$$

Or for $t \geq \max(k, 0)$ for sequences $(x_t)_{t\in\mathbb{Z}}$. With this notation the AR model can be written as:

$$\left(1 - \sum_{k=1}^{p} \alpha_k L^k\right) X_t = \varepsilon_t \tag{10}$$

5.1.2.3 MA(q) Model

A **MA(*q*)** (Moving Average with order *q*) model is an explicit formula for X_t, in terms of noise of the form:

$$X_t = \varepsilon_t + \beta_1 \varepsilon_{t-1} + \cdots + \beta_q \varepsilon_{t-q} \tag{11}$$

The process is given by a weighted average of the noise, but not an average from time zero to the present time *t;* instead, an average moving with *t* is taken, using only the last *q* +1 times.

Using the time lags this can be expressed as:

$$X_t = \left(1 + \sum_{k=1}^{q} \beta_k L^k\right) \varepsilon_t \tag{12}$$

5.1.2.4 ARMA(p,q) Model

An **ARMA (*p, q*)** (Autoregressive Moving Average with orders *p* and *q*) model is a discrete time linear equation which combines the AR and MA models, its form is:

$$\left(1-\sum_{k=1}^{p}\alpha_k L^k\right)X_t=\left(1+\sum_{k=1}^{q}\beta_k L^k\right)\varepsilon_t \tag{13}$$

Or explicitly

$$X_t = \alpha_1 X_{t-1}+\cdots+\alpha_p X_{t-p}+\varepsilon_t+\beta_1\varepsilon_{t-1}+\cdots+\beta_q\varepsilon_{t-q} \tag{14}$$

A non-zero average model can also be incorporated. If it is desired that X_t has an average μ, the natural procedure is to have a zero-average solution Z_t like the generalization with the AR(*p*) model, as following:

$$Z_t = \alpha_1 Z_{t-1}+\cdots+\alpha_p Z_{t-p}+\varepsilon_t+\beta_1\varepsilon_{t-1}+\cdots+\beta_q\varepsilon_{t-q} \tag{15}$$

And take $X_t = Z_t+\mu$, hence solution of:

$$X_t = \alpha_1 X_{t-1}+\cdots+\alpha_p X_{t-p}+\varepsilon_t+\beta_1\varepsilon_{t-1}+\cdots+\beta_q\varepsilon_{t-q}+\tilde{\mu} \tag{16}$$

Another important feature to consider is the difference operator. The **first difference operator,** Δ is defined as:

$$\Delta X_t = X_t - X_{t-1} = (1-L)X_t \tag{17}$$

Let be

$$Y_t = (1-L)X_t \tag{18}$$

Then X_t can be reconstruct from Y_t by integration:

$$X_t = Y_t + X_{t-1} = Y_t + Y_{t-1} + X_{t-2} = Y_t + \cdots + Y_1 + X_0 \tag{19}$$

Having the initial condition X_0.

The second difference operator, Δ^2**,** is defined as:

$$\Delta^2 X_t = (1-L)^2 X_t \tag{20}$$

Assuming

$$Y_t = (1-L)Z_t \tag{21}$$
$$Z_t = (1-L)X_t \tag{22}$$

So Z_t has to be first reconstructed from Y_t :

$$Z_t = Y_t + \cdots + Y_1 + Z_1 \tag{23}$$

Where

$$Z_1 = (1 - L)X_1 = X_1 - X_0 \tag{24}$$

Then X_t can be reconstructed from Z_t:

$$X_t = Z_t + \cdots + Z_1 + X_0 \tag{25}$$

Finally, this can be generalized to $\Delta^{\boldsymbol{d}}$, for any positive integer *d*.

5.1.2.5 ARIMA(p,q,d) Model

An **ARIMA(*p,d,q*)** (Autoregressive Integrated Moving Average with orders *p,d,q*) model is a discrete time linear equation with noise, of the form:

$$\left(1 - \sum_{k=1}^{p} \alpha_k L^k\right)(1 - L)^d X_t = \left(1 + \sum_{k=1}^{q} \beta_k L^k\right)\varepsilon_t \tag{26}$$

It is a particular case of ARMA models, but with a special structure. Set $Y_t \coloneqq (1 - L)^d X_t$. Then Y_t is an ARMA(*p*, *q*) model

$$\left(1 - \sum_{k=1}^{p} \alpha_k L^k\right)Y_t = \left(1 + \sum_{k=1}^{q} \beta_k L^k\right)\varepsilon_t \tag{27}$$

And X_t is obtained from Y_t by successive integrations. The number d is thus the order of integration. For many time series, it can be seen that, different snapshots taken in time do exhibit similar behavior except for the mean level of the process. Similarly, processes may show nonstationary in the slope as well. Therefore, a time series X_t, can be called homogeneous nonstationary if it is not stationary but its first difference, that is $Y_t = X_t - X_{t-1} = (1 - L)X_t$, or higher-order differences, $Y_t = (1 - L)^d X_t$, produce a stationary time series. Consequently, a X_t time series can be called an autoregressive integrated moving average (ARIMA) process of orders *p*, *d*, and q -that is, ARIMA(*p*,*d*,*q*)- if its *d*th difference, denoted by $Y_t = (1 - L)^d X_t$, produces a stationary ARMA(*p*,*q*) process (Montgomery, 2016).

The random walk with drift 0 or Naïve method is an ARIMA(0,1,0). A nonzero average can be incorporated in the auxiliary process Y_t and consider the equation:

$$\left(1 - \sum_{k=1}^{p} \alpha_k L^k\right)(1 - L)^d X_t = \left(1 + \sum_{k=1}^{q} \beta_k L^k\right)\varepsilon_t + \tilde{\mu} \tag{28}$$

$$X_{t+h} = X_{t-1} + \varepsilon_t, \qquad h = 1, 2, 3 \ldots$$

With

$$\tilde{\mu} = \mu - \alpha_1 \mu - \cdots - \alpha_p \mu$$

Hence, the **Naïve Method** assumes that the most current observation is the only important one and all previous observations provide no information for the future. This can be thought of a weighted average where all the weight is given to the last observation (Hyndman & Athanasopoulos, 2014).

There is another way of writing the AR, MA, ARMA and ARIMA models, namely, using **lag polynomials**. Let $\phi_0 = 1,\ \theta_0 = 1,\ \alpha_0 = 1$ and define the lag polynomials:

$$\phi(L) = 1 - \phi_1 L - \phi_2 L^2 - \cdots - \phi_p L^p \quad (29)$$
$$\theta(L) = 1 + \theta_1 L + \theta_2 L^2 + \cdots + \theta_q L^q \quad (30)$$
$$\alpha(L) = (1 - L)^d \quad (31)$$

Therefore, the processes can be written in a more compact way as:

$$AR\colon\ \phi(L)X_t = \varepsilon_t \quad (32)$$
$$MA\colon\ X_t = \theta(L)\varepsilon_t \quad (33)$$
$$ARMA\colon\ \phi(L)X_t = \theta(L)\varepsilon_t \quad (34)$$
$$ARIMA\colon \phi(L)\alpha(L)X_t = \theta(L)\varepsilon_t \quad (35)$$

5.1.2.6 Seasonal ARIMA(p,q,d)(P,D,Q) Model

Seasonal ARIMA. The seasonal ARIMA[1] model incorporates both non-seasonal and seasonal factors in a multiplicative model. One shorthand of the model is:

$$ARIMA(p,d,q)(P,D,Q)_s$$

$$p = non\ seasonal\ AR\ order$$
$$d = non\ seasonal\ differencing$$
$$q = non\ seasonal\ MA\ order$$
$$P = seasonal\ AR\ order$$
$$D = seasonal\ differencing$$
$$p = seasonal\ MA\ order$$

Without differencing operations, the model could be written more formally as:

$$\Phi(L^S)\phi(L)(1 - L^S)^D(1 - L)^d X_t = \theta(L)\Theta(L^S)\varepsilon_t \quad (36)$$

The two new items are the seasonal components, namely:

$$\Phi(L^S) = 1 - \Phi_1 L^S - \Phi_2 L^{2S} - \cdots - \Phi_P L^{PS} \quad (37)$$
$$\Theta(L^S) = 1 + \Theta_1 L^S + \Theta_2 L^{2S} + \cdots + \Theta_Q L^{QS} \quad (38)$$

So more explicitly the complete seasonal ARIMA model can be expressed as:

---

[1] A further explanation of the seasonal ARIMA model can be found on: https://onlinecourses.science.psu.edu/stat510/node/67

$$\left(1-\sum_{k=1}^{P}\Phi_k L^{kS}\right)\left(1-\sum_{k=1}^{p}\phi_k L^{k}\right)(1-L^S)^D(1-L)^d X_t=\left(1+\sum_{k=1}^{Q}\Theta_k L^{kS}\right)\left(1+\sum_{k=1}^{q}\theta_k L^{k}\right)\varepsilon_t+\tilde{\mu} \quad (39)$$

It is important to notice that the left side of equation (39) the seasonal and non-seasonal AR components multiply each other, and on the right side of equation (39) the seasonal and non-seasonal MA components also multiply each other.

5.1.2.7 Exponential Smoothing

Exponential Smoothing Methods are widely used in industry. Their popularity is due to several practical considerations in short-range forecasting. This method was originally developed by Brown and Holt in the 1950s (Brown, 1972). The Simple Smoothing represents the time series by $X_t = b + \varepsilon_t$. Where ε_t is a random component with mean zero and variance σ^2.The level b is assumed to be constant in any local segment of the series but may change slowly over time (Gardner, 2006). Exponential smoothing methods can be upgraded to more complexity; double and triple exponential smoothing applications can also be found in the literature (Shan, Hu, Wang, & Liu, 2014). Other extension of the Exponential Smoothing are the non-seasonal, additive seasonal and multiplicative seasonal models, examples of these is the Holt-Winters linear trend models (Gardner, 2006).

The idea behind exponential smoothing is that it may be sensible to attach larger weights to more recent observations than to observations from the distant past. In Simple Exponential Smoothing forecasts are calculated using weighted averages where the weights decrease exponentially as observations come from further in the past – the smallest weights are associated with the oldest observations:

$$\hat{X}_{T+1/T} = \alpha X_T + \alpha(1-\alpha)X_{T-1} + \alpha(1-\alpha)^2 X_{T-2} + \cdots \quad (40)$$

Where $0 \le \alpha \le 1$ is the smoothing parameter. The one-step-ahead forecast for time *T+1* is a weighted average of all the observations in the series $X_1, \ldots, X_T$. The rate at which the weights decrease is controlled by the parameter α (Hyndman & Athanasopoulos, 2014).

The packages *forecast* for *R*, includes a function called *ets* which can automatically select among the most important exponential methods, namely: (1) Simple Exponential Smoothing, (2) Holt's linear method, (3) Exponential trend method, (4) Additive damped trend methods, (5) Multiplicative Damped Trend Method, (6) Additive Holt-Winters method, (7) multiplicative Holt-Winters method and (8) Holt-winters damped method.

The taxonomy of Exponential Smoothing methods can be summarized as (Figures 2 to 4 and Table 2):

		Seasonal Component	
Trend	N	A	M
Component	(None)	(Additive)	(Multiplicative)
N (None)	(N,N)	(N,A)	(N,M)
A (Additive)	(A,N)	(A,A)	(A,M)
A_d (Additive damped)	(A_d,N)	(A_d,A)	(A_d,M)
M (Multiplicative)	(M,N)	(M,A)	(M,M)
M_d (Multiplicative damped)	(M_d,N)	(M_d,A)	(M_d,M)

Figure 2. A two-way classification of exponential smoothing methods (Hyndman & Athanasopoulos, 2014)

(N,N)	=	simple exponential smoothing
(A,N)	=	Holts linear method
(M,N)	=	Exponential trend method
(A_d,N)	=	additive damped trend method
(M_d,N)	=	multiplicative damped trend method
(A,A)	=	additive Holt-Winters method
(A,M)	=	multiplicative Holt-Winters method
(A_d,M)	=	Holt-Winters damped method

Figure 3. Exponential Smoothing Methods (Hyndman & Athanasopoulos, 2014)

Table 2. *Formulas for recursive calculations and point forecasts*

Trend	Seasonal N	Seasonal A	Seasonal M
N	$\hat{y}_{t+h\|t} = \ell_t$ $\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)\ell_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = \ell_t s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)\ell_{t-1}$ $s_t = \gamma(y_t/\ell_{t-1}) + (1-\gamma)s_{t-m}$
A	$\hat{y}_{t+h\|t} = \ell_t + hb_t$ $\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + hb_t + s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = (\ell_t + hb_t)s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} + b_{t-1})) + (1-\gamma)s_{t-m}$
A_d	$\hat{y}_{t+h\|t} = \ell_t + \phi_h b_t$ $\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + \phi_h b_t + s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - \phi b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = (\ell_t + \phi_h b_t)s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} + \phi b_{t-1})) + (1-\gamma)s_{t-m}$
M	$\hat{y}_{t+h\|t} = \ell_t b_t^h$ $\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}b_{t-1}$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t b_t^h + s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1}b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = \ell_t b_t^h s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1}b_{t-1})) + (1-\gamma)s_{t-m}$
M_d	$\hat{y}_{t+h\|t} = \ell_t b_t^{\phi_h}$ $\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}b_{t-1}^{\phi}$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}^{\phi}$	$\hat{y}_{t+h\|t} = \ell_t b_t^{\phi_h} + s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}^{\phi}$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}^{\phi}$ $s_t = \gamma(y_t - \ell_{t-1}b_{t-1}^{\phi}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = \ell_t b_t^{\phi_h} s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}^{\phi}$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}^{\phi}$ $s_t = \gamma(y_t/(\ell_{t-1}b_{t-1}^{\phi})) + (1-\gamma)s_{t-m}$

Source: (Hyndman & Athanasopoulos, 2014). In each case, ℓ_t denotes the series level at time *t*, b_t denotes the slope at time *t*, s_t denotes the seasonal component of the series at time *t*, and *m*, denotes the number of seasons in a year; α, β^*, γ and ϕ are smoothing parameters. $\phi_h = \phi + \phi^2 + \cdots + \phi^h$ and $h_m^+ = \lfloor (h-1) mod\ m \rfloor + 1$.

Method	Initial values
(N,N)	$\ell_0 = y_1$
(A,N) (A_d,N)	$\ell_0 = y_1, b_0 = y_2 - y_1$
(M,N) (M_d,N)	$\ell_0 = y_1, b_0 = y_2/y_1$
(A,A) (A_d,A)	$\ell_0 = \frac{1}{m}(y_1 + \cdots + y_m)$ $b_0 = \frac{1}{m}\left[\frac{y_{m+1} - y_1}{m} + \cdots + \frac{y_{m+m} - y_m}{m}\right]$ $s_0 = y_m - \ell_0,\ s_{-1} = y_{m-1} - \ell_0,\ \ldots,\ s_{-m+1} = y_1 - \ell_0$
(A,M) (A_d,M)	$\ell_0 = \frac{1}{m}(y_1 + \cdots + y_m)$ $b_0 = \frac{1}{m}\left[\frac{y_{m+1} - y_1}{m} + \cdots + \frac{y_{m+m} - y_m}{m}\right]$ $s_0 = y_m/\ell_0,\ s_{-1} = y_{m-1}/\ell_0,\ \ldots,\ s_{-m+1} = y_1/\ell_0$

Figure 4. Initialization strategies for some of the more commonly used exponential smoothing methods (Hyndman & Athanasopoulos, 2014)

5.1.2.8 Multivariate Time Series

Multivariate Time Series analysis considers simultaneously multiple time series. It is a branch of multivariate statistical analysis but deals specifically with dependent data. Understanding the relationships between those factors and providing accurate predictions of those variables are valuable in decision making. The objectives of multivariate time series analysis include (1) To study the dynamic relationships between variables, (2) to improve the accuracy of prediction.

Let $\{z_{it}\}$ be the *i*th component of the multivariate time series $\boldsymbol{z_t}$[2]. If there is interest in predicting $\boldsymbol{z_{T+1}}$ based on the data$\{\boldsymbol{z_1}, \ldots, \boldsymbol{z_T}\}$. To this end, the following model can be used:

$$\boldsymbol{\hat{z}_{T+1}} = \boldsymbol{g}(\boldsymbol{z_T}, \boldsymbol{z_{T-1}}, \ldots, \boldsymbol{z_1}) \tag{41}$$

Where $\boldsymbol{\hat{z}_{T+1}}$ denotes a prediction of $\boldsymbol{z_{T+1}}$ and *g(.)* is some suitable function. The goal of multivariate time series analysis is to specify the function *g(.)* based on the available data. In many applications, *g(.)*, is a smooth, differentiable function and can be well approximated by a linear function, say,

$$\boldsymbol{\hat{z}_{T+1}} \approx \boldsymbol{\pi_0} + \boldsymbol{\pi_1 z_T} + \boldsymbol{\pi_2 z_{T-1}} + \cdots + \boldsymbol{\pi_T z_1} \tag{42}$$

Where $\boldsymbol{\pi_0}$ is a *k*-dimensional vector, and $\boldsymbol{\pi_i}$ are *k x k* constant real-valued matrices (for *i* = 1,…,*T*). Let $\boldsymbol{a_{T+1}} = \boldsymbol{z_{T+1}} - \boldsymbol{\hat{z}_{T+1}}$ be the forecast error. The prior equation states that

$$\boldsymbol{\hat{z}_{T+1}} \approx \boldsymbol{\pi_0} + \boldsymbol{\pi_1 z_T} + \boldsymbol{\pi_2 z_{T-1}} + \cdots + \boldsymbol{\pi_T z_1} + \boldsymbol{a_{T+1}} \tag{43}$$

Under linearity assumption, $\boldsymbol{z_T}$ follows a continuous multivariate probability distribution.

There are some assumptions which must hold in order to work with multivariate time series, one of these is weakly stationarity. A *k*-dimensional time series $\boldsymbol{z_T}$ is said to be **weakly stationary** if (a) $(\boldsymbol{z_T}) = \boldsymbol{\mu}$, a k-dimensional constant vector, and (b) $Cov(\boldsymbol{z_T}) = \boldsymbol{E}[(\boldsymbol{z_T} - \boldsymbol{\mu})(\boldsymbol{z_T} - \boldsymbol{\mu})'] = \sum_{\boldsymbol{z}}$ **,** a constant *k x k* positive-definite matrix. Thus, the mean and covariance matrices of a weakly stationary time series $\boldsymbol{z_T}$ do not depend on time, that is, the first two moments of $\boldsymbol{z_T}$ are time invariant.

A *k*-dimensional time series $\boldsymbol{z_T}$ is **strictly stationary** if the joint distribution of the *m* collection, $(\boldsymbol{z_{t_1}}, \ldots, \boldsymbol{z_{t_m}})$ is the same as that of $(\boldsymbol{z_{t_{1+j}}}, \ldots, \boldsymbol{z_{t_m}})$ where *m, j* and $(t_1, \ldots, t_m)$ are arbitrary positive integers.

In statistical terms, strict stationarity requires the probability distribution of an arbitrary collection of $\boldsymbol{z_T}$ to be time invariant. For instance, the sequence of independent and identically distributed random vectors of a standard multivariate normal distribution. Notwithstanding, strict stationarity is hard to verify in practice.

As in univariate time series, multivariate time series can be represented as Moving Average (MA) and Autoregressive (AR) models. A *k*-dimensional multivariate time series $\boldsymbol{z_T}$ is **linear** if it can be represented as a Moving Average (MA) model, namely:

$$\mathbf{z_t} = \boldsymbol{\mu} + \sum_{\boldsymbol{i=0}}^{\infty} \boldsymbol{\psi_i a_{t-i}} \tag{44}$$

[2] Here the **bold letter** notation is introduced to indicate a matrix or a vector.

in which $\boldsymbol{\mu}$ is a *k*-dimensional constant vector, $\boldsymbol{\psi_o} = \boldsymbol{I_k}$ *k x k* identity matrix, $\boldsymbol{\psi_i}$ $(\boldsymbol{i} > \boldsymbol{0})$ *k x k* constant matrices, $\boldsymbol{a_t}$ sequence of independent and identity distributed random vectors with mean zero and positive definite covariance matrix $\sum_{\boldsymbol{a}}$.

A time series is to be **invertible** if it can be written as an Autoregressive Model (AR), namely:

$$\boldsymbol{z_t} = \boldsymbol{c} + \boldsymbol{a_t} + \sum_{\boldsymbol{j=1}}^{\infty} \boldsymbol{\pi_j z_{t-j}} \tag{45}$$

$\boldsymbol{c}$ is a *k*-dimensional constant vector, $\boldsymbol{a_t}$ as defined before, $\boldsymbol{\pi_j}$ *k x k* constant matrices. Consequently if $i \to 0, \boldsymbol{\pi_j} \to \infty$.

If the time series is both stationary and invertible, then these two model representations are equivalent, and one can obtain one representation from the other, namely, $\boldsymbol{\pi_l}$ can be obtained recursively from $\{\boldsymbol{\psi_i} | i = 1,2, \dots\}$ via

$$\boldsymbol{\pi_1} = \boldsymbol{\psi_1} \tag{46}$$

$$\boldsymbol{\pi_l} = \boldsymbol{\psi_l} - \sum_{i=0}^{l-1} \boldsymbol{\pi_l}\, \boldsymbol{\psi_{l-1}} \qquad l > 1$$

Notwithstanding, neither the AR representation nor the MA representation is particularly useful in estimation if they involve too many coefficient matrices. To facilitate model estimation, the coefficient matrices $\boldsymbol{\pi_j}$ and $\boldsymbol{\psi_i}$ will depend only on a finite number of parameters. This consideration leads to the use of vector autoregressive moving-average (VARMA) models, which are also known as the multivariate autoregressive moving-average (MARMA) models.

A general **VARMA(*p*,*q*)** model can be written as:

$$\boldsymbol{z_t} = \boldsymbol{\phi_0} + \sum_{i=1}^{p} \boldsymbol{\phi_i}\, \boldsymbol{z_{t-1}} + \boldsymbol{a_t} - \sum_{j=1}^{q} \boldsymbol{\theta_j}\, \boldsymbol{a_{t-j}} \tag{47}$$

Where *p* and *q* are nonnegative integers, $\boldsymbol{\phi_0}$ is a *k*-dimensional constant vector, $\boldsymbol{\phi_i}$ and $\boldsymbol{\theta_j}$ are *k x k* constant matrices, and $\boldsymbol{a_t}$ is a sequence of independent and identically distributed random vector with mean zero and positive-definite covariance matrix $\sum_{\boldsymbol{a}}$. Using the lag operator *L*, the VARMA model can be written in a more compact form as:

$$\boldsymbol{\phi}(L)\boldsymbol{z_t} = \boldsymbol{\phi_0} + \boldsymbol{\theta}(L)\boldsymbol{a_t} \tag{48}$$

Where $\boldsymbol{\phi}(L) = \boldsymbol{I_k} - \boldsymbol{\phi_1}L - \cdots - \boldsymbol{\phi_p}L^p$ and $\boldsymbol{\theta}(L) = \boldsymbol{I_k} - \boldsymbol{\theta_1}L - \cdots - \boldsymbol{\theta_q}L^q$ are matrix polynomials in *L*.

To measure the linear dynamic dependence of a stationary time series $\boldsymbol{z_t}$, the lag ***l*** **cross-covariance matrix** is defined, namely:

$$\boldsymbol{\Gamma_l} = cov(\boldsymbol{z_t}, \boldsymbol{z_{t-l}}) = \boldsymbol{E}[(\boldsymbol{z_t} - \boldsymbol{\mu})(\boldsymbol{z_t} - \boldsymbol{\mu})'] \tag{49}$$

$$= \begin{bmatrix} E(\boldsymbol{\tilde{z}_{1t}}, \boldsymbol{\tilde{z}_{1,t-l}}) & E(\boldsymbol{\tilde{z}_{1t}}, \boldsymbol{\tilde{z}_{2,t-l}}) & \cdots & E(\boldsymbol{\tilde{z}_{1t}}, \boldsymbol{\tilde{z}_{k,t-l}}) \\ \cdots & \cdots & \ddots & \cdots \\ E(\boldsymbol{\tilde{z}_{1t}}, \boldsymbol{\tilde{z}_{t-l}}) & E(\boldsymbol{\tilde{z}_{1t}}, \boldsymbol{\tilde{z}_{t-l}}) & \cdots & E(\boldsymbol{\tilde{z}_{1t}}, \boldsymbol{\tilde{z}_{2,t-l}}) \end{bmatrix}$$

$\boldsymbol{\mu} = \boldsymbol{E}(\boldsymbol{z_t})$ is the mean vector of $\boldsymbol{z_t}$ and $\tilde{\boldsymbol{z}}_{\boldsymbol{t}} = (\tilde{\boldsymbol{z}}_{\boldsymbol{1t}}, \dots, \tilde{\boldsymbol{z}}_{\boldsymbol{k\,t}}) = \boldsymbol{z_t} - \boldsymbol{\mu}$ is the mean adjusted time series.

This cross-covariance matrix is a function of *l*, not of the time index *t*, because $\boldsymbol{z_t}$ is stationary. For *l* = 0, $\boldsymbol{\Gamma_l}$ becomes $\boldsymbol{\Gamma_0}$, which is the covariance matrix of $\boldsymbol{z_t}$, that is, $\sum_{\boldsymbol{z}} = \boldsymbol{\Gamma_0}$.

Denote the (*i,j*)th element of $\boldsymbol{\Gamma_l}$ as $\gamma_{l,ij}$ that is, $\boldsymbol{\Gamma_l} = [\gamma_l, ij]$. $\gamma_{l,ij}$ is the covariance between $\boldsymbol{z_{i,t}}$ and $\boldsymbol{z_{j,t-1}}$. Therefore, for a positive lag *l*, $\gamma_{l,ij}$ can be regarded as a **measure of the linear dependence of the *i*th component** $\boldsymbol{z_{i,t}}$ on the *l*th lagged value of the *j*th component of $\boldsymbol{z_{j,t}}$.

Accordingly, the lag ***l*** **cross-correlation matrix (CCM)** $\boldsymbol{\rho_l}$ is defined as:

$$\boldsymbol{\rho_l} = \boldsymbol{D^{-1}}\Gamma_l\boldsymbol{D^{-1}} = \left[\rho_{l,ij}\right] \tag{50}$$

Where $\boldsymbol{D} = diag\{\sigma_1, \dots, \sigma_k\}$ is the diagonal matrix of the standard deviations of components of $\boldsymbol{z_t}$. Specifically, $\sigma_i^2 = Var(z_{it}) = \gamma_{l,ij}$, that is, the (*i,i)*th of $\boldsymbol{\Gamma_0}$. Obviously, $\boldsymbol{\rho_0}$ is symmetric with diagonal elements being 1. The off-diagonal elements of $\boldsymbol{\rho_0}$ are the instantaneous correlations between the components of $\boldsymbol{z_t}$. For > 0 , $\boldsymbol{\rho_l}$ is not symmetric in general because $\rho_{l,ij}$ is the correlation coefficient between z_{it} and $z_{j,t-l}$, whereas $\rho_{l,ji}$ is the correlation coefficient between z_{jt} and $z_{i,t-l}$. That is to say, the lag cross-correlation feature of a group of time series is **intransitive**.

Since in real life, there is normally only access to samples of the population of data, it is necessary to define the **Sample CCM.** Given the sample $\{z_t\}_{t=1}^{T}$, the sample mean vector and covariance matrix can be defined as:

$$\widehat{\boldsymbol{\mu}}_{\boldsymbol{z}} = \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{z_t} \tag{51}$$

$$\widehat{\boldsymbol{\Gamma}}_{\boldsymbol{0}} = \frac{1}{T-1}\sum_{t=1}^{T}(\boldsymbol{z_t} - \widehat{\boldsymbol{\mu}}_{\boldsymbol{z}})(\boldsymbol{z_t} - \widehat{\boldsymbol{\mu}}_{\boldsymbol{z}})' \tag{52}$$

The lag *l* sample cross-covariance matrix

$$\boldsymbol{\Gamma_l} = \frac{1}{T-1}\sum_{t=l+1}^{T}(\boldsymbol{z_t} - \widehat{\boldsymbol{\mu}}_{\boldsymbol{z}})(\boldsymbol{z_{t-l}} - \widehat{\boldsymbol{\mu}}_{\boldsymbol{z}})' \tag{53}$$

The *l* sample CCM is then:

$$\widehat{\boldsymbol{\rho}}_{\boldsymbol{l}} = \widehat{\boldsymbol{D}}^{\boldsymbol{-1}}\widehat{\Gamma}_l\widehat{\boldsymbol{D}}^{\boldsymbol{-1}} = \left[\hat{\rho}_{l,ij}\right] \tag{54}$$

Where $\widehat{\boldsymbol{D}} = diag\left\{\hat{\gamma}_{0,11}^{1/2}, \dots, \hat{\gamma}_{0,kk}^{1/2}\right\}$, in which $\hat{\gamma}_{0,11}$ is the (*i, i)*th element of $\widehat{\boldsymbol{\Gamma}}_{\boldsymbol{0}}$. If $\boldsymbol{z_t}$ is a stationary process and $\boldsymbol{a_t}$ follows a multivariate normal distribution, then $\widehat{\boldsymbol{\rho}}_{\boldsymbol{l}}$ is a consistent estimate of $\boldsymbol{\rho_l}$.

For the implementation of multivariate time series to a data set, it is necessary to test the zero cross-correlation using a statistical test with some significance level. A basic test is to detect the existence of linear dynamic dependence in the data. This amounts to testing the null hypothesis H_0: $\boldsymbol{p_1} = \cdots = \boldsymbol{p_m} = \boldsymbol{0}$ versus the alternative hypothesis H_0: $\boldsymbol{p_i} \neq \boldsymbol{0}$ for some *i* satisfying $1 \leq i \leq m$, where *m* is a positive integer. The ***Portmanteau test*** of univariate time series has been generalized to the multivariate case by several authors. In particular, the multivariate **Ljung-Box test** statistic is defined as:

$$Q_k(m) = T^2 \sum_{l=1}^{m} \frac{1}{T-l} tr(\hat{\boldsymbol{\Gamma}}_l' \hat{\boldsymbol{\Gamma}}_0^{-1} \hat{\boldsymbol{\Gamma}}_l \hat{\boldsymbol{\Gamma}}_l^{-1}) \tag{55}$$

Where $tr(\boldsymbol{A})$ is the trace of the matrix $\boldsymbol{A}$ and T is the sample size. This is referred to as the multivariate Portmanteau test. Under the null hypothesis that $\boldsymbol{\Gamma_l} = \boldsymbol{0}$ for $l > 0$ (no linear dependence in the data) and the condition that $\mathbf{z}_t$ is normally distributed, $Q_k(m)$ is asymptotically distributed as $\chi^2_{m\,k^2}$, that is, a chi-square distribution with $m\,k^2$ degrees of freedom.

5.1.2.9 Vector Autoregressive Model

Vector Autoregressive Model. The most commonly used multivariate time series model is the vector autoregressive (AR) model. First, the model is easy to estimate. One can use the least-squares (LS) method, the maximum likelihood (ML) method, or Bayesian method. All three estimates are asymptotically equivalent to the ML estimates and the ordinary least-squares (OLS). Second, the properties of VAR models are similar to the multivariate multiple linear regressions widely used in multivariate statistical analysis.

The multivariate time series $\boldsymbol{z_t}$ follows a VAR model of order *p*, VAR(*p*), if

$$\boldsymbol{z_t} = \boldsymbol{\phi_0} + \sum_{i=1}^{p} \boldsymbol{\phi_i}\, \boldsymbol{z_{t-1}} + \boldsymbol{a_t} \tag{56}$$

This is a special case of the VARMA(*p*,*q*) model with $q = 0$. With the back-shift operator, the model becomes $\boldsymbol{\phi}(L)\boldsymbol{z_t} = \boldsymbol{\phi_0} + \boldsymbol{a_t}$**,** as stated before**,** where $\boldsymbol{\phi}(L) = \boldsymbol{I_k} - \boldsymbol{\phi_1}L - \cdots - \boldsymbol{\phi_p}L^p$, $\boldsymbol{\phi}(L) = \boldsymbol{I_k} - \sum_{i=1}^{p} \boldsymbol{\phi_i}\, \boldsymbol{L^i}$ is a matrix polynomial of degree *p*. Therefore, the $\boldsymbol{\phi_i} = [\boldsymbol{\phi_{l\,,ij}}$] as the lag *l* AR coefficient matrix.

VAR(1) Models. This will be the VAR models considered in this document. A bivariate example will be used to clarify how the model works. The bivariate VAR (1) model can be written as:

$$\boldsymbol{z_t} = \boldsymbol{\phi_0} + \boldsymbol{\phi_1}\boldsymbol{z_{t-1}} + \boldsymbol{a_t} \tag{57}$$

This model can be written explicitly as

$$\begin{bmatrix} z_{1t} \\ z_{2t} \end{bmatrix} = \begin{bmatrix} \phi_{10} \\ \phi_{20} \end{bmatrix} + \begin{bmatrix} \phi_{1,11} & \phi_{1,12} \\ \phi_{1,21} & \phi_{1,22} \end{bmatrix} \begin{bmatrix} z_{1,t-1} \\ z_{2,t-1} \end{bmatrix} + \begin{bmatrix} a_{1t} \\ a_{2t} \end{bmatrix} \tag{58}$$

Or equivalently

$$\begin{aligned} z_{1t} &= \phi_{10} + \phi_{1,11} z_{1,t-1} + \phi_{1,12} z_{2,t-1} + a_{1t} \\ z_{2t} &= \phi_{20} + \phi_{1,21} z_{1,t-1} + \phi_{1,22} z_{2,t-1} + a_{2t} \end{aligned} \tag{59}$$

Thus, the (1,2)th element of $\boldsymbol{\phi_1}$, that is, $\phi_{1,12}$, shows the linear dependence of z_{1t} on $z_{2,t-1}$ in the presence of $z_{1,t-1}$. The (2,1)th element of $\boldsymbol{\phi_1}$, that is, $\phi_{1,21}$, measures the linear relationship between z_{2t} on $z_{1,t-1}$ in the presence of $z_{2,t-1}$.

If the off-diagonal elements of $\boldsymbol{\phi_1}$ are 0, that is, $\phi_{1,12} = \phi_{1,21} = 0$. Then z_{1t} and z_{2t} are not dynamically correlated. In this particular case, each series follows a univariate AR(1) model and can be handled accordingly. The two series are uncoupled.

If $\phi_{1,12} = 0$, but $\phi_{1,21} \neq 0$, then

$$\begin{aligned} z_{1t} &= \phi_{10} + \phi_{1,11} z_{1,t-1} + a_{1t} \\ z_{2t} &= \phi_{20} + \phi_{1,21} z_{1,t-1} + \phi_{1,22} z_{2,t-1} + a_{2t} \end{aligned} \tag{60}$$

This particular model shows that z_{1t} does not depend on the past value of z_{2t}, but z_{2t} does depends on the past value of z_{1t}. Consequently, there is a unidirectional relationship with z_{1t} acting as the input variable and z_{2t} as the output variable. In the statistical literature, the two series z_{1t} and z_{2t} are said to have a transfer function relationship. Transfer function models, which can be regarded as a special case of the VARMA model, are useful in control engineering as one can adjust the value of z_{1t} to influence of z_{2t}. In the econometric literature, the model implies the existence of **Granger causality** between the two series with z_{1t} causing z_{2t}, but not being caused by z_{2t}.

Granger introduces the concept of causality, which is easy to deal with for an AR model (Granger, 1969). Consider a bivariate series and the *h*-step-ahead forecast. In this case, a VAR model and a univariate model can be used for individual components to produce forecasts. It can be asseverated that z_{1t} causes z_{2t} if the bivariate forecast for z_{2t} is more accurate than its univariate forecast. That is to say, the accuracy of a forecast is measured by the variance of its forecast error; under Granger's framework, z_{1t} causes z_{2t} if the past information of z_{1t} improves the forecast of z_{2t} (Tsay, 2014).

5.1.3 Econometric Methods

Econometric methods, namely, Regression Analysis, Autoregressive Moving Average with exogenous inputs and Disaggregate Choice Models are also used in forecasting (Berkovec, 1985). The advantages of the Disaggregate Discrete Choice Approach are in its more plausible theoretical structure and the ability to represent a wider range of policy options due to the much greater degree of differentiation of the elements to be forecasted. This model tries to find an Equilibrium in price and in quantity for the demand and supply of multiple products.

5.1.4 Other Methods

Other methods like Linear Combination of Forecasts, Neural Networks and Prophet Algorithm can be applied to predict costumers demand.

5.1.4.1 Combination of Forecasts

When several forecasts of the same event are available, it is natural to try to find a (linear) combination of these forecasts that is the 'best' in some sense (Claeskens et al., 2016). If 'best' is defined in terms of the mean squared error and the variances and covariances of the forecast are known, then optimal weights can be derived. Empirical evidence and extensive simulations show that the estimated optimal forecast combination typically does not perform well, and that the arithmetic mean often performs better. Additionally, (Smith & Wallis, 2009) also said that a simple average of competing forecasts is expected to be more accurate than a weighted combination. This empirical fact has become known as the 'forecast combination puzzle'.

5.1.4.2 Artificial Neural networks

These are forecasting methods that are based on simple mathematical models of the brain. They allow complex nonlinear relationships between response variable and its predictors. A neural network can be thought of as a network of "neurons" organized in layers. The predictors (or inputs) form the bottom layer, and the forecast (or outputs) form the top layer. There may be intermediate layers containing "hidden neurons" (Hyndman & Athanasopoulos, 2014).

The very simplest networks contain hidden layers and are equivalent to linear regression. Figure 5 shows a neural network version of a linear regression with four predictors. The coefficient attached to these predictors are called weights. The forecasts are obtained by a linear combination of the inputs. The weights are selected in the neural network framework using a "learning algorithm" that minimizes a "cost function" such as Mean Squared Error.

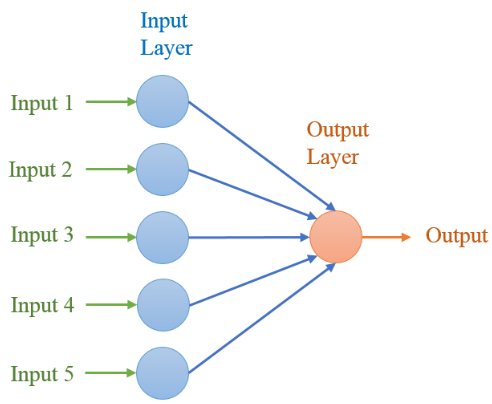


Figure 5. Simple Neural Network

Once an intermediate layer with hidden neurons is added, the neural network becomes non-linear. Figure 6 shows an example.

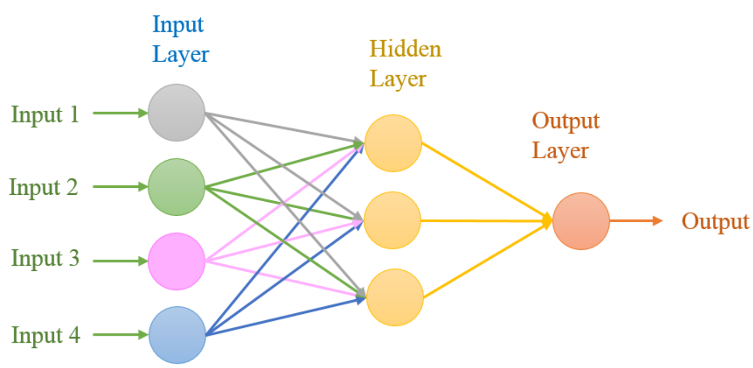


Figure 6. Neural Network with one hidden layer with three neurons

This is known as a **multilayer feed-forward network** where each layer of nodes receives inputs from the previous layers. The outputs of nodes in one layer are inputs to the next layer. The inputs to each node are combined using a weighted linear combination. The result is then modified by a nonlinear function before being output. For example, the inputs into hidden neuron *j* in Figure 3 are linearly combined to give:

$$z_j = b_j + \sum_{i=1}^{4} w_{i,j} x_i \tag{61}$$

In the hidden layer, this is then modified using a nonlinear such as a sigmoid:

$$s(z) = \frac{1}{1 + e^{-z}} \tag{62}$$

The parameters b_1, b_2, b_3 and w_1, w_2, w_3 are "learned" from the data. The values of the weights are often restricted to prevent them becoming too large. The parameters that restricts the weights is known as the "decay parameter" and is often set to be equal to 0.1.

The weights take random values to begin with, which are then updated using the observed data. Consequently, there is an element of randomness in the predictions produced by a neural network. Therefore, the network is usually trained several times using different random starting points, and the results are averaged.

The number of hidden layers, and the number of nodes in each hidden layer, must be specified in advance.

Neural Network auto-regression. With time series data, lagged values of the time series can be used as inputs to neural networks. Just as lagged values are used in a linear auto-regression model. The **nnetar()** function from the package forecast (Hyndman & Athanasopoulos, 2014) fits an $NNAR(p, P, k)_m$ model. Where p is the number of lagged input values, P is the number of last observed values of the same season, k is the number of nodes in the hidden layer, and m is the number of periods per season.

With seasonal data, it is useful to also add the last observed values from the same season as inputs. For example, an $NNAR(3, 2, 1)_{12}$ model has inputs $y_{t-1}, y_{t-2}, y_{t-3}$ and y_{t-12} and two neurons in the hidden layer. More generally, an $NNAR(p, P, k)_m$ model has inputs $y_{t-1}, y_{t-2}, y_{t-3}, \dots, y_{t-p}$ and $y_{t-m}, y_{t-2m}, y_{t-Pm}$ and k in the hidden layer. A $NNAR(p, P, 0)_m$ is equivalent to an $ARIMA(p, 0,0)(P, 0, 0)_m$ model but without the restrictions on the parameters to ensure stationarity.

In the *nntar()* function, if the values of p and P are not specified, then they are automatically selected. For non-seasonal time series, the default is the optimal number of lags (according to the AIC for a linear AR(p) model). For seasonal time series, the default values are P=1 and p is chosen from the optimal linear model fitted to the seasonally adjusted data. If k is not specified it is set to $k = (p + P + 1)/2$ (rounded to the nearest integer) (Hyndman & Athanasopoulos, 2014).

5.1.4.3 Prophet Algorithm

Prophet algorithm is a new forecasting algorithm released and developed by Facebook Research, it can be accessed as a free forecasting tool available in Python and R. They propose a modular regression model with interpretable parameters that can be intuitively adjusted by analysts with domain knowledge about time series. The algorithm uses a decomposable time series model with three main model components: trend *g(t)*, seasonality *s(t)*, and holidays *h(t).*

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \tag{63}$$

The essence of a structural time series model is that it is formulated in terms of independent component which have a direct interpretation in term of quantities of interest. One of the most important models for economic time series is the basic structural model: this consists of a trend, a seasonal and an irregular component (Harvey & Peters, 1990) . The specification is similar to a Generalized Additive Model (GAM), a class of regression model with potentially non-linear smoothers applied to the regressors. The GAM formulations has the advantage that it decomposes easily and accommodates new components as necessary; for instance, when a new source of seasonality is identified (Taylor & Letham, 2017). GAM's also fit very quickly, either using backfittig or Limited-Memory Broyden-Fletcher-Goldfarb-Schanno Algorithm (L-BFGS) (Byrd, Lu, Nocedal, & Zhu, 1994), this latter is an optimization algorithm in the family of quasi-Newton methods, popularly used for parameter estimation in machine learning.

The model is, in effect, framing the forecast problem as a curve-fitting exercise, which is inherently different from time series models that explicitly account for the temporal dependence structure in the data. While some important inferential advantages are discarded when using a generative model such as an ARIMA, this formulation provides several practical advantages:

(1) Flexibility can easily accommodate seasonality with multiple periods and let the analyst make different assumptions about trends.
(2) Unlike with ARIMA models, the measurements do not need to be regularly spaced, and missing values do not need to be interpolated *e.g.* from removing outliers.
(3) Fitting is very fast, allowing the analyst to interactively explore many model specifications, for example in an R-Shiny application.
(4) The forecasting model has easily interpretable parameters that can be changed by the analyst to impose assumptions on the forecast.

The **trend model** behaves as a sort of growth, which can be typically modeled using the logistic growth model, the resulting equation is:

$$g(t) = \frac{C(t)}{1 + \exp\left(-(k + \boldsymbol{a}(t)^{\mathrm{T}}\boldsymbol{\delta})\big(t - (m + \boldsymbol{a}(t)^{\mathrm{T}}\boldsymbol{\gamma})\big)\right)} \tag{64}$$

In which $C(t)$ is a vector of the expected carrying capacities of the system at any point in time, k is the growth rate, and m is an offset parameter. Trend changes in the growth model can be incorporated by explicitly defining changepoints where the growth rate is allowed to change. The model assumes there are S changepoint at times s_j for $j = 1, .., S$. Then, a vector of rate adjustments $\boldsymbol{\delta} \in \mathbb{R}^S$ is defined, where $\delta_{\boldsymbol{j}}$ is the change in rate that occurs at time $s_{\boldsymbol{j}}$. The rate at any time *t* is then the base rate *k*, plus all the adjustment up to that point: $k + \sum_{j\,:t>s_j} \delta_j$. This is represented more cleanly by defining a vector $\boldsymbol{a(t)} \in \{0,1\}^S$ such that:

$$a_j(t) = \begin{cases} 1, & if\ t < s_j \\ 0, & otherwise \end{cases} \tag{65}$$

Finally, when the rate *k* is adjusted, the offset parameter *m* must also be adjusted to connect the endpoints of the segments. The correct adjustment at changepoint *j* is recursively computed as:

$$\gamma_j = \left(s_j - m - \sum_{l<j} \gamma_l \right) \left(1 - \frac{k + \sum_{l<j} \delta_l}{k + \sum_{l\leq j} \delta_l} \right) \tag{66}$$

The changepoints s_j could be specified by the analyst using known dates of product launches and other growth-altering events or may be automatically selected given a set of candidates. Prophet algorithm specify a large number of changepoints (*e.g.*, one per month for a several year history and use the prior $\boldsymbol{\delta} \backsim Laplace(0, \tau)$). The parameter τ directly controls the flexibility of the model in altering its rate. Importantly, a sparse prior on the adjustments $\boldsymbol{\delta}$ has no impact on the primary growth rate *k*, so as τ goes to 0 the fit reduces to standard (not-piecewise) logistic or linear growth.

Business time series often have multi-period seasonality as a result of the human behaviors they represent. For instance, a 5-day work week can produce effects on a time series that repeat each week, while vacation schedules and school breaks can produce effects that repeat each year. The model relies on Fourier series to provide a flexible model of periodic effects, since to fit and forecast seasonality it is necessary to specify models that are periodic functions of *t*. *P* will be the regular period expected in the time series (e.g. $P = 365.25$ for yearly data or $P = 7$ for weekly data, when this variable is measured in days). Seasonal effects can be arbitrary approximated by:

$$s(t) = \sum_{n=1}^{N} \left(a_n cos\left(\frac{2\pi nt}{P}\right) + b_n sin\left(\frac{2\pi nt}{P}\right) \right) \tag{67}$$

Fitting seasonality requires estimating the *2N* parameters $\boldsymbol{\beta} = [a_1, b_1, \dots, a_N, b_N]^{\mathrm{T}}$. This is done by constructing a matrix of seasonality vectors for each value of *t* in the historical and future data, for example with yearly seasonality and $N = 10$.

$$X(t) = \left[cos\left(\frac{2\pi(1)t}{365{,}25}\right), \dots, sin\left(\frac{2\pi(10)t}{365{,}25}\right) \right] \tag{68}$$

The seasonal component is then:

$$s(t) = X(t)\,\boldsymbol{\beta} \tag{69}$$

In the generative model $\boldsymbol{\beta} \backsim Normal(0, \sigma^2)$ to impose a smoothing prior on the seasonality.

Holidays and events provide large, somewhat predictable shocks to many business time series and often do not follow a periodic pattern, so their effects are not well modeled by a smooth cycle. Many countries around the world have major holidays that follow the lunar calendar. The impact of a particular holiday on the time series is often similar year after year, so it is important to incorporate it into the forecast.

Prophet Algorithm allows the analyst to provide a custom list of past and future events, identified by the event or holiday's unique name. Incorporating a list of holidays into the model is made straightforward by assuming that effects of holidays are independent. For each holiday i, let D_i be the set of past and future dates for that holiday. An indicator function is added representing whether time *t* is during holiday i, and assign each holiday a parameter κ_i which is the corresponding change in the forecast. This is done in a similar way as seasonality by generating a matrix of regressors

$$Z(t) = [\mathbf{1}(t \in D_1), \dots, \mathbf{1}(t \in D_L)] \tag{70}$$

And taking

$$h(t) = Z(t)\,\boldsymbol{\kappa} \tag{71}$$

As with seasonality, a prior $\boldsymbol{\kappa} \backsim Normal(0, \sigma^2)$ is used.

It is often important to include effects for a window of days around a particular holiday, such as the weekend of Thanksgiving. To account for that, additional parameters for the days surrounding the holiday are included, essentially treating each of the days in the window around the holiday as a holiday itself.

5.2 Choosing between competitive models

Selecting the model that provides the best fit to historical data generally does not result in a forecasting method that produces the best forecasts of new data. Concentrating too much on the model that produces the best historical fit often results in overfitting, or including too many parameters or terms in the model just because these additional terms improve the model fit (Montgomery, 2016).

In general, the best approach is to select the model that results in the smallest standard deviation (or mean squared error) of the one-step-ahead forecast error when the model is applied to data that were not used in the fitting process. Some authors refer to this as an ***out-of-sample*** forecast error standard deviation (or mean squared error). A standard way to measure this out-of-sample performance is by utilizing some form of data splitting; that is, divide the time series data into two segments-one for model fitting and the other for performance testing. Sometimes data splitting is called **cross-validation**. A good rule is to have at least 20 to 25 observations in the performance testing data set.

When evaluating the fit of the model to historical data, there are several criteria that may be of value. The **mean squared error** of the residuals is:

$$s^2 = \frac{\sum_{t=1}^{T} e_t^2}{T - p} \tag{72}$$

Where T periods of data have been used to fit a model with p parameters and e_t is the residual from the model-fitting process in period t. The mean squared error s^2 is just the sample variance of the residuals and it is an estimator of the variance of the model errors.

Another criterion is the R-squared statistic

$$R^2 = 1 - \frac{\sum_1^T e_t^2}{\sum_1^T (y_t - \bar{y})^2} \tag{73}$$

The denominator of Eq. (73) is just the total sum of squares of the observations, which is constant (not model dependent), and the numerator is just the residual sum of squares. Therefore, selecting the model that maximizes R^2 is equivalent to selecting the model that minimizes the sum of the squared residuals. Large values R^2 suggest a good fit to the historical data. Because the residual sum of squares always decreases when parameters are added to the model, relying on R^2 to select a forecasting model encourages overfitting or putting in more parameters than are necessary to obtain good forecasts. A large value of R^2 does not ensure that the out-of-sample one-step-ahead forecast errors will be small.

A better criterion is the "adjusted" R^2 statistic, defined as:

$$R^2 = 1 - \frac{\frac{\sum_{t=1}^{T} e_t^2}{T-p}}{\sum_1^T (y_t - \bar{y})^2/(T-1)} = 1 - \frac{s^2}{\sum_1^T (y_t - \bar{y})^2/(T-1)} \tag{74}$$

The adjustment is a "size" adjustment- that is, adjust for the number of parameters in the model. Note that a model that maximizes the adjusted *R* statistic is also the model that minimizes the residual mean square.

Two other important criteria are the **Akaike Information Criterion (AIC)** and the **Schwarz Bayesian Information Criterion (abbreviated as BIC or SIC).** These two criteria penalize the sum of squared residuals for including additional parameters in the model. Models that have small values of the AIC or BIC are considered good models.

$$AIC = ln\left(\frac{\sum_{t=1}^{T} e_t^2}{T}\right) + \frac{2p}{T} \tag{75}$$

$$BIC = ln\left(\frac{\sum_{t=1}^{T} e_t^2}{T}\right) + \frac{p\, ln(T)}{T} \tag{76}$$

One way to evaluate model selection criteria is in terms of **consistency**. A model selection criterion is consistent if it selects the true model when the true model is among those considered with probability approaching unity as the sample size becomes large, and if the true model is not among those considered, it selects the best approximation with probability approaching unity as the sample size becomes large. It turns out that s^2, the adjusted R^2, and the AIC are all inconsistent, because they do not penalize for adding parameters heavily enough. Relying on these criteria tends to result in overfitting. The BIC, which carries a heavier "size adjustment" penalty, is consistent.

Consistency, however, does not tell the complete story. It may turn out that the true model and any reasonable approximation to it are very complex. An **asymptotically efficient** model selection criterion chooses a sequence of models as T (amount of data available) gets large for which the one-step-ahead forecast error variances approach the one-step-ahead forecast error variance for the true model at least as fast as any other criterion. The AIC is asymptotically efficient but the BIC is not (Montgomery, 2016).

5.3 Monitoring a Forecasting Model

5.3.1 Error measures

Another important feature of forecasting is monitoring, which is essential to ensure that the system remains in control. This is especially true when the system is based on simple exponential smoothing, which will lag any trend in the data. This is essential in inventory control, for example, because of the need to take-off-line action when there is a significant change in demand. If demand goes up, new orders should be placed on a priority basis. If demand goes down, any unneeded orders should be canceled to prevent excess inventory investment (Gardner, 2006).

At the forecast accuracy level, the Mean Absolute Deviation (MAD), Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE) and Average Error (AE) are the most common forecast error indicators used by forecasters. The most commonly used accuracy measure is the MAPE. This widespread use of MAPE suggests that many forecasters agree on the use of an error measure which can be adjusted for scale in the data (Fildes & Goodwin, 2007).

Errors are also called residuals, some authors have developed techniques to follow how residuals behave when forecast is carried out, this is called tracking signals. The first tracking signal used in forecasting was the simple cumulative sum *cusum* of the residuals, developed by (Brown, 1972). The simple *cusum* is defined as the ratio of the sum of the errors at the end of each period to the smoothed mean absolute deviation (MAD) of the errors, it will be explained further in more detail. The ratio should fluctuate around zero if the errors are unbiased.

One possible problem with the simple *cusum* is that it may give an unreasonable number of false alarms.

5.3.2 Control Charts

Developing and implementing procedures to monitor the performance of the forecasting model is an essential component of good forecasting system design. No matter how much effort has been expended in developing the forecasting model, and regardless of how well the model works initially, over time it is likely that its performance will deteriorate. The underlying pattern of the time series may change, either because the internal inertial forces that drive the process may evolve through time, or because of external events such as new customers entering the market.

The one-step-ahead forecast errors $e_t(1)$ are typically used for forecast monitoring. The reason for this is that changes in the underlying time series will also typically be reflected in the forecast errors. For example, if a level change occurs in the time series, the sequence of the forecast errors will no longer fluctuate around zero; that is, a positive or negative bias will be introduced.

Monitoring forecasting model performance can be achieved through (Montgomery, 2016) :

- Shewhart Control Charts
- Cumulative Sum Control Chart (CUSUM)
- Exponentially Weighted Moving Average (EWMA)

A **Shewhart Control Chart** is a plot of the forecast error versus time containing a center line that represents the average (or the target value) of the forecast error and a set of control limits that are designed to provide an indication that the forecasting model performance has changed. The center line is usually taken as either zero (which is the anticipated forecast error for an unbiased forecast) or the average forecast error, and the control limits are typically placed at three standard deviations of the forecast errors above and below the center line. Forecast error that plot outside the control limits would indicate model inadequacy, or possibly the presence of unusual observations such as outliers in the data.

Because a Shewhart control chart exhibits statistical control, a conclusion would be that there is no strong evidence of statistical inadequacy in the forecasting model. Therefore, these control limits would be retained and used to judge the performance of future forecasts (in other words, the control limits are not recalculated). However, the stable control chart does not imply that the forecasting performance is satisfactory in the sense that the model results in small forecast errors. In the quality control literature (Montgomery, 2016), these two aspects of process performance are referred to as control and capability, respectively. It is possible for the forecasting process to be stable or in statistical control but not capable, that is to say, the forecasting process produces forecast errors that are unacceptably large.

CUSUM and EWMA charts are more effective at detecting smaller changes or disturbances in the forecasting model performance than the Shewhart control chart.

The **CUSUM** is very effective in detecting level changes in the monitored variable. It works by accumulating deviations of the forecast error that are above the desired target value T_0 (usually either zero or the average forecast error) with one statistic C^+ and deviation that are below the target with another statistic C^-. The statistics C^+ *and* C^- are called the upper and lower CUSUMs, respectively. They are computed as follows:

$$C_t^+ = \max[0, e_t(1) - (T_o + K) + C_{t-1}^+] \quad (77)$$
$$C_t^- = \min[0, e_t(1) - (T_o - K) + C_{t-1}^-] \quad (78)$$

Where the constant K, usually called the reference value, is frequently chosen as $K=0.5\sigma_{e(1)}$ *and* $\sigma_{e(1)}$ is the standard deviation of the one-step-ahead forecast errors. The logic is that if the forecast errors begin to systematically fall on one side of the target value (or zero), one of the CUSUMs will increase in magnitude. When this increase becomes large enough, an out-of-control signal is generated. The decision rule is to signal if the statistic C^+ exceeds a decision internal $H=5\sigma_{e(1)}$ or if C^- exceeds –H.

A control chart based on the **EWMA** is also useful for monitoring forecast errors. The EWMA applied to the one-step-ahead forecast errors is:

$$\bar{e}_t(1) = \lambda e_t + (1 - \lambda)\bar{e}_{t-1}(1) \quad (79)$$

Where $0 < \lambda < 1$ is a constant (usually called the smoothing constant) and the starting value of the WEMA (required at the first observation) is either 0 or the average of the forecast errors. Typical values of the smoothing constant for an EWMA control chart are $0.05 < \lambda < 0.2$.

The EWMA is a weighted average of all current and previous forecast errors, and the weights decrease geometrically with the "age" of the forecast error. Recursively the following equation can be obtained:

$$\bar{e}_n(1) = \lambda \sum_{j=0}^{n-1} (1 - \lambda)^j \, e_{T-j}(1) + (1 - \lambda)^n \bar{e}_0(1) \quad (80)$$

The standard deviation of the EWMA is

$$\hat{\sigma}_{\bar{e}_t(1)} = \sigma_{e(1)} \sqrt{\frac{\lambda}{2 - \lambda}[1 - (1 - \lambda)^{2t}]} \quad (81)$$

$$\sigma_{e(1)} = \frac{0.8865\, MR}{n-1} \tag{82}$$

$$MR = \sum_{t=2}^{n} |e_t(1) - e_{t-1}(1)| \tag{83}$$

So, an EWMA control chart for the one-step-ahead forecast errors with a center line of *T* (target for the forecast errors) is defined as follows:

$$UCL = T + 3\sigma_{e(1)}\sqrt{\frac{\lambda}{2-\lambda}[1-(1-\lambda)^{2t}]} \tag{84}$$

$$Center\ Line = T$$

$$LCL = T - 3\sigma_{e(1)}\sqrt{\frac{\lambda}{2-\lambda}[1-(1-\lambda)^{2t}]} \tag{85}$$

5.3.3 Tracking Signals

The **Cumulative Error Tracking Signal (CETS)** (Montgomery, 2016) is based on the cumulative sum of all current and previous forecast errors:

$$Y(n) = \sum_{t=1}^{n} e_t(1) = Y(n-1) + e_n(1) \tag{86}$$

If the forecasts are unbiased, we would expect *Y(n)* to fluctuate around zero. To operationalize this, suppose that we have an estimate and form **the cumulative error tracking signal:**

$$CETS = \left|\frac{Y(n)}{\hat{\sigma}_{Y(1)}}\right| \tag{87}$$

If the CETS exceeds a constant, say, K_1, we would conclude that the forecasts are biased and that the forecasting model may be inadequate.

It is also possible to devise a **Smoothed Error Tracking Signal** based on the smoothed one-step Forecast Errors (SETS) (Montgomery, 2016) (Vidal Holguín, 2010). This would lead to a ratio:

$$SETS = \left|\frac{\bar{e}_n(1)}{\hat{\sigma}_{e_n(1)}}\right| \tag{88}$$

If the SETS exceed a constant, say K_2, this in an indication that the forecasts are biased and that there are potentially problems with the forecasting model. The CETS is very similar to the CUSUM control chart and the SETS is essentially equivalent to the EWMA control chart.

5.4 Outlier Detection Procedures

Most time series data are observational in nature. In addition to possible gross errors, time series data are often subject to the influence of some nonrepetitive events; for example, implementation of a new regulation, major changes in political or economic policy, or occurrence of a disaster. Consequently, discordant observations and various types of structural changes occur frequently in time series data. Whereas the usual time series model is designed to grasp the homogenous memory pattern of a time series, the presence of outlying observations or structural changes raises the question of efficiency and adequacy in fitting general autoregressive moving average (ARMA) models to time series data.

A common approach to deal with outliers in a time series is to identify the locations and the types of outliers and then use intervention models (Box & Tiao, 1975). There are some main important issues caused by outliers, i.e., (a) the presence of outliers may result in an inappropriate model, (b) even if the model is appropriately specified, outliers in a time series may still produce bias in parameter estimates and hence may affect the efficiency of outlier detection. A typical difficulty found in this approach was that both the types and locations of outliers may change at different iterations of model estimation, and (c) some outliers may not be identified due to a masking effect.

For problems b and c (Chen & Liu, 1993) designed a procedure that is less vulnerable to the spurious and masking effects during outlier detection and is able to jointly estimate the model parameters and outlier effects.

This procedure can be applied to general seasonal and nonseasonal ARMA processes. They defined a Y_t time series, which follows a general ARMA process of the form:

$$Y_t = \frac{\theta(L)}{\alpha(L)\phi(L)} a_t \qquad t = 1, \dots, n \tag{89}$$

Where n is the number of observations for the series; $\theta(L)$, $\alpha(L)$ and $\phi(L)$ are lag polynomials of L. The model may include a constant term when the nonstationary operator $\alpha(L)$ is contained on the left side of the model equation. To describe a time series subject to the influence of a nonrepetitive event, they consider the following model:

$$Y_t^* = Y_t + \omega \frac{A(L)}{G(L)H(L)} I_t(t_1) \tag{90}$$

Where Y_t follows a general ARMA process described in (89). And:

$$I_t(t_1) = \begin{cases} 1, & if\ t = t_1 \\ 0, & otherwise \end{cases} \tag{91}$$

Here $I_t(t_1)$ is an indicator function for the occurrence of an outlier impact, t_1 is the possibly unknown location of the outlier, and ω and $A(L)/G(L)H(L)$ denote the magnitude and the dynamic pattern of the outlier effect. If the location and the dynamic pattern of an event in known, then model (90) is the intervention model studied by Box and Tiao (1975). Chen & Lui consider the estimation problem when both the location and the dynamic pattern are not known a priori. The approach is to classify an outlier impact into four types by imposing a special structure on $A(L)/G(L)H(L)$. The types include an innovational outlier (IO), an additive outlier (AO), a level shift (LS), and a temporary change (TC). They can be defined as follows:

$$IO: \frac{A(L)}{G(L)H(L)} = \frac{\theta(L)}{\alpha(L)\phi(L)} \tag{92}$$

$$AO: \frac{A(L)}{G(L)H(L)} = 1 \tag{93}$$

$$TC: \frac{A(L)}{G(L)H(L)} = \frac{1}{(1-\delta L)} \tag{94}$$

$$LS: \frac{A(L)}{G(L)H(L)} = \frac{1}{(1-L)} \tag{95}$$

The four outliers represent various types of simple outlier effects; more complicated responses usually can be approximated by combinations of the four types. In principle, the proposed procedure can handle any other specific form of outlier responses.

It is remarkable that, except for the case of an IO, the effects of outliers on the observed series are independent of the model. Moreover, AO and LS are two boundary cases of TC, where $\delta = 1$ and $= 0$. For TC, the outlier produces an initial effect ω at time t_1 , and this effect dies out gradually with time. The parameter δ is designed to model the pace of dynamic dampening effect. In practice, the value of δ can be specified by the analyst. Chen & Lui recommend that $\delta = 0.7$ be used to identify a TC. In the case of an AO, the outliers cause an immediate and one-shot effect on the observed series. A LS produces an abrupt and permanent step change in the series. On the other hand, the effect of an IO is more intricate than the effects of the other types of outlier. Using the formulation of model (92), when IO occurs at $t = t_1$, the effect of this outlier on Y_{t_1+k}, for $k > 0$, is equal to $\omega\psi_k$, where ω is the initial effect and ψ_k is the kth coefficient of the $\psi\,(L)$ polynomial where

$$\psi\,(L) = \{\theta(L)\}/\{\alpha(L)\phi(L)\} \tag{96}$$
$$\psi\,(L) = (\psi_0 + \psi_1 L + \psi_2 L^2 + \cdots), \quad with\, \psi_0 = 1$$

For a stationary series, an IO will produce a temporary effect because ψ_j decay to 0 exponentially. The pattern of ψ_j for a nonstationary series can be quite different. Depending on the model of Y_t, an IO may produce (a) an initial effect at the time of the intervention and a level shift from the second period of intervention, if the time series model follows an autoregressive integrated moving average $ARIMA(0,1,1)$ model; (b) an initial effect at the time of intervention, gradually converging to a permanent level shift if Y_t follows an $ARIMA(1,1,1)$ model; (c) a seasonal level shift if the time series model follows a pure seasonal $ARIMA(0,1,1)_s$ and annual trend changes if Y_t follows a multiplicative seasonal $ARIMA(0,1,1)(0,1,1)_s$ model.

To examine the effects of outliers on the estimated residuals, Chen & Liu assume that the time series parameters are known and the series is observed from $t = -J$, $t = n$, wehre J is an integer larger than $p + d + q$, and that $1 \leq t_2 \leq n$, where p, d and q, are orders of the polynomials $\phi(L), \alpha(L)$ and $\theta(L)$. We define the $\pi(L)$ polynomial as

$$\pi\,(L) = \frac{\{\alpha(L)\ \phi(L)\}}{\{\theta(L)\}} = 1 - \pi_1 B - \pi_1 B^2 - \cdots \tag{97}$$

Where the π_j weights for j beyond a moderately large J become essentially equal to 0, because the roots of $\theta(L)$ are all outside the unit circle. The estimated residuals $\hat{e}_t$, which may be contaminated with outliers, can be expressed as:

$$\hat{e}_t = \pi(L) Y_t^*, \;\; for\ t = 1,2, \ldots \tag{98}$$

For the four types of outliers:

$$IO: \hat{e}_t = \omega I_t(t_1) + a_t \tag{99}$$

$$AO: \hat{e}_t = \omega\ \pi(L)\ I_t(t_1) + a_t \tag{100}$$

$$TC: \hat{e}_t = \omega\ \{\pi(L)/(\ 1 - \delta L)\} I_t(t_1) + a_t \tag{101}$$

$$LS: \hat{e}_t = \omega\ \{\pi(L)/(\ 1 - L)\} I_t(t_1) + a_t \tag{102}$$

Equation (99-102) can be alternatively written as:

$$\begin{gathered} \hat{e}_t = x_{it} + a_t \\ t = t_1, t_1 + 1, \ldots, n \\ i = 1,2,3,4 \end{gathered} \tag{103}$$

Where $x_{it} = 0$ for all I and $t < t_1, x_{it_1} = 1$ for all i and $k \geq 1$, $x_{1(t_{1+k})} = 0$, $x_{2(t_{1+k})} = -\,\pi_{k,}\ x_{3(t_{1+k})} = 1 - \sum_{j=1}^{k} \pi_j$, and $x_{4(t_{1+k})} = \delta^k - \sum_{j=1}^{k-1} \delta^{k-j} \pi_j - \pi_k$. Hence the least squares estimate for the effect of a single outlier at $t = t_1$ may be expressed as:

$$\widehat{\omega}_{IO}(t_1) = \hat{e}_{t_1} \tag{104}$$

$$\widehat{\omega}_{AO}(t_1) = \frac{\sum_{t=t_1}^{n} \hat{e}_t\, x_{2t}}{\sum_{t=t_1}^{n} x_{2t}^2} \tag{105}$$

$$\widehat{\omega}_{LS}(t_1) = \frac{\sum_{t=t_1}^{n} \hat{e}_t\, x_{3t}}{\sum_{t=t_1}^{n} x_{3t}^2} \tag{106}$$

$$\widehat{\omega}_{TC}(t_1) = \frac{\sum_{t=t_1}^{n} \hat{e}_t\, x_{4t}}{\sum_{t=t_1}^{n} x_{4t}^2} \tag{107}$$

It is important to note that for the last observation (i.e., $t_1 = n$) ,$\widehat{\omega}_{IO}(t_1) = \widehat{\omega}_{AO}(t_1) = \widehat{\omega}_{LS}(t_1) = \widehat{\omega}_{TC}(t_1) = \hat{e}_{t_1}$. As a result, it is impossible to empirically distinguish the type of an outlier occurring at the very end of a series. A possible approach for detecting outliers is to examine the maximum value of the standardized statistics of the outlier effects:

$$\hat{\tau}_{IO}(t_1) = \widehat{\omega}_{IO}(t_1)/\hat{\sigma}_a \quad (108)$$

$$\hat{\tau}_{AO}(t_1) = \{\widehat{\omega}_{AO}(t_1)/\hat{\sigma}_a\}\left(\sum_{t=t_1}^{n} x_{2t}^2\right)^{1/2} \quad (109)$$

$$\hat{\tau}_{LS}(t_1) = \{\widehat{\omega}_{LS}(t_1)/\hat{\sigma}_a\}\left(\sum_{t=t_1}^{n} x_{3t}^2\right)^{1/2} \quad (110)$$

$$\hat{\tau}_{TC}(t_1) = \{\widehat{\omega}_{TC}(t_1)/\hat{\sigma}_a\}\left(\sum_{t=t_1}^{n} x_{4t}^2\right)^{1/2} \quad (111)$$

For a given location, theses standardized statistics follow an approximately normal distribution. Knowing the type and location of an outlier, one can adjust the outlier effects on the observations and the residuals using Equation (92) and Equations (99-102). In general, the adjusted observations at t_1, denoted $\tilde{Y}_{t_1}$, can be expressed as a weighted sum of the entire observed series. In the case of IO, it can be shown that the adjusted observation $\tilde{Y}_{t_1}$, is the conditional expectation of $\tilde{Y}_{t_1}$m given the past observations. Under an AO, the adjusted observation is the interpolation based on both the past and the future $Y's$, but does not involve Y_{t_1}. This suggests a possible approach to estimating missing values in a time series by treating the missing data as an AO. Another important feature of the model is that they found that when the critical values are too low, there is a higher frequency to misidentify the location of an outlier by one time period.

When finding the outliers, the authors have also investigated the performance of the proposed procedure under certain non-Gaussian noise, such as noise with exponential distribution. In such situations, they found that the proposed procedure is effective in determining extreme values in a time series, but it cannot distinguish such extreme values as outliers or regular observations associated with the inherent nature of the distribution.

Following this approach described by Chen & Liu (1993), an automatic procedure for detection of outliers in time series is implemented in the package ***tsoutliers*** for *R*. The procedure may in turn be run along with the automatic ARIMA model selection strategy available in the package forecast.

The function ***tso*** is the main interface for the automatic procedure. Although the purpose of the package is to provide an automatic procedure, the implementation allows the user to do a manual inspection of each step of the procedure. Thus, the package is also useful to track the behavior of the procedure and come up with ideas for possible improvements or enhancements.

5.5 Forecast in the Supply Chain

Supply Chain Management is a difficult task due to the uncertainty associated with the demand. In addition, the frequency at which forecasts are produced varies considerably not only among the various supply chain organizations but also within each of those organizations depending on the decision-making processes they serve. Retail inventory replenishments, for example, rely upon frequent short-term forecast, whereas aggregate sales planning may take place quarterly.

The objective of every supply chain should be to maximize the overall value generated. The value (also known as supply chain surplus) a supply chain generates is the difference between what the final product is worth to the customer and the costs the supply chain incurs in filling the customer's requests. Such costs are an increasing function of the uncertainty associated with the demand and thus supply chain forecasting plays a major role in increasing the overall value.

The longer supply chains are and the more organizations they involve, the more difficult it becomes to coordinate them. The collaborative practices within supply chains vary considerably. There are three key features that have implications for supply chain forecasting:

(1) Under certain conditions, the variance of demand is amplified as it progresses upstream, making it more difficult to forecast accurately;

(2) There are potential gains in forecast accuracy which may be achieved by different forms of collaboration, including sharing of demand information between different levels of the supply chain;

(3) The practice of collaboration has resulted in some major initiatives like Collaborative Planning, Forecasting and Replenishment (CPFR) and Vendor Managed Inventory(VMI) systems that have had important implications for the practice of supply chain forecasting (Syntetos et al., 2016).

Syntetos at al. (2016) also developed a four-dimensional Supply Chain Structure within which supply chain forecasting hierarchies may be positioned, namely: Product, Time, Location and Echelon. The echelon dimension is necessary for any consideration of forecasting that relates to inventory management. The location dimension is also relevant to inventory management and is essential for any consideration of forecasts that inform transport planning. It is also crucial for inventory management/warehouse location decisions, as well as the decision to allocate given areas to different warehouses. The product dimension relates to inventory management, transport planning and also warehouse planning (e.g. where to locate products within a warehouse). Finally, the time dimension is essential for all forecasting problems, not just those that relate to supply chain forecasting (Figure 7).

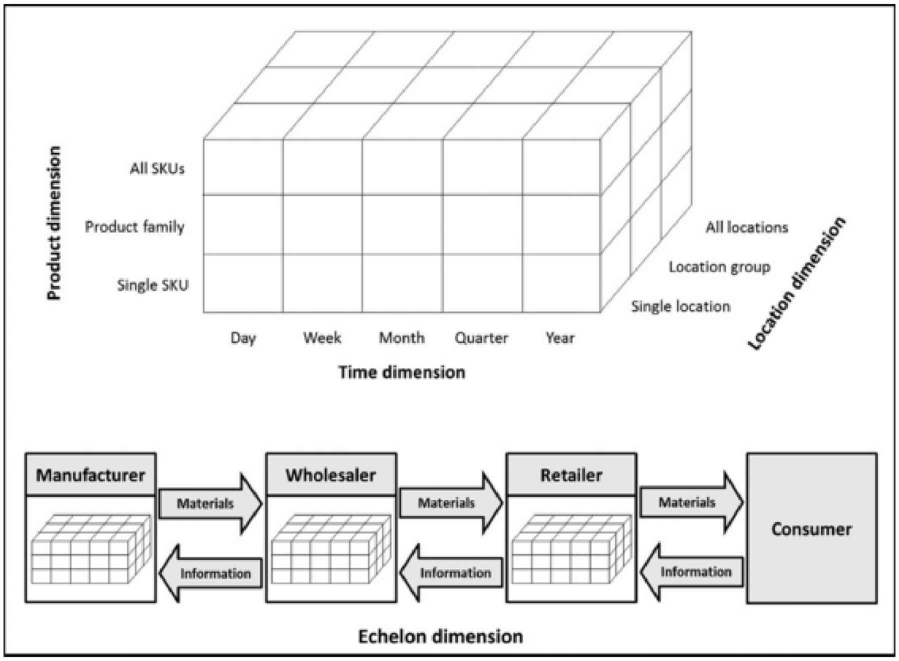


Figure 7. Supply Chain Structure (Syntetos et al., 2016)

5.5.1 The importance of demand planning

Most executives agree that the ability to generate an accurate forecast has a significant impact on long-term business success. The forecast directly affects an organization's ability to satisfy customers, manage resources and grow the business cost effectively. An improvement in forecast accuracy can have a ripple effect across the business, namely a 1% improvement of the forecast can lead to (Logility, 2017):

- 2.4% decrease in order-to-deliver days (cycle time)
- 0.4% increase in perfect order performance
- 2.7% reduction in finished goods inventory (days)
- 3.2% reduction in transportation cost (percentage of sales)
- 3.9% reduction in inventory obsolescence (percentage of inventory value)

There is also an equation which can be used to analyze the effect of an improvement in the forecast on the return on the Shareholder's value, this is called the Dupont Equation (Figure 8)

The Dupont Equation indicates that just a 10% increase in forecast accuracy can cause a Return on Shareholder Value increase up to 22.43%. Thence, the importance of improving forecast accuracy.

5.5.2 Improving forecast accuracy

According to the Consulting Group Logility (Logility, 2016), for many supply chain scenarios, it is typically best to employ a variety of methods to obtain optimal forecasts. Ideally, managers should take advantage of several different methods and build them into the foundation of the forecast.

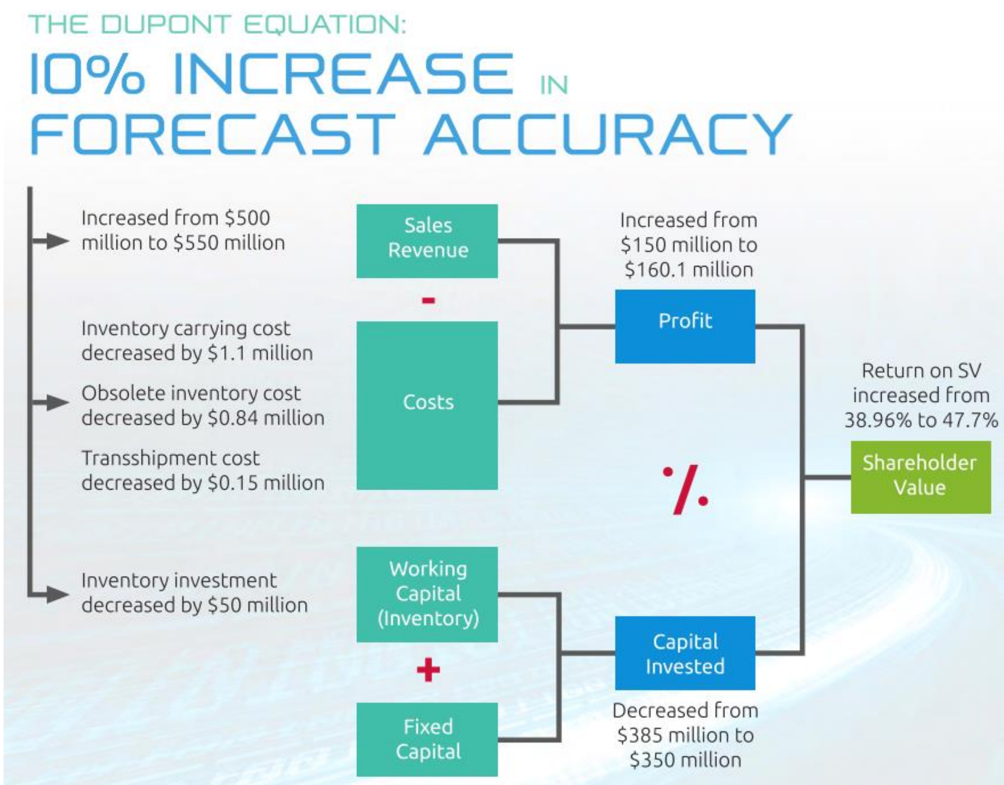


Figure 8. Dupont Equation Source: (Logility, 2017)

The best practice is to use automated methods switching to accommodate selection and deployment of the most appropriate forecast method for optimal results.

For most levels of management within an organization, aggregated demand history for product family, brand category, and/or selling region are good predictors of future performance. Such demand history serves as a baseline for effectively forecasting Stock Keeping Units (SKUs). When there are more than four-to-six periods of sales history, SKUs can be effectively forecast with moving average and basic trend methods. SKUs with at least one year of sales history offer sufficient information to incorporate a seasonal profile into the projected trend.

A modified Holt-Winters decomposition model with best-fit-analysis can generate forecasts based on demand history that incorporate trends and seasonal information. The method "senses" the amount of history available for each time series or segment to create a basic model that best fits the history. Then it uses the best combination of smoothing factors to enable the model to react to changing conditions going forward without overreacting to anomalies in demand (such as unplanned seasonal events, transportation disruption, and so on).

For factors relating to seasonality, planners need the ability to weight the historical demand. Under the assumption that the previous year is the best indicator of what will happen next year, most forecast systems apply a higher weighting factor to the previous year's demand, less to the year before and even less to the years before that. But if the previous year was unusual in any significant way, the planner must have the capability to change the historical weighting factors so as not to under- or over-forecast the business.

Seasonal methods can be effective with less than 23 months of history; the minimum required is twelve months. An effective approach for expected seasonal items with less than twelve months of history is to assign a seasonal curve that has been captured from a similar item or item group.

A powerful best-fit statistical method should include flexible features such as trend, seasonal-with-trend, moving average and low-level pattern fitting, as well as trend models for products with sporadic, low-volume demand. The method should provide limiting and damping, as well as seasonal smoothing, demand filtering, reasonability tests, tracking signals and test for erratic nature that evaluate the validity of each element, determining which are anomalous and should be filtered. These parameters give the planner the flexibility to tune the process to best fit conditions at any element of the organization (Table 3).

Table 3. *Parameters for an effective forecast*

Parameter	Description
Limiting	Confidence Limits describe the spread of the distribution above and below the point forecast.
Smoothing	Removes random variation (noise) from the historical demand, enabling better identification of demand patterns (primarily trend and seasonality patterns) and demand levels. Result in a closer estimate of future demand.
Damping	Applies various "weights" to each period to achieve the desired results. These weights are expressed as percentages. Total weights must add up to 100%.
Filtering	Forecast error, viewed as the difference between forecast value and actual value, is usually normally distributed. A demand filter is usually set to +/- 4 Mean Absolute Deviation against the forecast value. Whenever the deviation is more than that, the adequateness of the forecast model should be reviewed by analyzing the actual data.
Forecast Error	The difference between actual demand and forecast demand. Error can occur in two ways: bias or random variation. Bias is a systematic error that occurs when cumulative actual demand is consistently above or below the cumulative forecast demand. Type 1 Bias is subjective and occurs due to human intervention. Type 2 Bias is a manifestation of a business process that is specific to the product (for instance, persistent demand trend and forecast adjustments do not correct fast enough for items specific to a few customers).
Reasonability Tests	An important type of reasonability measure is the tracking signal, which can be used to monitor the quality of the forecast. There are several procedures that can be used, but one of the simplest is based on a comparison of the cumulative sum of the forecast errors to the mean absolute deviation (cusum).

Source: (Logility, 2016)

The best fit refers then to the ability to change forecast methods as a product evolves through its life cycle. The process may start out as a demand profile method (this is a technique which uses user-defined attributes to model new product introductions and product end-of-life retirement), evolve to a modified Holt-Winters method as the product becomes stable, and ultimately transition to a demand profile method again as the product life cycle comes to an end.

6. CURRENT FORECAST PROBLEM ANALYSIS

6.1 Material Flow Forecast Description

As mentioned in the problem description, the problem concerns an international automotive company which produces vehicles and other products in 23 assembly plants around the world and has more than 32 logistics service providers. The company has currently more than 4000 suppliers which deliver different vehicles parts to their corresponding consolidation center in their forwarding area. To be precise, the company has an area forwarding based inbound logistics network.

The current problem regarding the supply chain structure framework can be described as follows (Syntetos et al., 2016): (1) *at the product dimension level:* the forecasts regard all finished goods or components flows aggregated as tons (2) *at the location dimension:* it is about all main leg flows from the forwarding areas to the plants in Europe, (3) *at the time dimension:* forecasts are made monthly, and finally (4) *at the echelon dimension:* the supply chain level corresponds to the flows among the consolidation centers (source) and the plants (sink) (see Figure 1).

471 flows were initially considered into the forecasting process designed by the company. The data available corresponds to the time frame January 2014 to the current ongoing date; however, there are some flows which do not have information between 2014 and 2015. Moreover, there are 5 methods already implemented, i.e. Naïve, Auto ARIMA, Neural Networks, Exponential Smoothing and Ensemble Forecasts. The first four methods are available in the package *forecast* based on R-programming, which allows to make an automatic implementation.

At this time, a 4-step-ahead forecast is carried out and forecasts are monthly upgraded, this methodology allows the company to reduce the uncertainty implicit in long term forecasts.

It is also important to mention that the forecasts are not made directly using the time series with its values in tons but with another variable called alpha α, which is the result of dividing the time series in tons over the production plan time series of the corresponding plant for which this main leg material flow is forwarded. This allows to take advantage of an existing high correlation between these two variables and carry out a time series normalization to reduce the variability of the observations. This solution implemented by the company is also another way to reduce the uncertainty due to current adjustments to the data bases which are being made for the synchronization of the information systems and the human error. Then, given the forecast for the Production Plans, which are provided by other department, the **α-time series** can be set back to tons by multiplying it by its corresponding production plan forecast.

Nevertheless, there are some flows which do not deliver material to the plants but to some consolidation centers owned by the company, where the material here is then forwarded to plants outside Europe. Since there are not production plans for these consolidation centers, forecasts for these flows have not been considered yet.

Along this research the terms **α-time series** and **ton-time series** will be used referring in the first case, the ton-time series divided by its corresponding production plan and in the second case, the ton-time series regarding the time series without any transformation, i.e. in tons.

6.2 Time Series Analysis

6.2.1 Demand Patterns

The flows display almost all possible demand patterns, i.e. positive and negative trends, seasonality, irregular demand, outliers, and missing values, except intermittent demand. One particular feature of the problem is the availability of two time series referring to the same variable. That is to say, the tons per month, forwarded from the main legs forwarding areas to the plants, register two values from the same variable, one is called the "*actual_ton*" the other is called "*should_ton*". This is done to avoid data bases inconsistency which is mostly caused by human error or lacking synchronization among the information systems. The *actual_ton* time series is the one being used to make the forecasts; however, when missing values or outliers are found, the corresponding value in the *should_ton* time series is used instead. Figure 9 depicts a time series example.

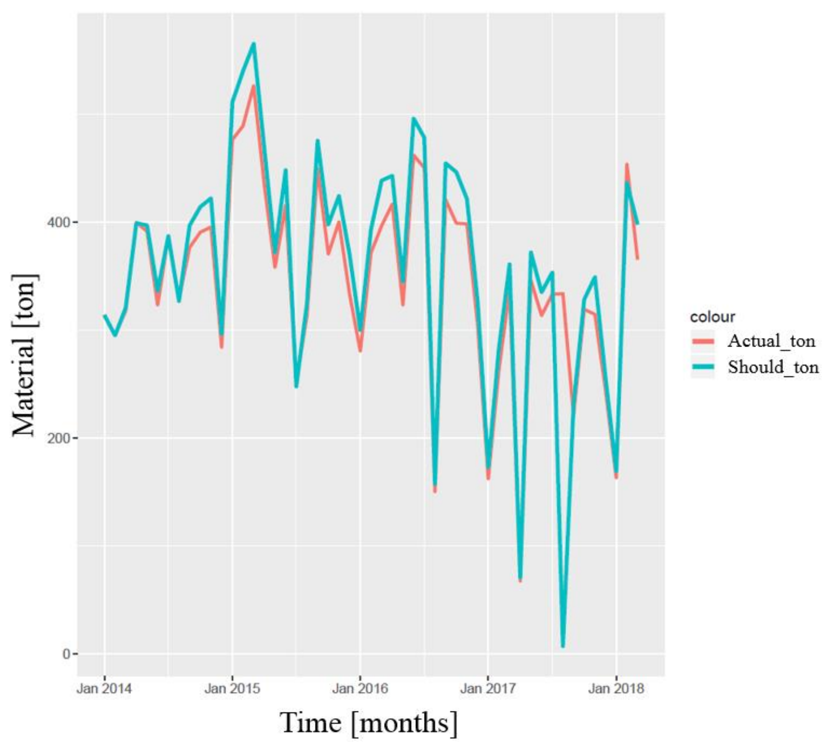


Figure 9. Example of Actual and Should ton for a Material Flow Time Series

6.2.2 Outliers

The outlier detection method currently used by the company's software is described as follows: Values exceeding a threshold (5 times the Interquartile Range plus the 3^{rd} quantile) in both the *actual_ton* and *should_ton* time series should be replaced by the median of the corresponding observations. Then, a validation function is used to clean the data, namely, if a value in the *actual_ton* time series is found to be an outlier then it is replaced by its corresponding value in the *should_ton* time series; as long as this value itself is not an outlier;

if so is the case, i.e. both the values of a given month for the *actual_ton* and the *should_ton* time series are outliers, then the median of the *actual_ton* time series is used instead.

6.3 Current Error Measures Analysis

The software uses an out-of-sample forecast measure based on the Mean Absolute Percentage Deviation (MAPE). Specifically, the time series are split up into subsets, one subset is used to make the model fitting, and the other to make validation tests.

6.4 Current Best Error Measures Evaluation Decision

Given the resulting MAPE out of the fitting model process for each time series for each model; the model with the lowest MAPE is chosen to make the 4-step-ahead forecast.

In a more detailed way, the choosing process is carried out as follows: the time series are split up into samples to make cross-validation tests, then the Mean Absolute Percentage Error (MAPE) is calculated for each validation test to make a 4-month-ahead forecast. For example, if a time series has n values, then the process will be so: initially the first $k < n$ values are taken, and the corresponding model is fitted, then a 4-month-ahead forecast is carried out, and with this forecast, the MAPE for this validation test is calculated. Then $k + 1$ values are taken as the training set and the same 4-month-ahead forecast is made. This step will produce up to $n - k - 4$ cross-validation tests and MAPE's, which are then averaged, giving the decision variable to choose the best model.

The same process is made for the other 4 methods, which also delivers their corresponding cross-validation test averaged MAPE. This methodology is called ***rolling forecasting***, because the software is trying to simulate the data generation behavior over time. This is certainly useful, since forecast automated methods are being used by R; specifically, every time the forecast is made, new parameters for the models are calculated. So the idea behind this, is to let the software decide which automated method behaves the best along the data generation process by considering the averaged MAPE of all cross-validation tests (Logility, 2016).

6.5 Current Implemented Software Description

The original state of the software can be summarized in Figure 10.

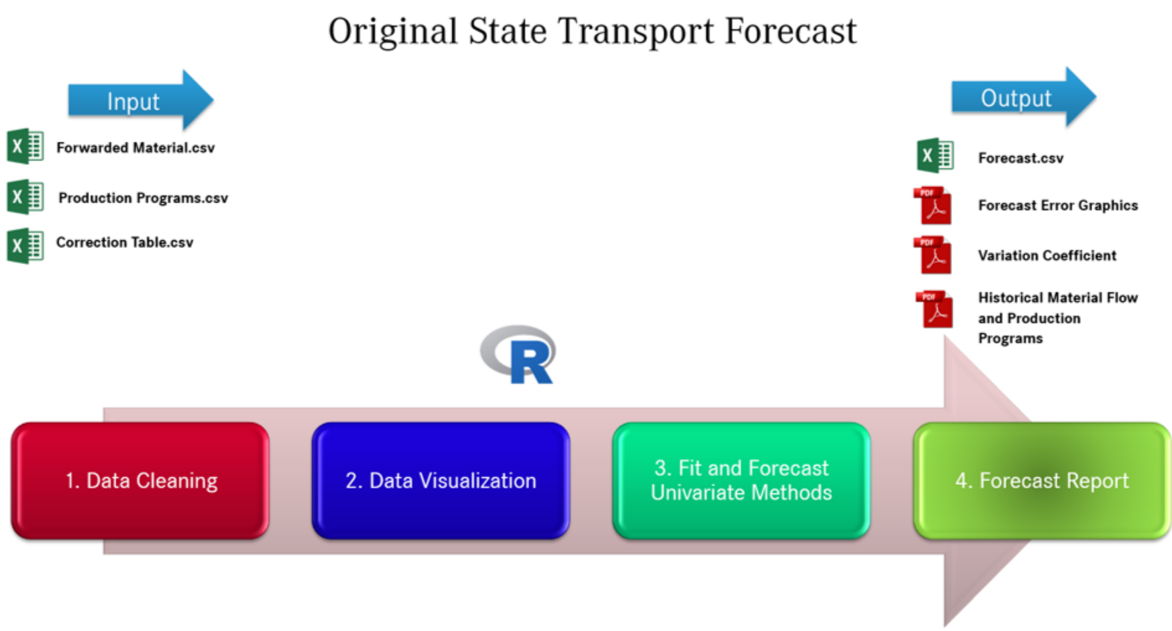


Figure 10. Original State Transport Forecast Software

6.5.1 Input Data

In order to obtain the forecasts, certain input is needed. There are three csv files which trigger the process:

(1) **Historical Forwarded Material**, which is upgraded monthly and contains the historical forwarded material flow information since 2014. This data base can be described as follows:

- 1: Time in months (date)
- 2: Logistics Service Provider Identification Number (string)
- 3: Logistics Service Provider Name (string)
- 4: Bordereau Prefix Number (Integer)
- 5: Plant Identification Number (string)
- 6: Transport mode (string)
- 7: Inbound or Outbound Transport (string: Inbound/Outbound)
- 8: Actual_kg (Integer)
- 9: Should_kg (Integer)

(2) The second file is the **Production Programs**, which contains the units which were produced and will be produced until some point in the future. Each column represents a plant, and its values are integer values. The (3) last file is a **Correction Table**, which have some negative material flow values for some months in the past, which then are added to the historical values to adjust values which still have not been corrected in the main Material Flow data base. This file has the same data structure as the first csv file, except that its Actual_kg and Should_kg values are negative.

6.5.2 Output Data

The main output file is the Forecast file as a csv file, this document delivers the following information:

- 1: Numeration (integer)
- 2: Logistics Service Provider Identification Number (string)
- 3: Logistics Service Provider Name (string)
- 4: Plant Identification Number (string)
- 5: Plant Name (string)
- 6: Current Month actual ton (double)
- 7: 1-month ahead forecast in ton (double)
- 8: 2-month ahead forecast in ton (double)
- 9: 3-month ahead forecast in ton (double)
- 10: 4-month ahead forecast in ton (double)
- 11: Current Month absolute percentage (string) – always 100%
- 12: 1-month ahead forecast in percentage relative to current month (string)
- 13: 2-month ahead forecast in percentage relative to current month (string)
- 14: 3-month ahead forecast in percentage relative to current month (string)
- 15: 4-month ahead forecast in percentage relative to current month (string)
- 16: Forecast Error classified as: low $(< 20\%)$, medium $(between\ 20\ ; 30\%)$ or high (> 30%) (string)

The second file corresponds to the **Forecast Error Graphics**, which shows the Mean Absolute Percentage Deviation (MAPE) from each of the main leg material flows for each logistics service providers and all the plants to which it delivers material. Below this graphic, the corresponding average monthly material flow is also plotted, this allows the analyst to check the relationship between the MAPE and the average monthly material flow for each main leg. An example is shown (Figure 11).

The third file shows the **Historical Incoming Material Flow for each plant and the corresponding Production Plan** (Figure 12). The most important information of this analysis is to realize the high existing correlation mentioned before between these two variables; i.e. the production program is indeed the trigger of the quantity of input materials purchased by a company. The analyst uses this graphics to find out whether any unusual behavior from any of those two variables might have raised. That is the reason why the forecasts are made using the $\alpha -$ transformation, instead of using the tons themselves. Further, an experiment will be carried out on this matter, and it will be demonstrated that the MAPE worsen if the forecasts are made directly using the tons instead of the α time series.

The last output file is the **Variation Coefficient Analysis for the Material Flows** (Figure 13), which allows the analyst to recognize if there is any uncommon variation in the data; a low variation coefficient should be found on material flows with high monthly material volumes, i.e. the uncertainty of the forecast should also be lower and additional high costs can be avoided; whereas for a high variation coefficient, low monthly material volumes should be the rule; that means, even though the variability is high, the additional costs should be low since there is not a great amount of material compromised.

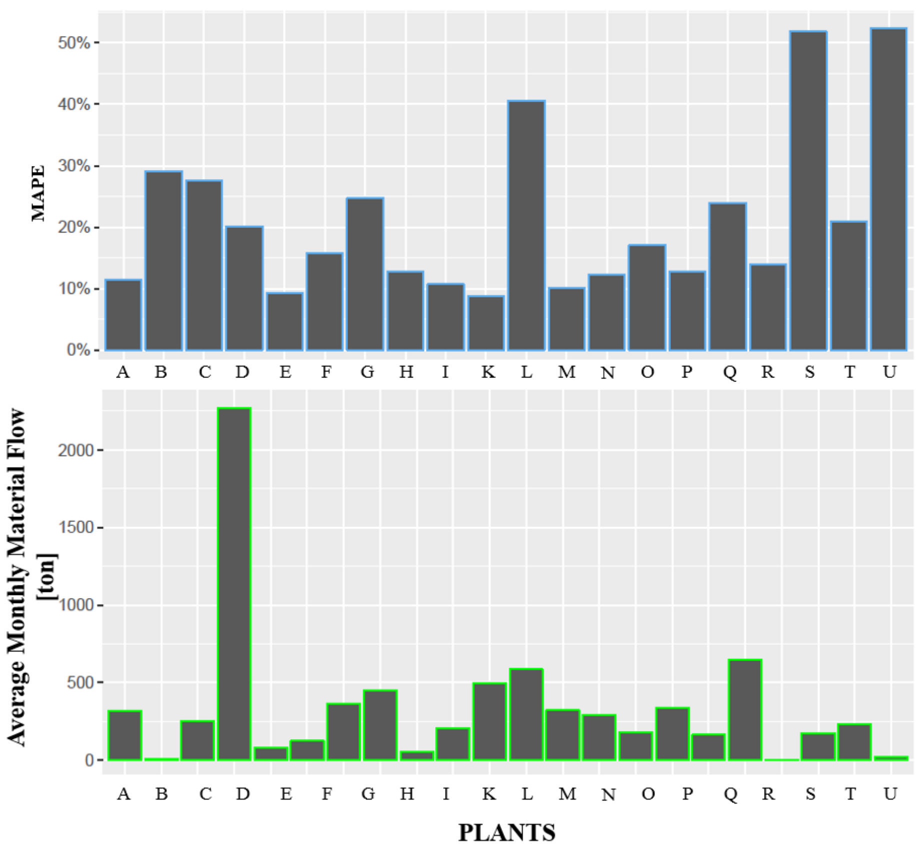


Figure 11. Forecast Error (MAPE) and Monthly Average Material Flow for a single Logistics Service Provider

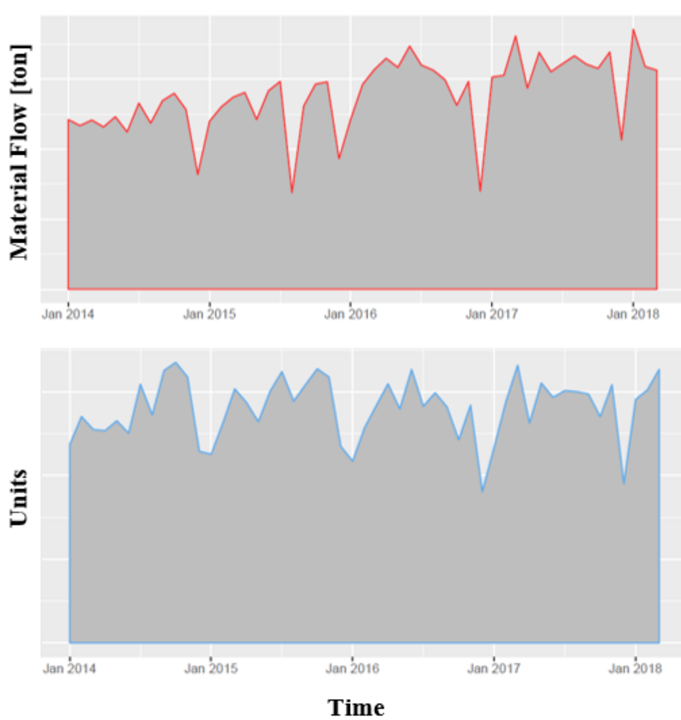


Figure 12. Material Flow and Production Program Analysis for a single plant

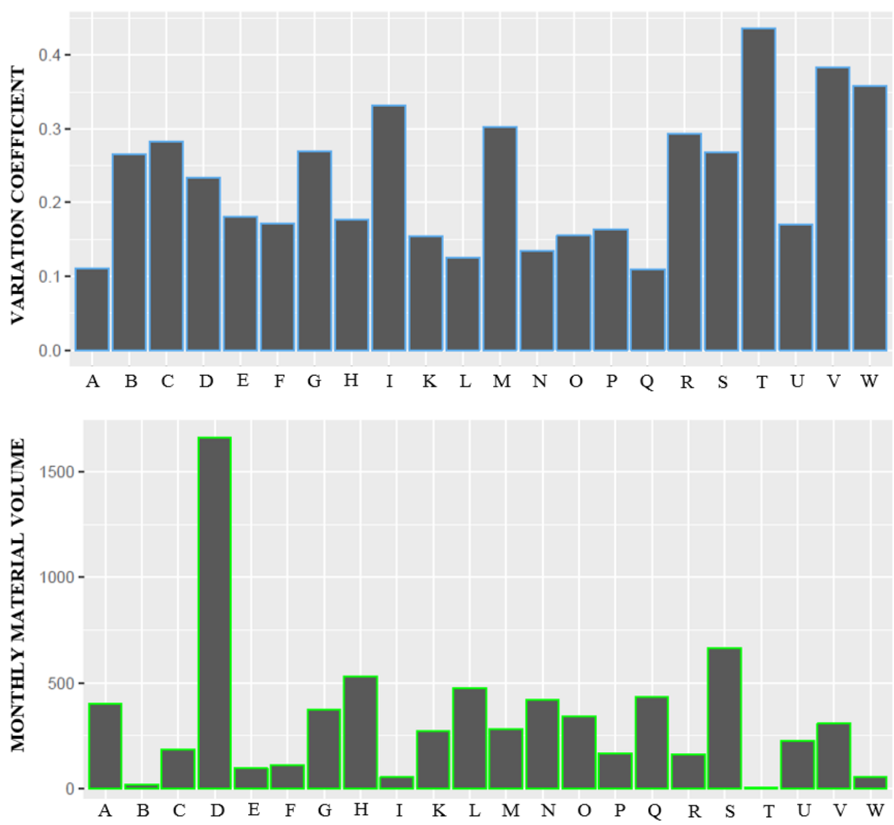


Figure 13. Variation Coefficient and Monthly Material Volume for a single Logistics Provider

6.5.3 R Scripts

The R-Scripts are the single steps of code shown in Figure 10. The first R-Script called **Data Cleaning** performs different activities which allow the user to adapt the data before the forecasting fitting process is carried out. The Correction Table file is used as input to eliminate some wrong values in the historical material flows. The outliers are afterwards eliminated using the outlier detection procedure described before.

The second R-Script called **Data Visualization** allows the user to plot the Variation Coefficient with the Monthly Material Volume for every single Logistics Provider as well as the Material Flow and Production Program Analysis for every single plant. As mentioned before, these graphics are used to recognize unusual behavior in the data.

The third R-Script named **Fit and Forecast Univariate Methods** carries out firstly the forecasting fitting process, i.e. the cross-validation tests. As mentioned in 6.4, the data is split up into subsets for training and test, then the average MAPE is the criterion to choose which method better fits the data. Secondly, with the selected methods, a 4-month-ahead forecast is carried out for every single main leg material flow.

The last step is the R-Script **Forecast Report**, which takes the results from the 3rd R-Script and turn it into a csv file report called Forecast with the respectively current date, as mentioned in 6.5.2.

6.6 Potential Improvement Points

6.6.1 Triple for loop

The fit and forecast univariate methods R-Script performs its processes by using a triple for loop, this is the so called cubic algorithm. To have solutions in the order of minutes, triple algorithms are only typically capable of solving problems on the size of thousands, while quadratic algorithms can solve problems in the order of tens of thousands (Sedgewick & Wayne, 2011). This triple algorithm causes the forecast process to take a long time, approximately 1 hour, on a 4 GB RAM, Intel Core i5 4310 2.7 GHz Laptop. Thence, the analyst had used so far, a workstation with 64 GB RAM and an Intel Xeon E5-2699 2-processors 2.30 GHz with 72 cores, which is able to make this step in 5 minutes. Nevertheless, the workstation is not always available, which causes delays on the Forecast Report when it comes to make analysis tests. A possible solution would be to try at least to turn the triple for loop into a double for loop in order to achieve a quadratic algorithm.

6.6.2 Historical MAPE

Currently, the forecast fitting process and the forecast calculation are carried out in a single R-Script. This fitting process has been repeatedly carried out every single month, which also causes the software to take up so much computer time. A possible solution is to split up these two steps, namely, to make the fitting process and the forecasting in different R-Scripts. This idea would allow to save the resulting averaged MAPE's from the fitting process in a separate file in order to reuse it for future forecasts; allowing the software to focus only on the forecast calculation.

6.6.3 Choosing between competing models

As stated before, the decision criterion to select which method better fits the data is the MAPE. Other out-of-sample error measure as MSE or RMSE could be used to choose between competing models and evaluate the model's variability.

6.6.4 Implementation of new forecasting methods

There are forecasting methods which could be tested that are likely to outperform the results of the current methods. It is important to select methods which can model and fit the demand patterns described before, namely positive and negative trends, seasonality, irregular demand, outliers, and missing values.

One good example is the Prophet Algorithm. There are two main features that this algorithm displays which might bring positive results:

(1) Parameters can easily accommodate seasonality with multiple periods and let the analyst make different assumptions about trends.
(2) Unlike ARIMA models, the measurements do not need to be regularly spaced, and missing values do not need to be interpolated *e.g.* from removing outliers.

These characteristics makes this new released algorithm a good option to implement (Taylor & Letham, 2017).

Additionally, there are also important features which are left out when only using univariate methods, which is the current case for the company's software, since the current used methods are ARIMA, Neural Network, Naïve, Exponential Smoothing and Ensemble Forecasts. On the other hand, Multivariate Methods are able to consider lag-cross correlations among different time series (Tsay, 2014). This cross-correlation feature along the historical data regards the influence of a past value of a time series A on the future value of a time series B and vice versa; this feature is highly relevant, since if the parameters for this influence can be found and are statistically significant, then it may allow to improve the forecast of groups of time series with high autocorrelation values.

Finally, a simple but useful method still not considered by the company is the Simple Moving Average. (Logility, 2016) mentions that the Moving Average is the best model for products whose demand histories have random variations, including no seasonality or trend, or fairly flat demand. Therefore, it is relevant to implement this method, considering an automation which allow to find the optimal parameters automatically.

Furthermore, the Ensemble Forecast method, already implemented by the company, which consider a simple linear combination (average) of the forecast values from the other methods, can be also extended, i.e. the Prophet Algorithm, Simple Moving Average and Multivariate Time Series can also be included in the linear combination, so that the likelihood of better forecasts accuracy increase (Smith & Wallis, 2009).

Finally, another possibility is to implement an algorithm which can complete the missing values in the time series. Recently, a new data base update to the main Material Software Server eliminated values in some periods. This leads to incomplete time series. Allowing a linear interpolation algorithm to find the missing values instead of using the mean of the observations can also improve the forecasting accuracy. Linear interpolation is easy to implement (Lepot et al., 2017), which will allow to find missing values for the 471 time series in short computer time. (Gnauck, 2004) demonstrated that this method is efficient, and most of the time it is better than non-linear interpolations for predicting missing values.

7. RESULTS

7.1 Forecast Accuracy Improvement

Three new forecast algorithms were implemented, namely, Prophet Algorithm, Automated Simple Moving Average and Multivariate Time Series: Vector Auto-regressive. For testing the performance of the algorithms, the current software's approach was implemented; i.e. the rolling forecasting method (see 6.4) with the averaged MAPE, so that the same criteria could be used in order to compare how better the new methods behave regarding the current ones. Moreover, a forecast test using the ton-time series instead of the α-time series was also realized.

Two types of tests are carried out. They are called:

(1) **Test type 1**, in which 3 plants and their respective logistics service providers are chosen, reducing the amount of material flows to forecast from 472[3] to 84. This approach has the advantage of faster computer time calculation and also a briefly overview of how the algorithm performs in a data subset.

(2) **Test type 2,** in which all plants and their respective logistics service providers are chosen, i.e. the complete 472 material flows. This test takes up a long computer time but allows to see what the overall performance of the algorithm is.

Then two types of files were developed to assess the performance:

(1) **Algorithm_name_difference.csv** shows how many of the material flow forecasts achieved a better MAPE with the new algorithm. The data file has the following structure:
- Column 1: Numeration (integer)
- Column 2: Logistics Service Provider Identification Number (String)
- Column 3: Plant Identification Number (String)
- Column 4: MAPE difference (double) – Percentage of MAPE improvement
- Column 5: Old_Method (String)
- Column 6: New_Method (String)

(2) **Comparison Empirical Distribution MAPE as pdf**, which shows how the cumulative percentage of MAPE's are spread along the number of material flows forecasted. The x-axis represents the number of cumulative material flows as percentage. The y-axis represents the cumulative percentage of MAPE's. Two lines will be shown, the red one stands for the new method's performance, the blue one, stands for the old method's performance. This graphic can be interpreted as follows: if the new forecast method has a better performance, then the red curve must be directed either in the upper direction, to the left or in both directions. This interpretation means that more material flow forecasts have a lower MAPE than before.

An important observation in most of the below forecasting methods is that almost never the new forecasting method worsens the MAPE. In other words, the method either improves the MAPE or it remains unchanged, but it does not worsen.

[3] There was one new available material flow at the time where the tests were carried out. That is why, the tests are made with 472 material flows instead of 471.

7.1.1 Prophet forecast Implementation

As described in section 5.1.4.3, the Prophet Algorithm is a new forecast method, which can be implemented through a package based on R. The algorithm can be fully automated, so that the parameters selection is based on routines available in the R-package *Prophet*. It uses different parameters like yearly, weekly and daily seasonality. In order to test the feasibility of the algorithm, the two types of tests were carried out.

7.1.1.1 Test type 1: Prophet

The first test, which analyzes only 3 plants, improved 4 flows out of 84. The results are presented in Table 4.

Table 4. *Prophet Algorithm, Test Type 1*

Numeration	MAPE Improvement (%)	Old_Method	New_Method
1	4.39	Ensemble Forecast	Prophet Forecast
2	4.31	Exponential Smoothing Forecast	Prophet Forecast
3	3.43	Exponential Smoothing Forecast	Prophet Forecast
4	2.73	Naive Forecast	Prophet Forecast

This represents an improvement of 3.72% on the averaged MAPE's, and of 4.72% on the number of material flow forecasts.

The corresponding Empirical Distribution Function is shown in Figure 14.

7.1.1.2 Test type 2: Prophet

The second test, which considers all plants, delivered 43 improved flows out of 472. A briefly view of the results are presented in Table 5.

Table 5. *Prophet Algorithm, Test Type* 2

Numeration	MAPE Improvement (%)	Old_Method	New_Method
1	24.31	Naive Forecast	Prophet Forecast
2	21.53	Ensemble Forecast	Prophet Forecast
3	19.46	Exponential Smoothing Forecast	Prophet Forecast
4	18.22	Ensemble Forecast	Prophet Forecast
5	16.40	Exponential Smoothing Forecast	Prophet Forecast
6	15.74	Ensemble Forecast	Prophet Forecast
7	15.19	Naive Forecast	Prophet Forecast
8	13.77	Ensemble Forecast	Prophet Forecast
9	13.10	Ensemble Forecast	Prophet Forecast
10	12.11	Ensemble Forecast	Prophet Forecast

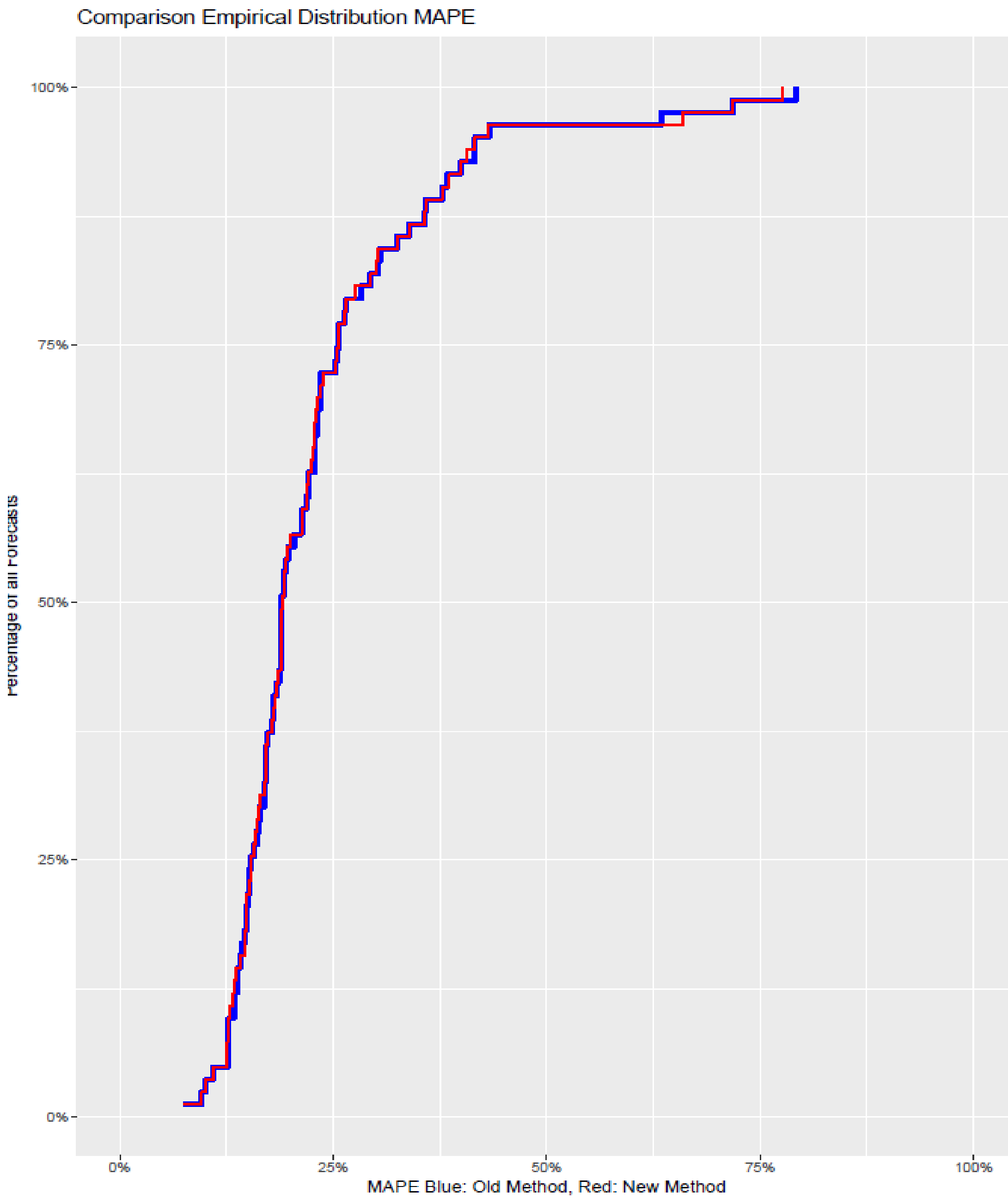


Figure 14. Comparison Cumulative MAPE Distribution Function Prophet Algorithm, Test Type 1

The Test Type 2 for the Prophet Algorithm represents an improvement of 8.31% on the averaged MAPE's, and of 9% on the number of material flow forecasts.

The corresponding Empirical Distribution Function is shown in Figure 15.

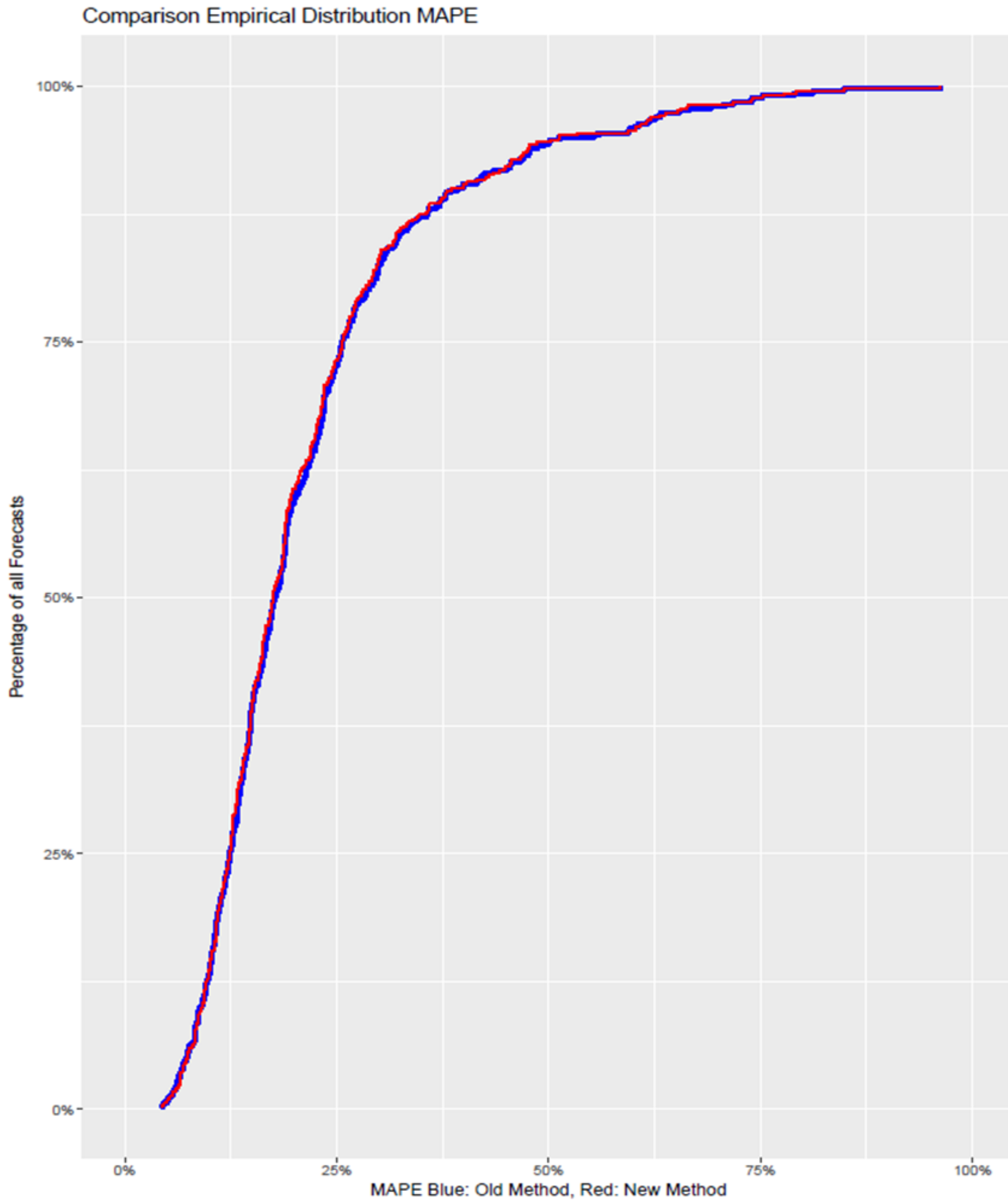


Figure 15. Comparison Cumulative MAPE Distribution Function Prophet Algorithm, Test Type 2

7.1.2 Automated Simple Moving Average

The function SMA from the R-Package *Smooth* applies the Simple Moving Average method on a time series vector. The SMA order was set to be chosen automatically by the function, which chooses the optimal one. In order to test the performance of the algorithm the Tests Type 1 and 2 were made.

7.1.2.1 Test Type 1: Automated Simple Moving Average

The Test type 1, which considers only 3 plants, improved 20 flows out of 84. The results are presented in Table 6.

Table 6. *Automated Moving Average, Test Type 1*

Numeration	MAPE Improvement (%)	Old_Method	New_Method
1	6.73	Ensemble Forecast	Simple Moving Average
2	6.3	Ensemble Forecast	Simple Moving Average
3	5.96	Ensemble Forecast	Simple Moving Average
4	5.86	ARIMA Forecast	Simple Moving Average
5	5.25	Exponential Smoothing Forecast	Simple Moving Average
7	4.11	Exponential Smoothing Forecast	Simple Moving Average
9	2.94	Exponential Smoothing Forecast	Simple Moving Average
10	2.83	Ensemble Forecast	Simple Moving Average
12	2.06	Ensemble Forecast	Simple Moving Average
13	1.98	Ensemble Forecast	Simple Moving Average
14	1.85	Exponential Smoothing Forecast	Simple Moving Average
15	1.85	Ensemble Forecast	Simple Moving Average
16	1.58	Exponential Smoothing Forecast	Simple Moving Average
17	1.39	Exponential Smoothing Forecast	Simple Moving Average
18	0.99	Exponential Smoothing Forecast	Simple Moving Average
19	0.17	Naive Forecast	Simple Moving Average
20	0.09	Exponential Smoothing Forecast	Simple Moving Average

This result represents an improvement of 3.12% in the averaged MAPE's, and of 23.8% on the number of material flow forecasts.

The corresponding Empirical Distribution Function is shown in Figure 16.

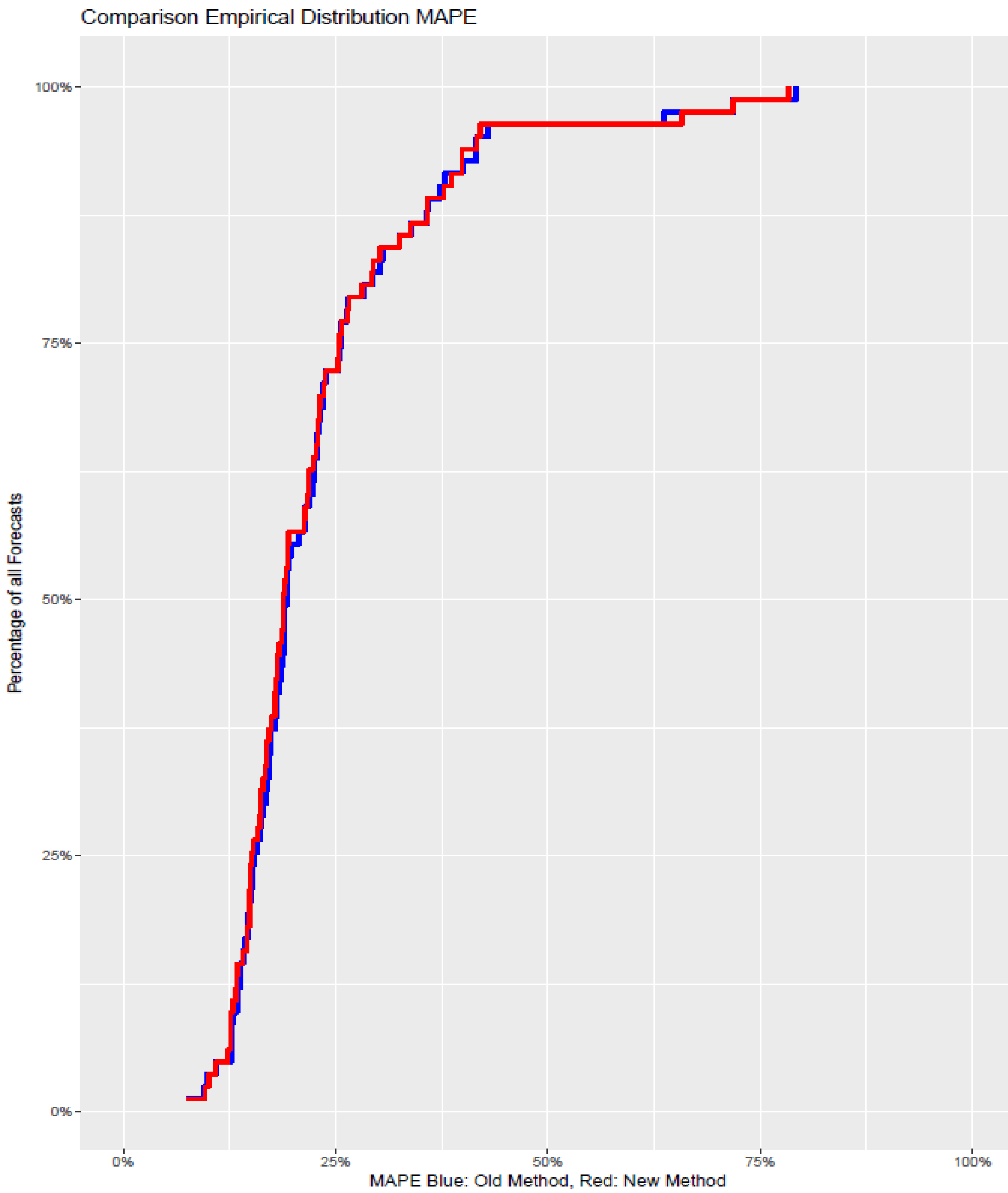


Figure 16. Comparison Cumulative MAPE Distribution Function Automated Moving Average, Test Type 1

7.1.2.2 Test Type 2: Automated Simple Moving Average

The Test Type 2, which considers all plants, improved 80 flows out of 472. A briefly summary of the results is presented in Table 7.

Table 7. *Automated Moving Average, Test Type 2*

Numeration	MAPE Improvement (%)	Old_Method	New_Method
1	21.34	Exponential Smoothing Forecast	Simple Moving Average
2	13.98	Ensemble Forecast	Simple Moving Average
3	12.92	Ensemble Forecast	Simple Moving Average
4	9.89	Naive Forecast	Simple Moving Average
5	9.10	Ensemble Forecast	Simple Moving Average
6	8.53	Exponential Smoothing Forecast	Simple Moving Average
7	8.44	ARIMA Forecast	Simple Moving Average
8	8.42	Ensemble Forecast	Simple Moving Average
9	8.18	Exponential Smoothing Forecast	Simple Moving Average
10	7.97	ARIMA Forecast	Simple Moving Average
11	7.84	Naive Forecast	Simple Moving Average
12	7.18	Naive Forecast	Simple Moving Average
13	7.12	Ensemble Forecast	Simple Moving Average
14	7.11	Ensemble Forecast	Simple Moving Average
15	6.98	Exponential Smoothing Forecast	Simple Moving Average
16	6.95	Ensemble Forecast	Simple Moving Average

This result represents an improvement of 3.88% on the averaged MAPE's, and of 17% on the number of material flow forecasts.

The corresponding Empirical Distribution Function is shown in Figure 17.

7.1.3 Multivariate Time Series Implementation

For the implementation of the Vector Autoregressive method, there are a couple of things which must be considered in advance. First, The Ljung-Box test is used to test the lag-cross-correlation along n time series (see 5.1.2.8). Since there are 472 material flows, the possible groups would be $472! = 7\ x\ 10^{1058}$. This is a huge number, which could not be solved in a reasonable computational time. That is why, the time series are divided into groups which are more likely to have the highest lag-cross-correlation coefficient, and namely, all the material flows coming to a single plant.

A plant has up to 32 incoming flows, this value represents the number of logistics service providers which currently deliver material to the plants. Secondly, the automated Vector Autoregressive method might break down if too many time series with too few values are calculated, explicitly, the algorithm takes up too much memory and time to calculate all the parameters involved in the matrices. Moreover, the number of lags consider fitting the model also affects the algorithm performance, that is why a **1-lagged Automated Vector Autoregressive** model is implemented in this case.

Finally, another routine is implemented to eliminate the parameters with a significance level lower than 5%. This step improves the model accuracy as well as the final forecasts errors. To sum up, the multivariate time series implemented through the Automated Vector Autoregressive model is shown in Figure 18.

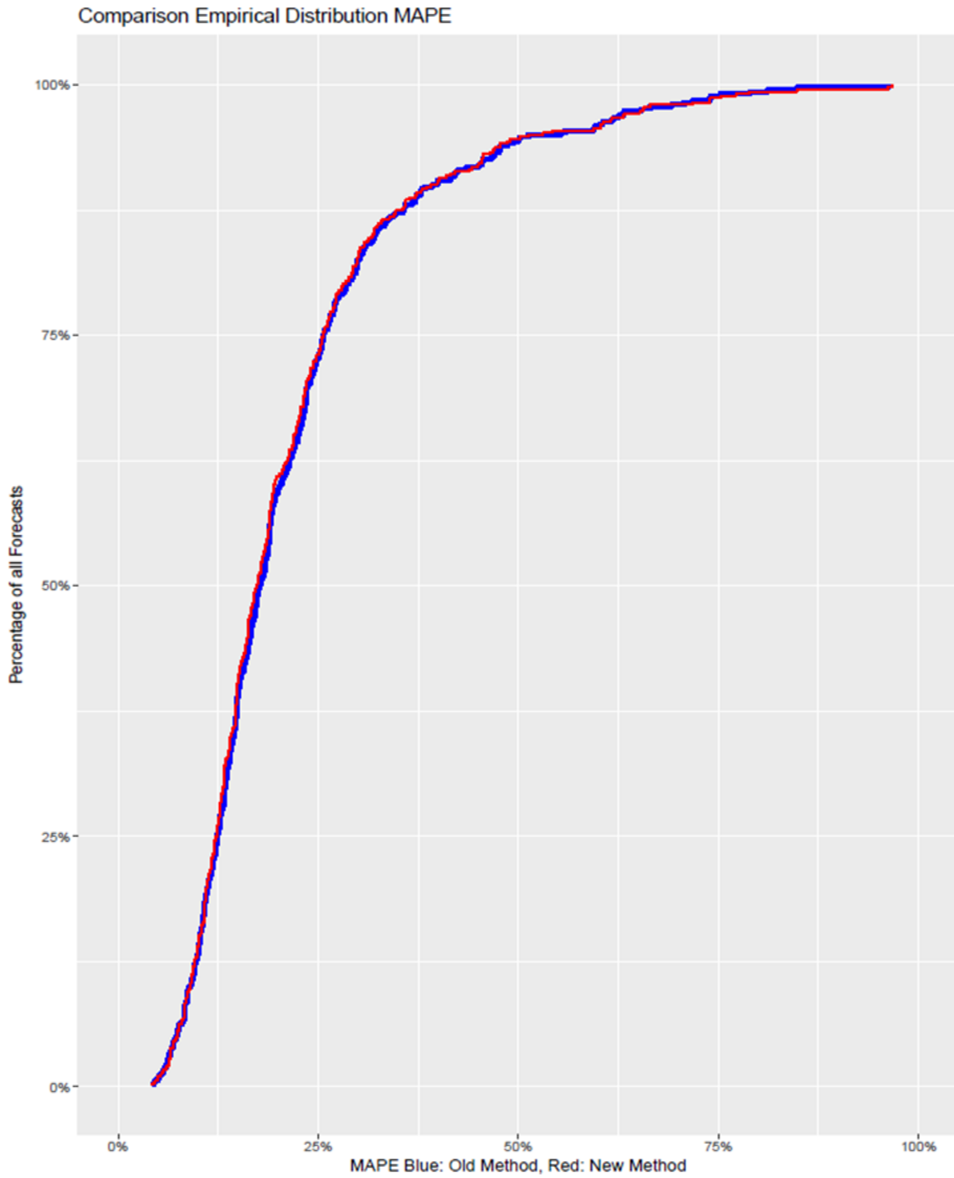


Figure 17. Comparison Cumulative MAPE Distribution Function Automated Moving Average, Test Type 2

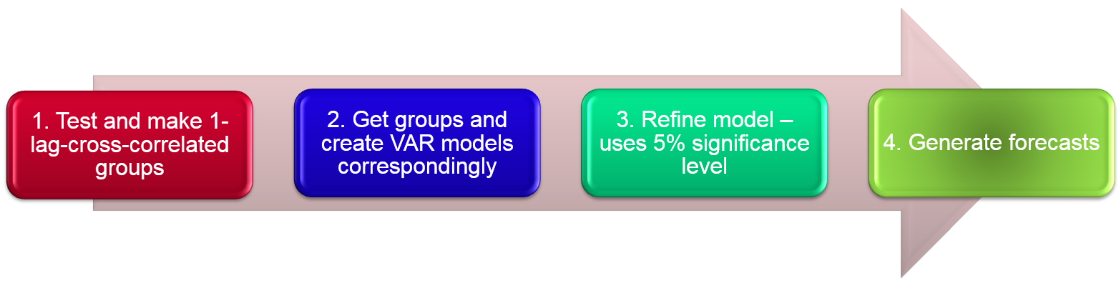


Figure 18. Multivariate Time Series Implementation

The first step in figure 18 takes, for example, 5 flows and makes the possible combinations of groups which have two things: on the one hand, that the time series in the group have a 1-lag-cross-correlation; on the other hand, that the higher possible number of time series joins in this group. These two constrains make sure that the groups obtained do not follow a factorial function. After that, step two creates the VAR models, step three refine them, and step four generates the 4-month-ahead forecasts.

The results of the implementation can be briefly summarized in Table 8. For this algorithm, only the Test Type 2 was carried out due to the complexity of the implementation.

Table 8. *Multivariate Time Series, Test Type 2*

Numeration	MAPE Improvement (%)	Old_Method	New_Method
1	68.64	Exponential Smoothing Forecast	Multivariate Time series
2	59.13	Naive Forecast	Multivariate Time series
3	57.58	Naive Forecast	Multivariate Time series
4	52.96	Neural Network Forecast	Multivariate Time series
5	46.28	Naive Forecast	Multivariate Time series
6	39.06	Prophet Forecast	Multivariate Time series
7	37.96	Naive Forecast	Multivariate Time series
8	33.75	Simple Moving Average	Multivariate Time series
9	33.23	Ensemble Forecast	Multivariate Time series
10	27.54	Ensemble Forecast	Multivariate Time series
11	26.40	Ensemble Forecast	Multivariate Time series
12	24.78	Naive Forecast	Multivariate Time series
13	24.02	Exponential Smoothing Forecast	Multivariate Time series
14	22.15	Exponential Smoothing Forecast	Multivariate Time series
15	21.85	Ensemble Forecast	Multivariate Time series

The Vector Autoregressive model improved 32 out of 472 material flows, representing an improvement of 6.8%. The averaged MAPE was improved in 23.78% (Figure 19). This implies that the VAR model may not improve as many flows as the SMA algorithm, but the accuracy of this improvement is higher than both Prophet and SMA algorithm.

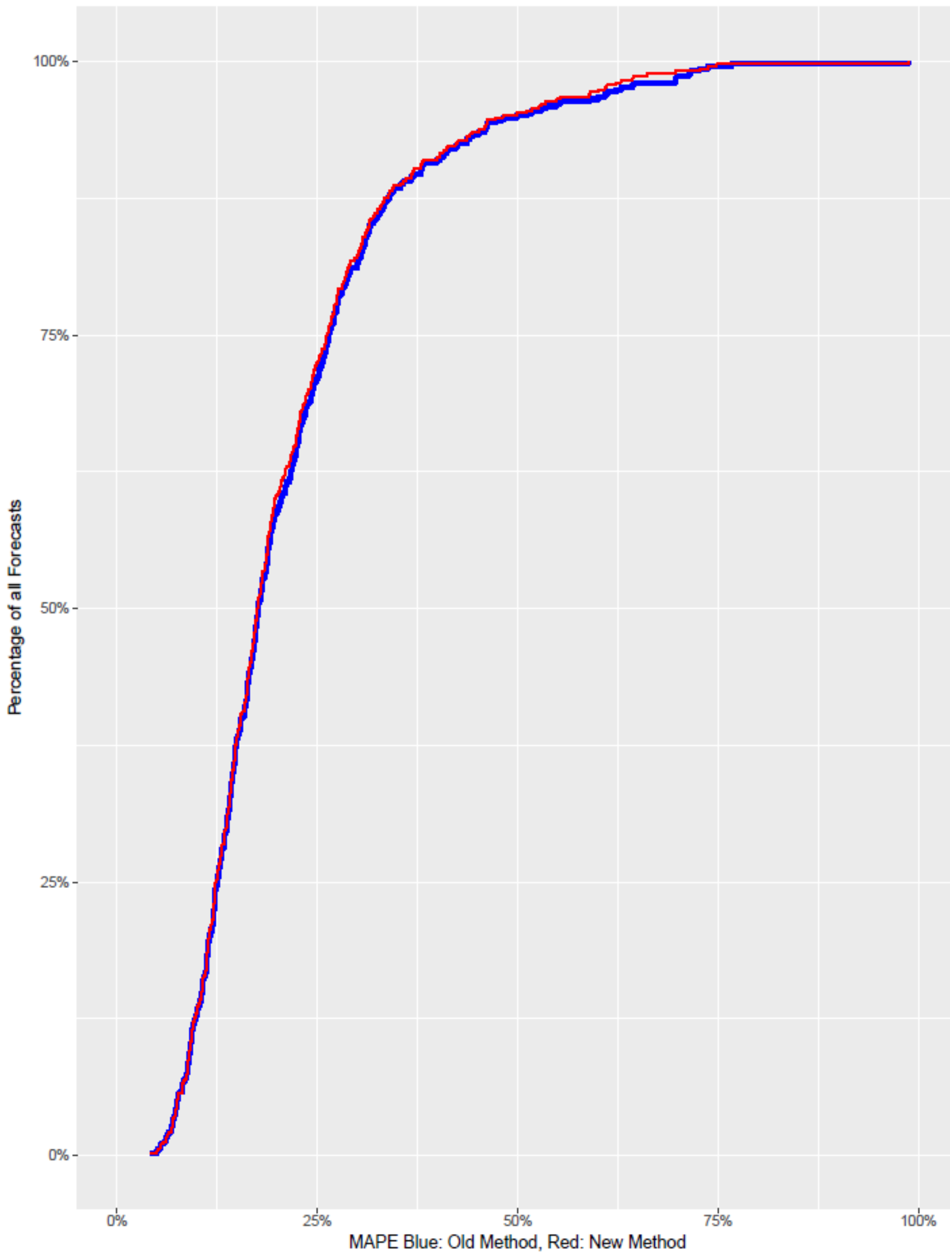


Figure 19. Comparison Cumulative MAPE Distribution Function Multivariate Time Series, Test Type 2

7.1.4 Prophet, SMA and Multivariate Time Series Combination

Finally, all the three algorithms implemented were tested together, the results are presented in Table 9.

Table 9. *Prophet, SMA and Multivariate Algorithms Performance Summary*

Algorithms	Averaged MAPE Improvement	Improved Material Flows	Stand
Prophet Algorithms	8.31 %	43 out of 472 (9%)	January 2018
Automated Simple Moving Average	3.78 %	82 out of 472 (17.4%)	January 2018
Multivariate Time Series	23.78 %	32 out of 472 (6.8%)	March 2018
Prophet + SMA + MVTS	7.82 %	201 out of 472 (42.8%)	March 2018

Table 9 shows how the algorithms performed individually and combined. It is important to highlight that the Multivariate Time Series algorithm obtained the highest Averaged MAPE improvement, namely, 23.8%, which tells about the potential of finding lag cross-correlation among different time series. Therefore, it can be said that, there is indeed many time series' past values which are correlated with future values of other time series, and that the decision of making times series groups of those that go to the same plant, delivered a significant improvement. Moreover, the Automated Simple Moving Average was able to improve more time series than the other two algorithms, representing a 17.4% of the total time series considered. Finally, the Prophet algorithm also had a significant contribution, above all, because the averaged MAPE improvement was higher than the improvement made by the Automated Simple Moving Average.

When the three algorithms are combined, dynamical relationships appear. That is to say, there should be maximum 157 improved material flows (the sum of the individual improvements), but their respective new forecasts are now included in the Ensemble Forecast, i.e. Ensemble Forecast now considers the average of the forecasts delivered by ARIMA, Neural Networks, Exponential Smoothing Methods, Naïve, Prophet, SMA and Multivariate. This new combination leads to an overall improvement of 201 material flows out of 472, representing the 42.8% of the forecasts. As already stated by (Logility, 2017), just an improvement of 10% of forecasting accuracy can reduce inventory costs up to 10%, in this particular case, the average MAPE improvement was 7,82%, this could reduce inventory costs up to 7%.

7.1.5 Ton-based Forecast

As mentioned before, the forecasts are carried out by using a variable transformation. This allows to take advantage of an existing high correlation between these two variables and carry out a time series normalization to reduce the variability of the observations. The values are originally in tons, but due to the high correlation between the material flows and the production programs, a variable transformation is realized, namely:

$$\alpha_t = \frac{Material\ Flow\ [ton]}{Production\ Program\ [Units]} \quad for\ t = given\ month \tag{112}$$

The forecasts are then made using the values delivered by this new time series, also called α-time series, and multiplied by the future production programs. In order to test if this transformation, implemented initially by the company, better performs than using the time series in tons, Test Type 1 and 2 were carried out. Therefore, in this case the software was adjusted so that the forecasts were made in tons.

7.1.5.1 Test Type 1: Ton-based Forecast

The Test Type 1 delivers the performance comparison based on the material flows to three plants. Figure 20 depicts the results of this test. It can be found out that the forecasts performance worsened considerably when the ton-time series are used instead of the α time series.

7.1.5.2 Test Type 2: Ton-based Forecast

Test Type 2 reinforces the results delivered by Test Type 1. Therefore, the α time series have a better out-of-sample MAPE performance than the ton time series (Figure 21). That is why, all future adjustments to the software should be done for the forecast based on the α transformation.

The variability reduction reached by this transformation also explains the data inaccuracy of the Bill of Materials' data bases. This is the problem currently found owing to the company's data bases' synchronization.

7.5 Time Series Data Cleaning

7.5.1 Outliers detection

The outlier detection routines were analyzed, and it was found out that, a simple Interquartile-Range-based Method for the outlier detection was not enough to clean the data time series correctly and completely. This is so because this method considered statistical data, which only includes correlation terms; however, time series models consider lag-autocorrelations values which are also important when it comes to detecting outliers in time series. Thence, the method proposed by (Chen & Liu, 1993) was implemented. This method is capable of fitting a corresponding ARIMA model to every single time series, and then it applies a structure, which is described in 5.4, that looks for 4 different types of outliers: An Innovational Outlier (IO), an Additive Outlier (AO), a Level Shift (LS), and a Temporary Change (TC). Depending on the type of outlier, interpolation or the median of the time series is used to replace the values.

Figure 22 shows the results of the implementation. The algorithm improved the data quality, that is to say, the step 1 in Figure 10: the data cleaning. These results allowed the model to deliver better forecasts. The forecasts were carried out namely using the first three new algorithms: Prophet, Vector Autoregressive and Automated Simple Moving Average. The number of improved material flow forecasts were 314 out of 473, it means an improvement of 66.4%, whereas the average improvement of the MAPE was 22.25%. This is a much better result than when only the three new algorithms were implemented. This result shows how important the quality of the input data for the forecasting process is. Just by adjusting the outliers of the time series, the overall improved material flows forecasts went from 42.8%, as stated in Table 9, to 66.4%. This represents a 23.6 p.p. improvement. Additionally, the forecast accuracy improvement went from 7.82% to 22.25%, representing a 14.4 p.p. improvement. Therefore, improving the data quality itself can lead to higher forecast accuracy. The current result now represents an Inventory Cost reduction up to 20%.

For this algorithm only the Test Type 2 was carried out (Figure 22). A flow chart, which explains how the algorithm's routine was implemented for this particular case, is found in Figure 23.

It is also important to mention here that an automatic deletion or replacement of outliers is better preceded by an analysis of the causes of the outlier. For example, if the cause is human error when entering data, the outlier should be immediately replaced. However, another cause might be a real change in demand trend. Automatic replacement could hide real changes in demand trends.

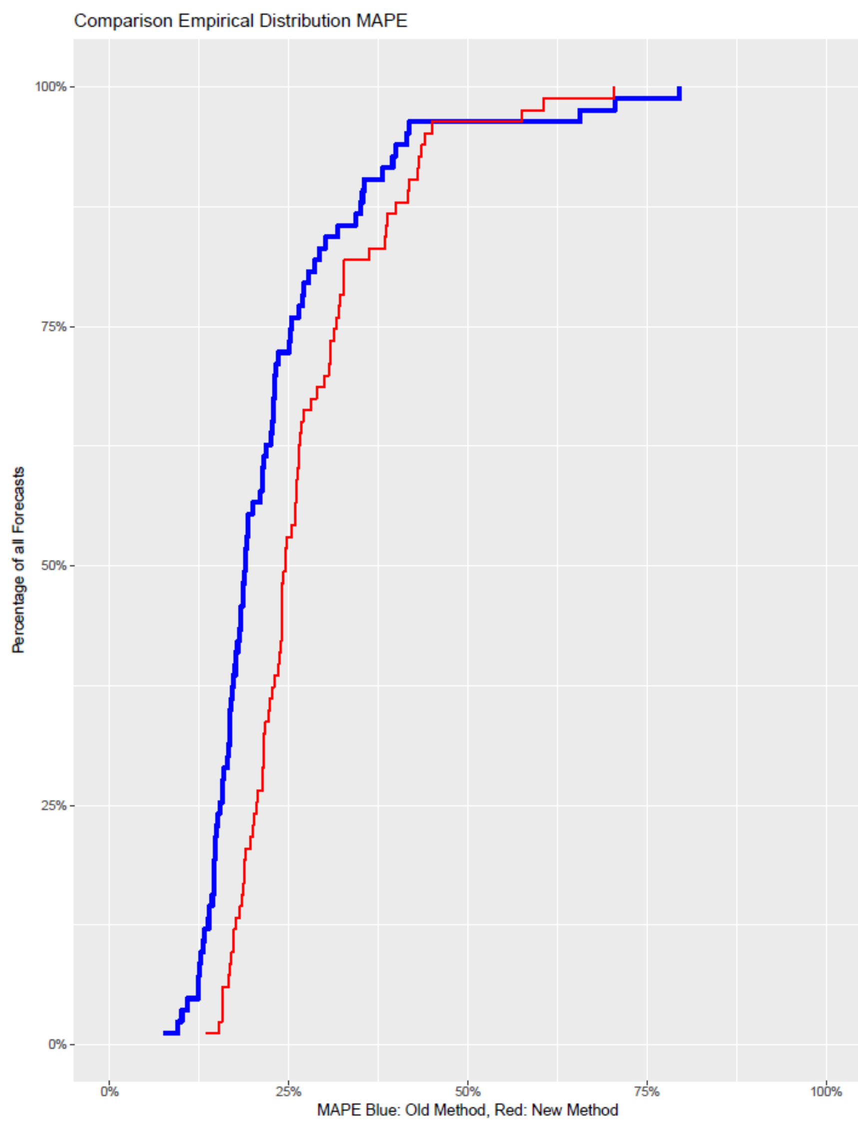


Figure 20. Comparison Cumulative MAPE Distribution Function Ton-Based Forecast, Test Type 1

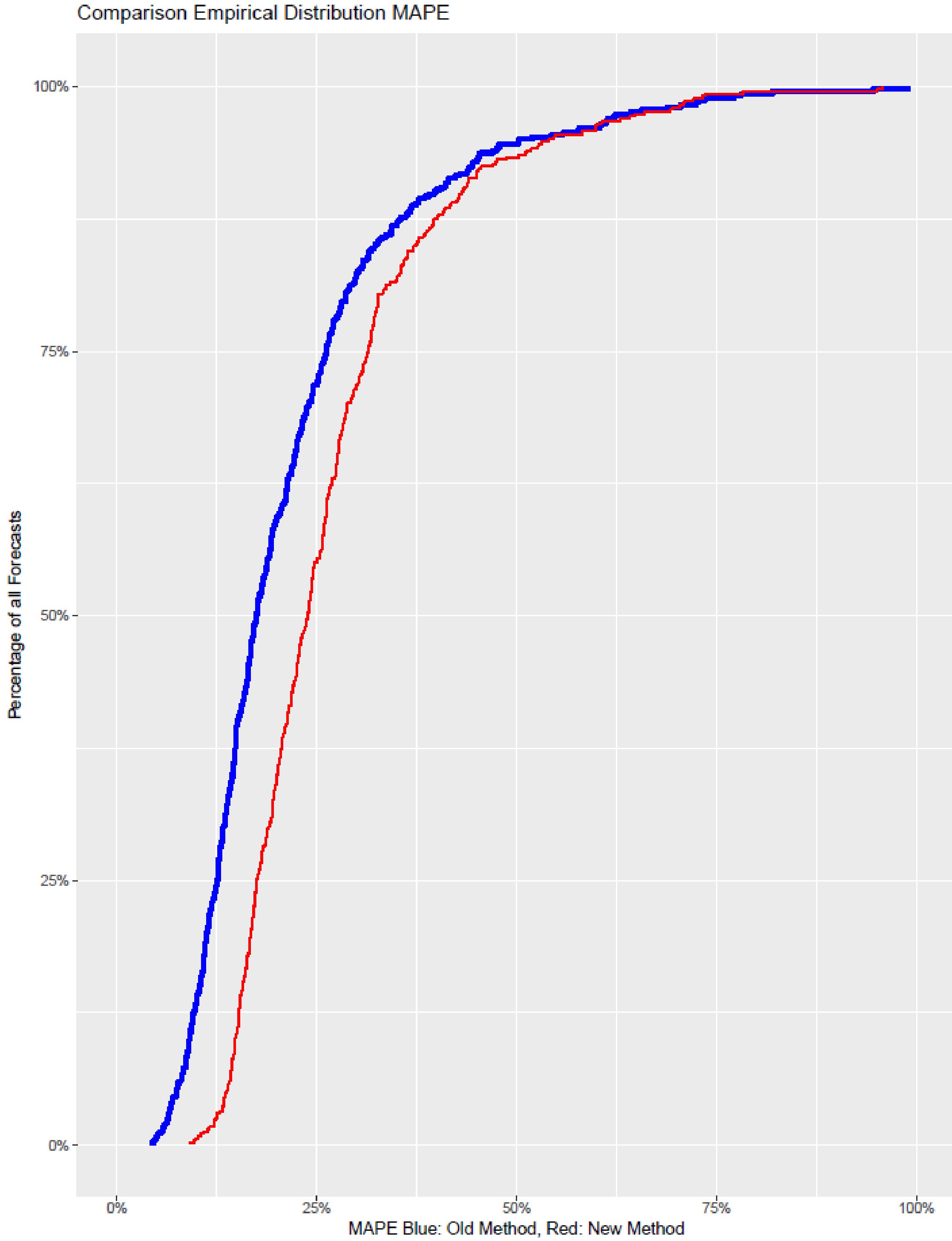


Figure 21. Comparison Cumulative MAPE Distribution Function Ton-Based Forecast, Test Type 2

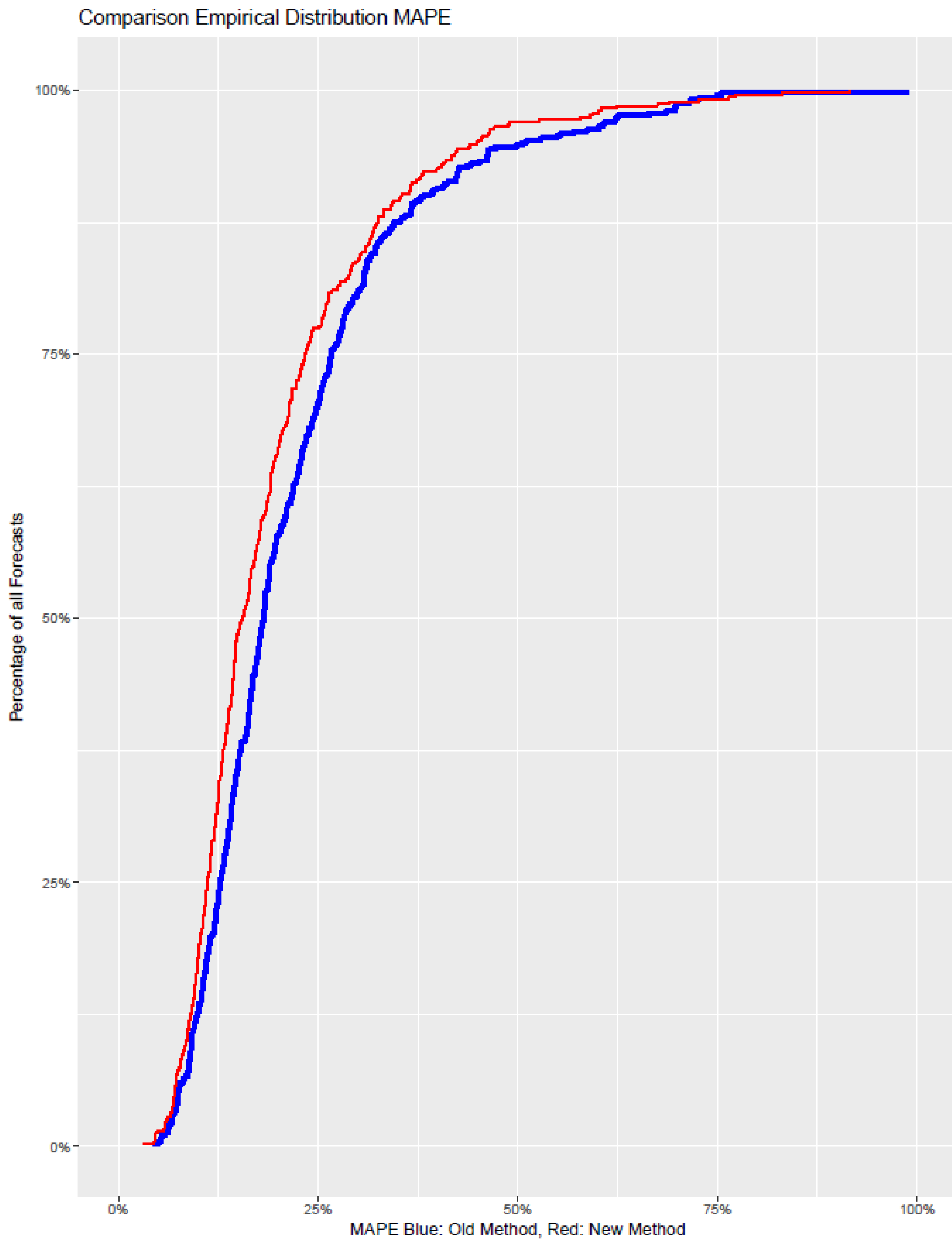


Figure 22. Comparison Cumulative MAPE Distribution Function Outlier Detection Algorithm, Test Type 2

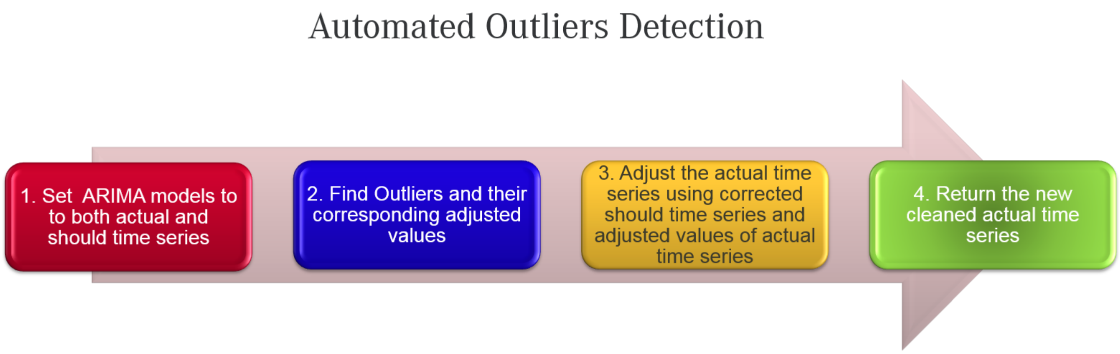


Figure 23. Automated Outliers Detection Flow Chart

Due to the particular data available, namely the two time series Actual_ton and Should_ton, it is important to consider both when replacing outliers. The process starts at setting ARIMA models to both the actual and the should time series, then the automated outlier detection method is applied to these ARIMA models, which delivers the outliers and their corresponding adjusted value. Then the actual time series is adjusted using both the corrected should time series and its adjusted values. This is done by using a simple decision rule similar to the one used for the Interquartile Range Method. The rule is if a value in the actual time series is an outlier, then replace it by the corresponding value in the should time series, as long as this value is not itself an outlier; if the value of a month is an outlier for both time series then the adjusted value, found by the automated outliers' detection method, is used instead. Finally, the new adjusted actual time series is returned, and this will be the one used to make the forecasts.

7.5.2 Linear Interpolation

As stated before, improving the data quality leads to improvement of the forecasts' accuracy. Given that the outlier detection algorithm only works for complete time series, there are some information loss for material flows having missing values. This is so because the information before a missing value is not considered by the forecasting process. The time frame considered in this analysis is the monthly material flows between January 2014 and April 2018. Missing values frequently appear between years 2014 and 2015.

A linear interpolation algorithm was used in order to find the missing values. Then a routine was implemented to complete all the missing values found in every single material flow time series.

This new implementation was added to the software combining the Prophet Algorithm, the Automated Simple Moving Average, the Multivariate Time Series and the outlier detection procedure. Table 10 summarizes all the results obtained with the implemented algorithms. The very last implementation is then the linear interpolation which allows the final software, compared to the original software, to improve 325 material flows out of 481[4] representing now the 67.6% of all material flows. Moreover, the averaged MAPE was also improved to 24.84%. Therefore, better data quality leads to better forecasts. This can be implied from the results,

[4] As of April 2018, there are new material flows having more than 10 monthly values, which is the requirement for a material flow time series to be forecasted.

given that the averaged MAPE improvement by the new algorithm was 7.82%, whereas the outlier detection procedure and the interpolation allowed this improvement to rise on 17.82 p.p. This is an extremely important result, since a merely improvement in data quality, i.e. the Bill of Material's data bases information accuracy, has the potential to reduce Inventory Costs up to 17%. Finally, the Empirical Distribution of the MAPE's is depicted in Figure 24.

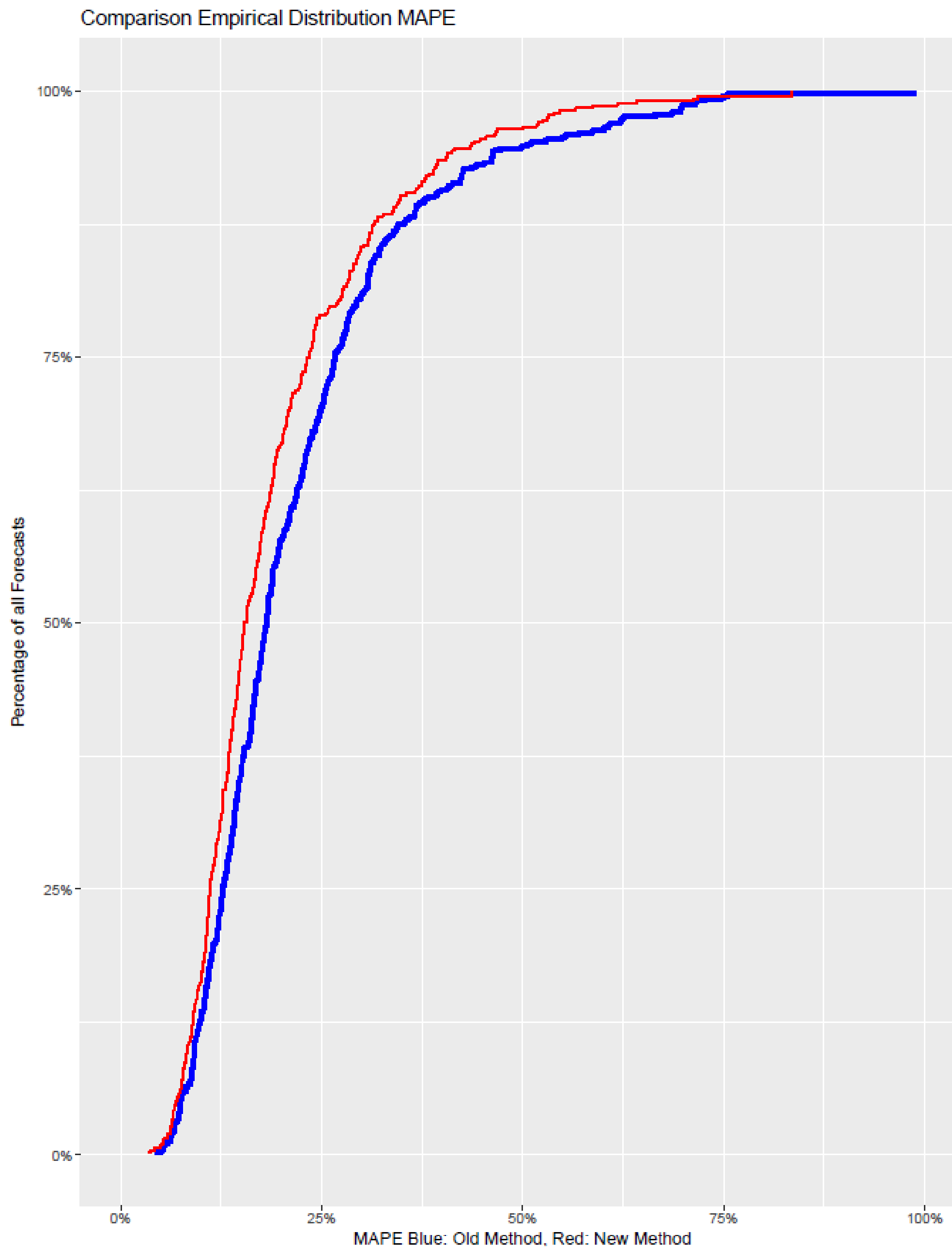


Figure 24. Comparison Cumulative MAPE Distribution Function Interpolation Algorithm, Test Type 2

Table 10. *All Algorithms Performance Summary*

Algorithms	Averaged MAPE Improvement	Improved Material Flows	Stand
Prophet Algorithms	8.31 %	43 out of 472 (9%)	January 2018
Automated Simple Moving Average	3.78 %	82 out of 472 (17.4%)	January 2018
Multivariate Time Series	23.78 %	32 out of 472 (6.8%)	March 2018
Prophet + SMA + MVTS	7.82 %	201 out of 472 (42.8%)	March 2018
P + SMA + MVTS + Outliers **detection**	22.25%	314 out of 473 (66.4%)	March 2018
P + SMA + MVTS + Outliers detection + **Interpolation**	24.84%	325 out of 481 (67.6%)	April 2018

7.6 Run Time Improvements

7.6.1 For-loop Indexing

In the R-Script Fit and Forecast Univariate Methods there was initially an indexing error in the most internal for-loop, which is the one in charge of making the cross-validation tests for every time series. The indexing was beyond the length of the training data set for each time series, adding a false cross-validation out-of-sample MAPE at the end of the training process. The overall averaged MAPE for each test was then either under or overestimated. Thence, the indexing problem was adjusted correspondently, allowing the software to deliver reliable averaged MAPE's from each cross-validation test. All the analysis since the implementation of the Prophet Algorithm were carried out using the indexing-problem-corrected software.

7.6.2 Triple for loop to double for loop

The R-Script Fit and Forecast Univariate Methods (Figure 10) was featured by three for loops. The first for loop calls every logistics service provider, the second for loop calls every plant. At this point the software is able to call all possible material flows combinations. Finally, a third for loop was used to carry out the cross-validation tests from time $k < n$ to time $n - k - 4$, where n is the number of periods of the given Time Series, k is the starting time for the training set, and 4 owing to the 4-step-ahead forecasts.

Afterwards, still in the second for loop, the 4-month-ahead forecasts are carried out, using the chosen algorithm for the given material flow. Therefore, instead of making the software to look for all possible combinations among all logistics service providers and all plants, these combinations are already available in the Historical_Forwarded_Material.csv file, so they are provided directly to a for loop before starting the process, and then an inner for loop does the cross-correlation calculation. This changes allows the calculations to reduce from 3 for loops to 2 for loops, reaching a quadratic algorithm, which computationally better performs than a cubic algorithm (Sedgewick & Wayne, 2011). This implementation reduced the computational time to the half.

7.6.3 Historical MAPE values saved

Instead of calculating the cross-validation tests every month, the out-of-sample MAPE's can be saved in a file, which then can be used as an input for the next month's forecasts. This was an implementation made in order to improve the computational time. Using the saved historical MAPE values from the cross-validation tests, the computational time for the now double for-loop, in the R-Script Fit and Forecast Univariate Methods, was reduced up to 96%. The whole calculation including the triple for loop, the prophet algorithm and the Automated Simple Moving Average would take up to 2,3 hours; while, with the double for-loop and using the saved historical MAPE's as input, the forecasts can be obtained in 5 minutes in a 4 GB RAM, Intel Core i5 4310 2.7 GHz Laptop. This result allows now the logistics analysts to avoid waiting time to use the workstation.

7.7 Forecasting for Consolidation Centers

Given that the company's analyst has used the $\alpha -$time series to make all forecasts, no forecasts for their own Consolidation Centers have been made so far. This is so because the Consolidation Center do not have any Production Program to relate with. Notwithstanding, even though making forecasts with the $ton -$time series results in less accurate results than the forecasts based on $\alpha -$time series, this is a feasible solution which can deliver this important information.

Therefore, $ton -$time-series-based forecasts were made for 4 consolidations centers. Both univariate and multivariate methods were used to make the forecasts, that is to say, all the forecasts for the Consolidation Center's observations are all made based on univariate methods and then all made based on multivariate methods. These two results are then compared, and the most accurate calculations are chosen, i.e. better MAPE.

This added to the calculation 56 additional material flows, so that the new number of total material flows time series available to forecasts is 537.

7.8 Error Measures Improvements

7.8.1 MAPE Evaluation

When evaluating the model fitting, there are several criteria that may be of value. Notwithstanding, as (Montgomery, 2016) mentioned, concentrating too much on the model that produces the best historical fit often results in **overfitting.** Overfitting is one main feature to avoid in this forecasting process. This is achieved by allowing the software to recalculate the models' parameters monthly. Hence, the best approach is to select the model that results in the smallest standard deviation (or Mean Squared Error -MSE) of the one-step-head forecast error when the model is applied to data that were not used in the fitting process. This is called the out-of-sample forecast error standard deviation. This is the process which has been applied to the forecast evaluation by the company so far; however, the Mean Absolute Percentage Error (MAPE) was used instead.

The forecast process follows a rolling forecast method, which considers the MAPE out-of-sample forecast error for several cross-validation tests, namely, $number\_of\_tests: (n - k - 4)$, where n is the number of observation in a time series and k is the minimum number of training values to fit a forecasting model. The *k* value was originally set to be 6, i.e. the number of minimum observations to be found in a time series must be at least 10, specifically, a training set of 6 values and 4 values to calculate the out-of-sample MAPE with the 4-step-ahead forecasts obtained by each method. Finally, 4 stands for the 4-step-ahead forecast carried out in each test.

Due to this approach, there is some bias introduced for the first tests, which only use few values from the time series of the main leg material flow. For example, a given time series with 20 values, will have 10 cross-validation tests, however only the very last test will use 16 values of the time series to predict values 17-20. Thence, the accuracy of the model, on the first 1-9 cross-validation tests, is not the same as the accuracy of the cross-validation test which uses the maximum amount of information possible from the time series. Therefore two main changes will be introduced: (1) the Smoothed Errors (7.8.1.1), (2) the minimum value of *k* is now *n/2*, this last requirement makes sure to have at least 20 observations in the training set as stated by (Montgomery, 2016).

7.8.1.1 Smoothed Errors

In consequence of the previous description, due to the time relationship between the cross-validation tests and their corresponding MAPE out-of-sample forecast errors, this measure can be seen as an error tracking signal (Vidal Holguín, 2010) , and using a smoothing parameter will allow the model to have a smoothed 4-step-ahead forecast error, creating a exponentially weighted moving average (EWMA) tracking signal (Montgomery, 2016). The EWMA is a weighted average of all current and previous forecast errors, whose weights decrease geometrically with the “age” of the forecast error.

As stated by (Montgomery, 2016) and (Fildes & Goodwin, 2007)’s approach, when evaluating forecast accuracy, it is better to have different forecast error measures which can be then compared. Thence, the Mean Squared Error (MSE) was added to the software as a forecast error measure. Now, all the material flows forecasts obtain both a 4-ahead out-of-sample MAPE and MSE. According to (Montgomery, 2016), the MSE is a better measure of the out-of-sample forecast standard deviation. Therefore, the MSE is now the criterion to decide which forecasting method better performs at the cross-validation tests for a given material flow time series.

The Exponentially Weighted Moving Average approach was applied to both the MAPE and the MSE showing significant positive results. Firstly, it eliminated previous biased information in the historical MAPE’s evaluation. This result can be translated into lower historical MAPE values for each material flow time series (Figure 25). However, this does not necessarily mean an improvement in the software’s accuracy but rather a change in the forecast’s error approach. Secondly, the information is now more reliable, since the Exponentially Weighted Moving Average MAPE and MSE set the highest weights to the most recent cross-validation tests’ errors. In other words, cross-validation tests, having few values, are set to the lowest weights while cross-validation tests with the most values are set to the highest weights. Finally, the EWMA-MSE now leads the automation for which a given method is chosen to forecast a material flow time series; this is an important result for the next section, in which the tracking signals will be discussed.

7.8.2 Tracking Signals and Control Charts

Control charts are a well-known tool in today’s industry, and Shewhart control charts are the best known of these. Despite their popularity, they are unable to detect small shifts in a process quickly enough. For this reason other charts have been implemented, such as the Cumulative

Sum (CUSUM) and the Exponentially Weighted Moving Average (EWMA) charts (Maravelakis, Panaretos, & Psarakis, 2004).

EWMA charts are explained in section 5.3.2. These charts allow the analyst to check more easily a change in the mean of a process by allowing the lower and upper Control Limits to adjust as time evolves. This methodology was therefore applied to the forecasts using (Montgomery, 2016)'s approach.

For every single time series, the following simulation was made: Since the current modified and improved software is just available in April 2018, a historical data simulation was made. That is to say, the data were split into subsets and the 4-step-ahead forecasts for the months October 2017, November 2017, December 2017, January 2018, February 2018, March 2018 and April 2018 were calculated. Then the 4-step-ahead out-of-sample MAPE as well as the MSE were obtained. Using the corresponding MAPE's an EWMA Shewhart Control Chart for every single material flow time series was created.

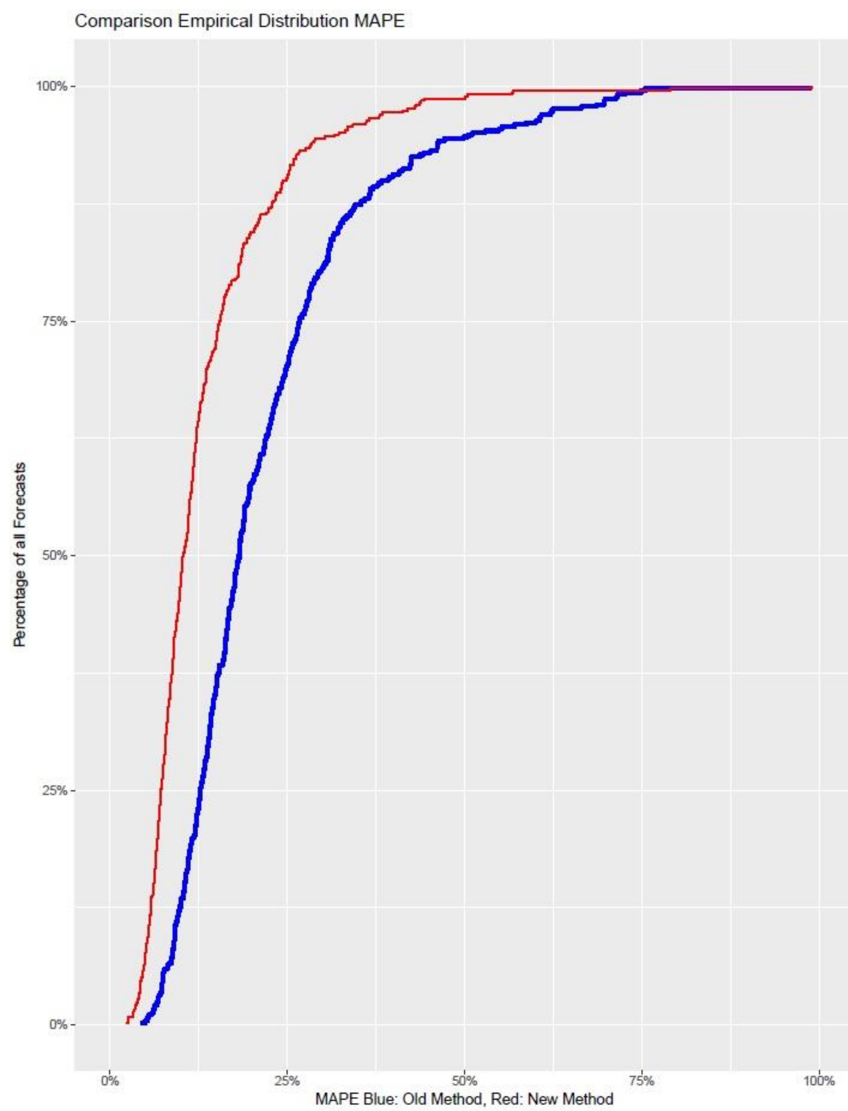


Figure 25. EWMA Cumulative MAPE Distribution Function

The results are highly important. Since the software calculates the model parameters in every single simulation, i.e. for every month in which the 4-month-ahead forecast was carried out, the best model chosen was not the same as the former one (in most of the cases). As there is new information, the added observed values are used to (1) recalculate the model parameters, (2) make one additional cross-validation test, (3) add the corresponding out-of-sample MAPE and MSE to the historical forecast error data base (described in 7.6.3), and then (4) decide, based on the lowest MSE, which method better now performs with the added information, and (5) use it correspondingly to obtain the new forecasts.

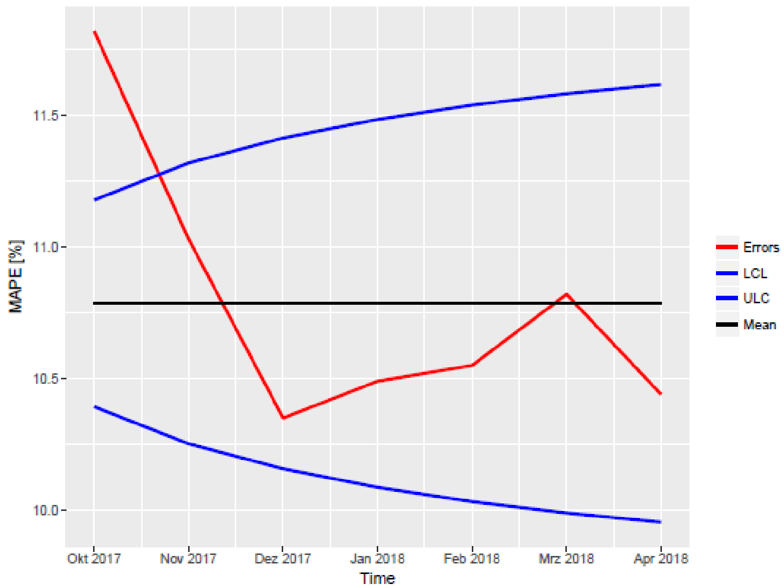
The EWMA Control Charts shown in Figures 26, 28 and 30 contain the Upper and Lower Control Limits, which change over time (blue lines). The MAPE measure was the one chosen since this allows a faster and easier interpretation of the forecasts errors by the logistics analysts. The mean of the MAPE's (black line) and the corresponding MAPE's over time (red line) represent the material flow time series' residuals. Moreover, a chart on the right displays which method was chosen in every single month. There are currently 536 Control Charts available, including both plants and consolidation center material flow forecasts.

The first highlighted result is that some time series follow the same Forecasting Model in all the simulations. For example, Figure 26 follows the Prophet Forecast Algorithm, the corresponding time series with their respective actual and should values are depicted in Figure 27. This time series is clearly non-stationary, but Prophet Algorithm is able to capture its underlying features and over time "learns" how the values will behave in the future so that the error measure improves. The errors lie between the control limits.

On the other hand, there are other time series in which the software chooses different forecasting methods over the time, improving the error measure considerably, so that the residuals go below the Lower Control Limit (Figures 28 and 29). This does not mean that the forecasts are out of control but rather that the mean of the process is changing, since the software is selecting the best method each time and this decision leads to better forecast errors.

Finally, there are some cases, in which the Error Measure worsened and went above the Upper Control Limit. A sample case is shown in Figure 30. This indicates that the new observed values increased the time series variability, so this is a good input for the logistic analyst to check whether the new information in the data bases is correct or whether some unexpected situation caused changes in the material flow volume.

All this information is plotted on a pdf file called **Control_Charts.pdf**, i.e. the Control Charts for every single material flow. Moreover, the first page displays a summary of the underlying results (Figure 32). For the whole simulation, in the month of April the material flow time series' residuals are featured as: Out of 536 time series, 95 MAPE's are above the Upper Control Limit, 106 are between the control limits and 335 are below the Lower Control Limits. This indicates that 62.7% of the forecast, corresponding to the ones below the Lower Control Limit, have changed their respective Forecasting Method and their mean forecasting error has changed. Additionally, only 17.7% of the forecasts are above their Upper Control Limits. This is explained by the information accuracy problem that the company currently faces and the simulation time frame, which only covers 7 months. Finally, the 20% of the Time Series MAPE are under control. Here it is important to apply further research, since 7 values for the Control Charts are still too few to correctly assess the Error Measure's performance. Unfortunately, the information accuracy is the main factor which constrains the research's scope.



	Time	Methods
1	Okt 2017	Prophet Forecast
2	Nov 2017	Prophet Forecast
3	Dez 2017	Prophet Forecast
4	Jan 2018	Prophet Forecast
5	Feb 2018	Prophet Forecast
6	Mrz 2018	Prophet Forecast
7	Apr 2018	Prophet Forecast

Figure 26. *Control Chart Sample, MAPE between control limits*

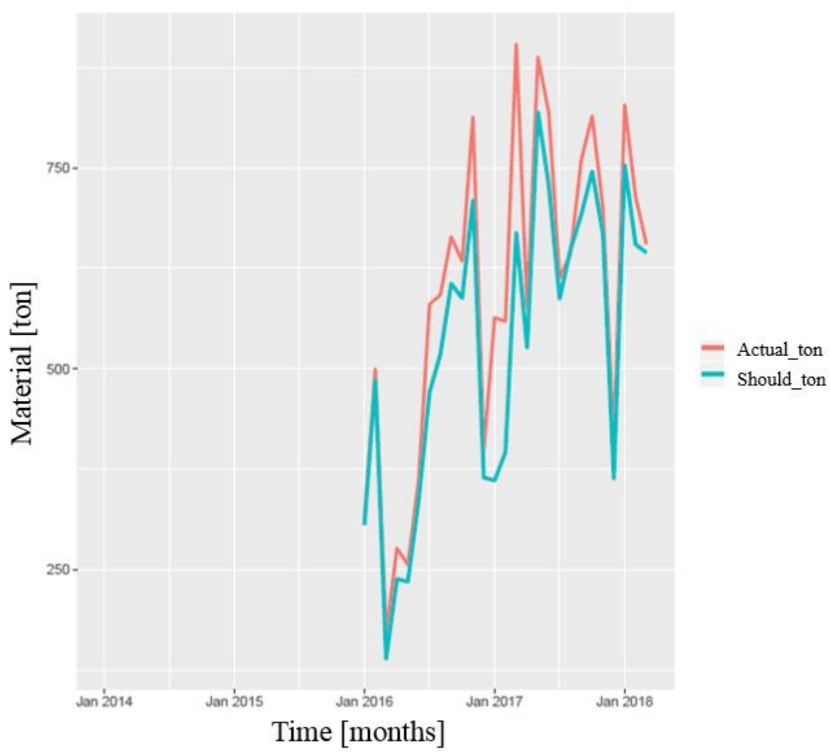
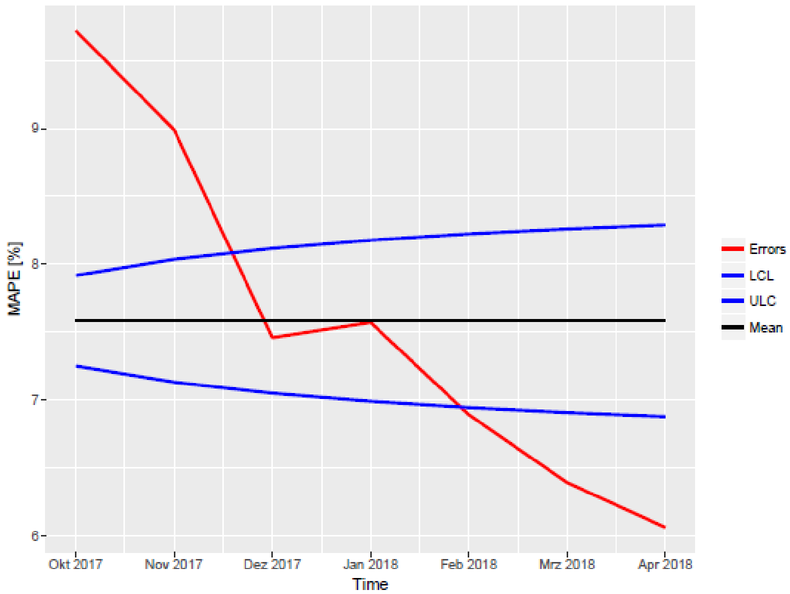


Figure 27. Material Flow from Sample Chart Figure 26



	Time	Methods
1	Okt 2017	ARIMA Forecast
2	Nov 2017	ARIMA Forecast
3	Dez 2017	Multivariate Time series
4	Jan 2018	Simple Moving Average
5	Feb 2018	Simple Moving Average
6	Mrz 2018	Simple Moving Average
7	Apr 2018	Exponential Smoothing Forecast

Figure 28. Control Chart Sample, MAPE below lower control limit

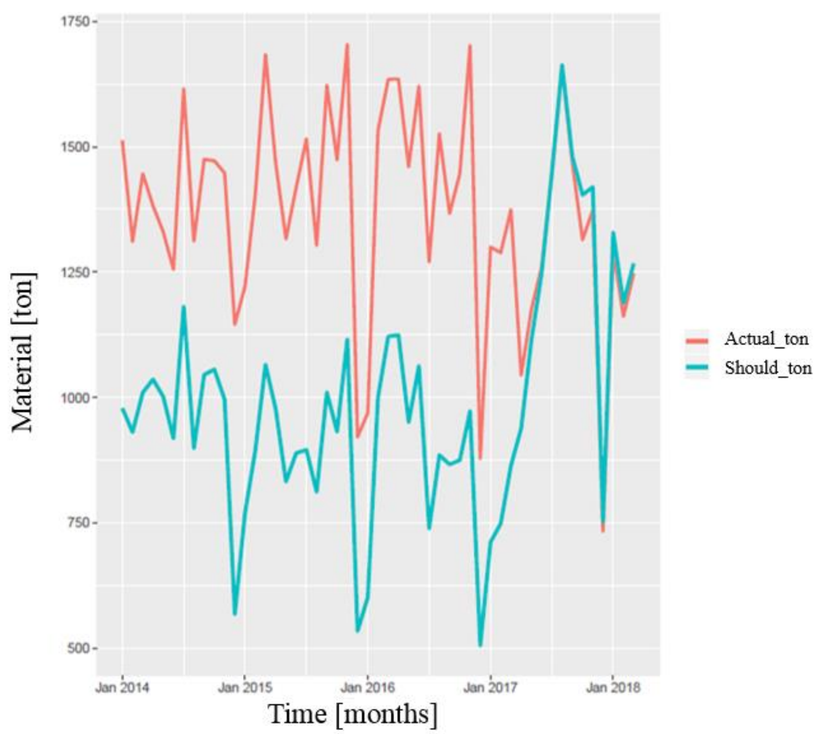
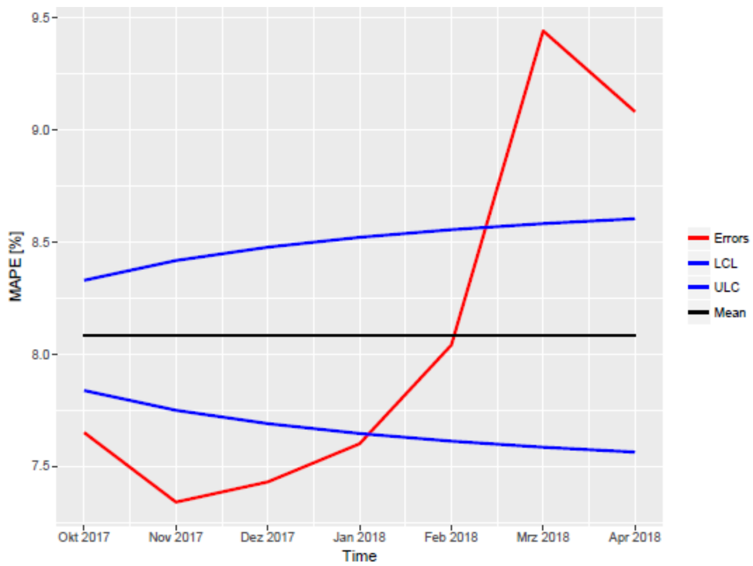


Figure 29. Material Flow from Sample Chart Figure 28



	Time	Methods
1	Okt 2017	ARIMA Forecast
2	Nov 2017	ARIMA Forecast
3	Dez 2017	ARIMA Forecast
4	Jan 2018	ARIMA Forecast
5	Feb 2018	ARIMA Forecast
6	Mrz 2018	Neural Network Forecast
7	Apr 2018	Neural Network Forecast

Figure 30. Control Chart Sample, MAPE above upper control limit

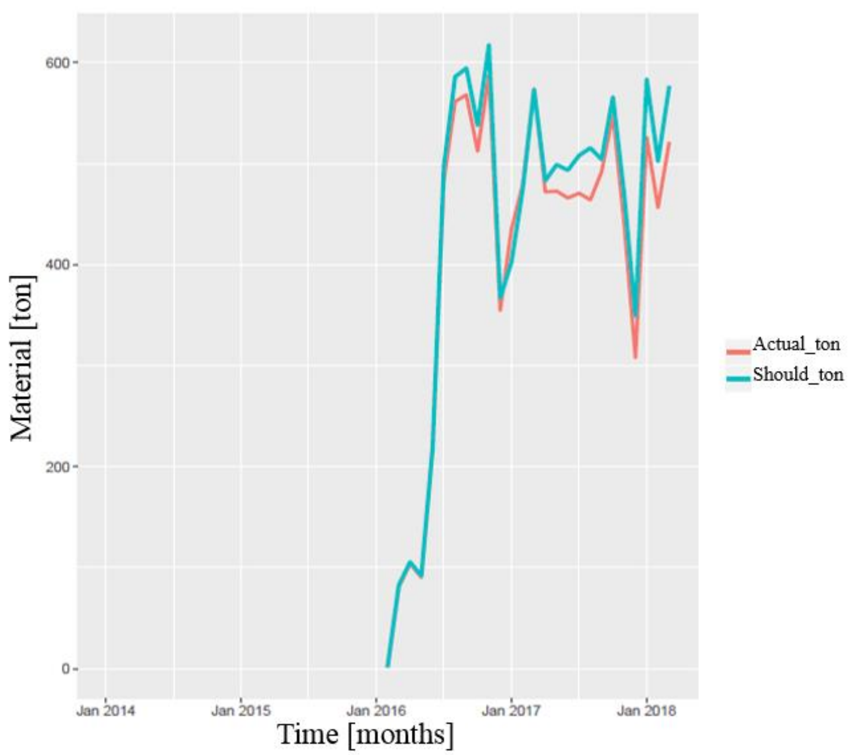


Figure 31. Material Flow from Sample Chart Figure 30

	Group	Quantity	Percentage
1	Time Series above UCL	95	17.72 %
2	Time Series into the Control Limits	106	19.78 %
3	Time Series below the LCL	335	62.5 %
4	Total	536	100 %

Figure 32. Summary Control Charts April 2018

7.9 Final Software Procedure Description

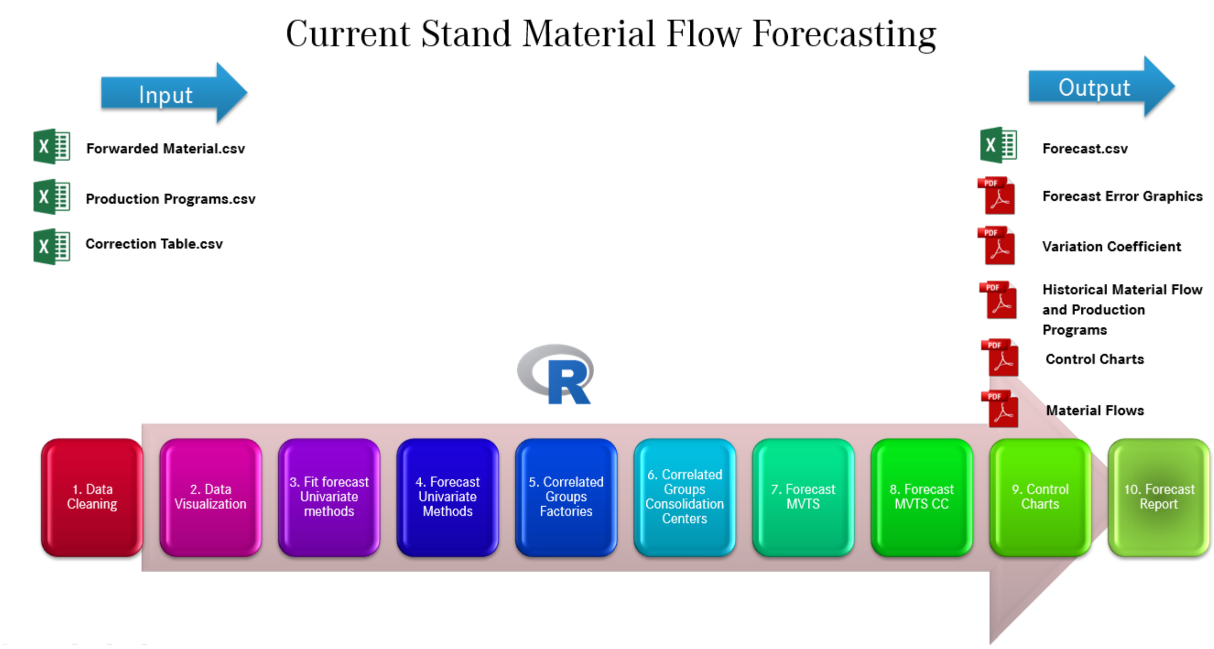


Figure 33. Final Software Flow Chart Description

The final software procedure designed in this work is depicted in Figure 23. Now there are 10 R-Scripts which apply all the added functions and new algorithms. The process can be described as follows:

(1) **Data Cleaning:** Cleans the data; corrects wrong values with correction table; applies two outlier detection processes, namely, Interquartile Range and *tsoutliers;* then applies linear interpolation for missing values. Finally, it delivers a pdf with all the material flows charts, showing both actual and should time series (see Figure 9, for example). It also delivers the Material Flows pdf if the user wishes.

(2) **Data Visualization:** Delivers the Variation Coefficient charts as well as the Historical Material Flow and Production charts.

(3) **Fit Forecast Univariate Methods:** Applies univariate methods to the historical data of both plants and consolidation centers. The methods used are: Naïve, ARIMA, Neural Network, Exponential Smoothing, Prophet Algorithm and Automated Simple Moving Average. It delivers the historical MAPE's and MSE's, which can then be used to make the forecasts. This step should be done once for the next coming months.

(4) **Forecast Univariate Methods:** Applies univariate methods to both plants and consolidation centers. The methods used are: Naïve, ARIMA, Neural Network, Exponential Smoothing, Prophet Algorithm and Automated Simple Moving Average. If the historical MAPE's and MSE's are available, step 3 can be skipped. The historical MAPE's and MSE's work as input and then it is upgraded with the current new information, so that next month's forecasts can reuse them.
(5) **Correlated groups Factories:** Finds all possible 1-lag-cross correlated material flows that go to a plant.
(6) **Correlated groups Consolidation Centers:** Finds all possible 1-lag-cross correlated material flow that go to the Company's Consolidation Centers.
(7) **Forecast MVTS:** Applies Vector Autoregressive Model to all correlated group of material flows on the Factories and delivers the corresponding Forecast based on this Multivariate Method.
(8) **Forecast MVTS CC:** Applies Vector Autoregressive Model to all correlated groups of material flows on the Consolidation Centers and delivers the corresponding Forecast based on this Multivariate Method.
(9) Control Charts: Delivers the control charts as pdf and join all the forecasts delivered by Forecast Univariate, Forecast MVTS and Forecasts MVST CC. The final forecast data is then called Joint Forecasts.
(10) Forecast Report: Delivers the forecast report as csv file, as well as the Forecast Error Graphics as pdf.

The final software performance can also be assessed by the distribution of the forecasting methods. This distribution can be found in Figure 34. It is important to highlight, that now the three new introduced algorithms take up to 29% of all forecasts. However, 15.2% of the algorithms are forecasted by the Ensemble Forecast Method, which is also influenced by the three new algorithms. Hence, it can be stated that, the new algorithms take part in up to 44.2% of the forecasts. Another important result is that Naïve Algorithm takes up to 10% of the forecasts, this tells about the high variability and inconsistence patterns that some material flows show.

Furthermore, Figure 35 shows a scattered plot of the single MAPE values for each material flow and their corresponding monthly average material volume. This graphic allows to understand how the distribution of the forecast is, and how they may influence on extra costs. When a material flow forecasts has a high forecast error and high monthly material volume, then this high variability is likely to cause additional transportation costs with high impact. On the other hand, a high forecast error but a low monthly material volume will not have a great impact in additional transportation costs. That is why, the ideal MAPE distribution is low forecast errors on high material volume and ideally not too high forecast errors on low material volumes.

Figure 35 shows then how the new software improved the MAPE distribution. For this graphic the forecasts for the month April 2018 were carried out using both the new and the original software, and then the results were plotted together. The red points representing the new method are now lower than the blue ones. This demonstrates graphically how the new software performs better than the original version.

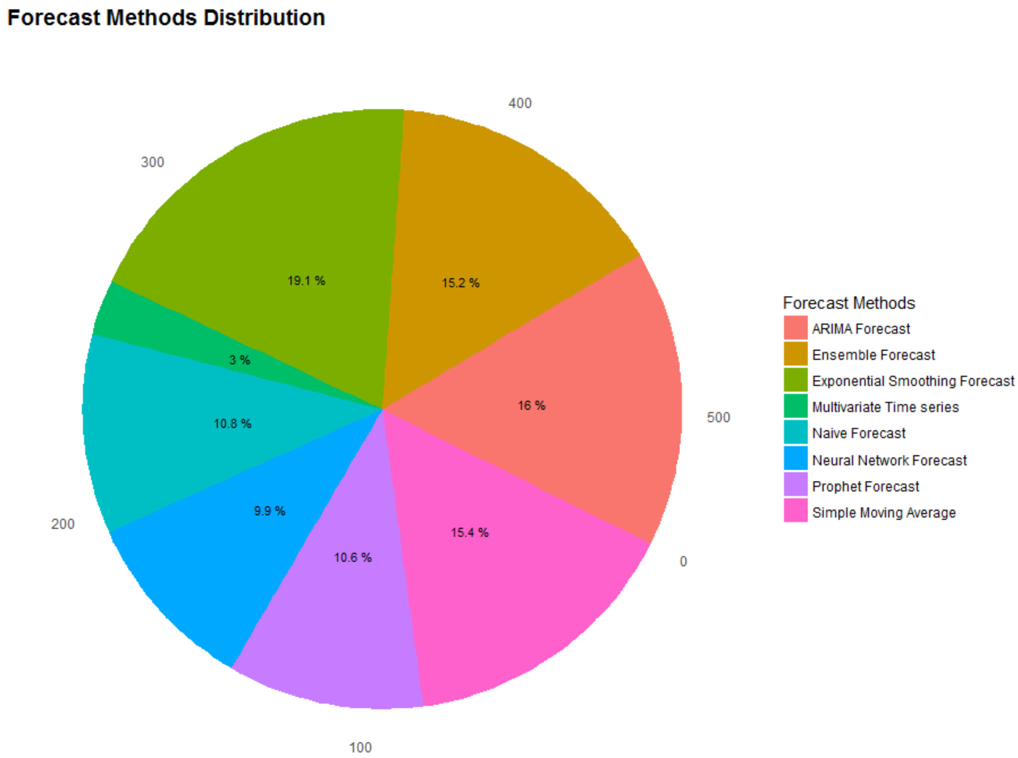


Figure 34. Forecast Methods Distribution

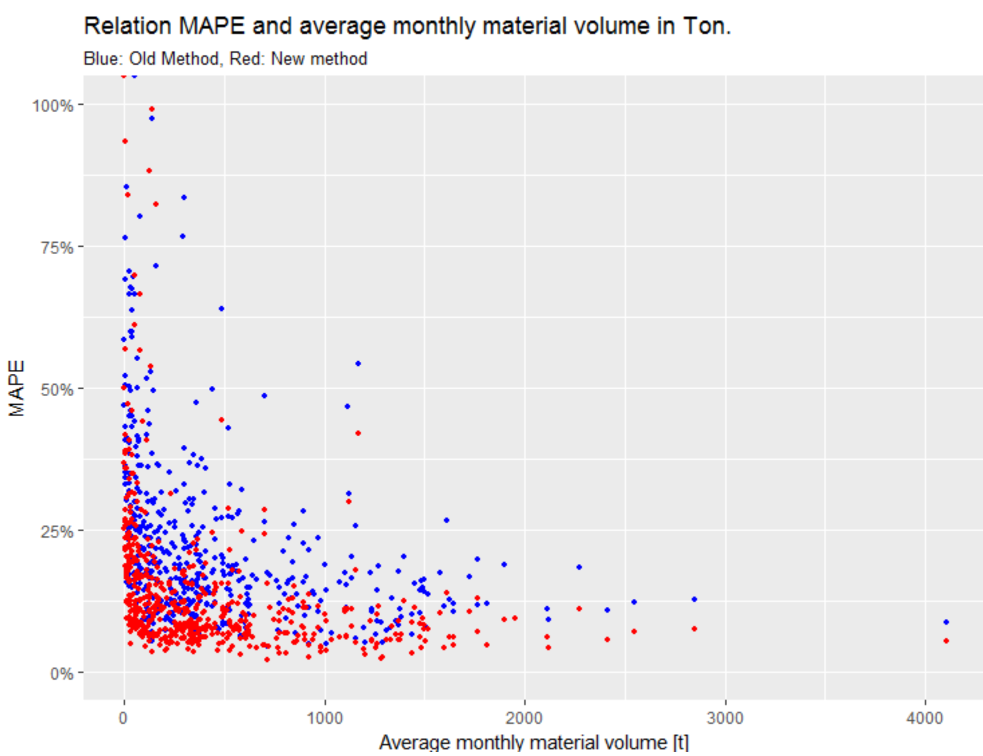


Figure 35. Relation MAPE and average monthly material flow in tons April 2018

Finally, Table 11, summarizes the overall change applied to the software's performance. At the starting time in February 2018, the software was able to produce forecasts to 471 material flows, the consolidation centers did not have any forecast and only 13% of the forecasts' MAPE were lower than 10%. The new software, up to April 2018, is now capable of delivering forecasts for 537 material flows, including the consolidation centers. Forecasts having less than 10% MAPE now represent the 43.6% of all the material flows. This result shows an improvement of 30.6 p.p. Another relevant feature to highlight is the forecasts having more than 40% MAPE, these were reduced from 11% to 4.5%.

To conclude, it can be implied from Table 11, that up to April 2018, 80.1% of all material flows have a MAPE of less than or equal to 20%.

Table 11. *MAPE distribution Comparison Table February vs April 2018*

MAPE Category	Frequency Original Software Feb 2018	Percentage Original Software Feb 2018	Frequency Improved Software Apr 2018	Percentage Improved Software Apr 2018
lower than 10%	61	13.0%	234	43.6%
between 10% and 20%	215	45.6%	196	36.5%
between 20% and 30%	106	22.5%	62	11.5%
between 30% and 40%	37	7.9%	21	3.9%
higher than 40%	52	11.0%	24	4.5%
TOTAL	**471**	**100,0%**	**537**	**100%**

8 CONCLUSIONS

The current research extended the forecasting methods' scope for the monthly material flows of an international automotive company by allowing the implementation of three new algorithms, namely: The Prophet Algorithm, the Vector Autoregressive (Multivariate Time Series) and Automated Simple Moving Average, and two new data cleaning methods: Automated Outlier Detection and Linear Interpolation.

Figure 35 shows then how the new software improved the MAPE distribution. The forecast for the month of April 2018 were carried out using both the new and the original software, and then the results were plotted together. The blue points (upgraded software) are now lower than the red ones (original software). This proves graphically how the upgraded software performs better than the original version. Quantitatively, it can be stated that, according to Table 11, up to April 2018, 80.1% of all material flows have a MAPE of less than or equal to 20%, in comparison with the 58.6% of all material flows which had the same behavior in the original software.

All the analysis realized in this research were made with actual data from the company, and the upgraded software was approved by the logistics analysts to make all future material flow forecasts. The software's copyright is property of the company.

To sum up, the new algorithms' performance, it can be stated that, the Prophet Algorithm allows the analyst to adjust manually or automatically different parameters like seasonality, trends and holidays. In this research, these parameters were chosen automatically by the algorithm. Further research may test how individual parameters can be assigned to the material flow time series. This algorithm proved to have a high performance even if the parameters are automated. 10.6% of all forecasts are obtained now by the Prophet Algorithm.

Moreover, Simple Moving Average can capture the average trend of a time series. This feature is suitable for time series with high variability, like many of the company's material flows. This algorithm proved to have a high positive impact on the forecasts and had the highest improvement on the number of material flows. 15.4% of all forecasts are now obtained by the Automated Simple Moving Average. This is a significant result since Simple Moving Average is the simplest method to forecast.

On the other hand, Vector Autoregressive Method is certainly a powerful algorithm which, based on the Multivariate Time Series Theory, is able to take advantage of the lag-cross correlation among different time series and create models capable of predicting future values with high accuracy. This algorithm had the highest improvement on the average MAPE (23.78%), which also demonstrates its potential. Further research can be also applied, like extending the 1-lag VAR model to 2-lag VAR model; or even using the Vector Autoregressive Moving Average (VARMA) model, which can also consider the Moving Average of the lag-cross correlations.

The three new algorithms represent now 29% of all forecasts. Moreover, since the Ensemble Forecast method takes the average of all forecasts delivered by all methods, the new algorithms also influence these results. It can be implied then that the three new algorithms take part in up to 44.3 % of the forecasts.

Given the available information, namely, the forecasts for the production programs in the coming months; these data could be used as input to make a Multivariate Linear Regression Analysis, which is also likely to deliver good forecasts outperforming the current methods.

The outlier detections methods, explicitly, the Interquartile Range and the Time Series Outlier Detection are robust enough for the current input data. These methods are complemented by the Linear Interpolation Algorithm, which prepares the data, filling in the missing values, before the time series go to the outlier detection process.

The two new cleaning data procedures allowed the software to reach a much higher accuracy than before. This demonstrates that the most important feature when making forecasts is the data quality. This is indeed the most important issue that the company currently faces. The company is undergoing a data bases upgrading and synchronization process in all its plants, which causes the company to have information quality problems.

At the current research, the main problem was that the Bill of Materials is not clear to every product. That is why, the monthly forwarded material is forecasted, so that the logistics managers can have a clearer view of how much material in aggregated units they will need to transport in the coming months.

9 REFERENCES

Berkovec, J. (1985). Forecasting automobile demand using disaggregate choice models. *Transportation Research Part B*, *19*(4), 315–329. https://doi.org/10.1016/0191-2615(85)90039-6

Box, G., & Tiao, G. (1975). Intervention Analysis with Applications to Economic and Environmental Problems. *Journal of the American Statistical Association*, *70*(349), 70–79. https://doi.org/10.2307/2285379

Brown, R. C. (1972). Detection of Turning Points in a Time Series. *Decision Sciences*, *3*(4), 1–18. https://doi.org/10.1111/j.1540-5915.1972.tb01664.x

Byrd, R. H., Lu, P., Nocedal, C., & Zhu, C. (1994). A limited memory algorithm for bound constrained optimization, 1–12. https://doi.org/10.1145/279232.279236

Chen, C., & Liu, L.-M. (1993). Joint Estimation of Model Parameters and Outlier Effects in Time Series. *Journal of the American Statistical Association*, *88*(421), 284–297. https://doi.org/10.2307/2290724

Claeskens, G., Magnus, J. R., Vasnev, A. L., & Wang, W. (2016). The forecast combination puzzle: A simple theoretical explanation. *International Journal of Forecasting*, *32*(3), 754–762. https://doi.org/10.1016/j.ijforecast.2015.12.005

Fildes, R., & Goodwin, P. (2007). Against your better judgment? How organizations can improve their use of management judgment in forecasting. *Interfaces*, *37*(6), 570–576. https://doi.org/10.1287/inte.1070.0309

Gardner, E. S. (2006). Exponential smoothing: The state of the art—Part II. *International Journal of Forecasting*, *22*(4), 637–666. https://doi.org/10.1016/j.ijforecast.2006.03.005

Gnauck, A. (2004). Interpolation and approximation of water quality time series and process identification. *Analytical and Bioanalytical Chemistry*, *380*(3 SPEC.ISS.), 484–492. https://doi.org/10.1007/s00216-004-2799-3

Granger, C. W. J. (1969). Investigating Causal Relations by Econometric Models and Cross-spectral Methods. *Econometrica*, *37*(3), 424. https://doi.org/10.2307/1912791

Harvey, A., & Peters, S. (1990). Estimation procedures for structural time series models. *Journal of Forecasting*, *9*(2), 89–108. https://doi.org/10.1002/for.3980090203

Hyndman, R. J., & Athanasopoulos, G. (2014). *Forecasting: Principles and Practice: Notes*. *OTexts* (1st. Editi). Texts, Online Open-Acces Textbooks. Retrieved from https://www.otexts.org/fpp/2/5%5Cnhttps://play.google.com/store/books/details?id=gDuRBAAAQBAJ%5Cnhttps://play.google.com/books/reader?id=gDuRBAAAQBAJ&printsec=frontcover&output=reader&hl=el

Hyndman, R. J., & Khandakar, Y. (2008). Automatic Time Series Forecasting: the forecast Package for R, *27*(3), 23. https://doi.org/10.18637/jss.v000.i00

Lepot, M., Aubin, J.-B., & Clemens, F. (2017). Interpolation in Time Series: An Introductive Overview of Existing Methods, Their Performance Criteria and Uncertainty Assessment. *Water*, *9*(10), 796. https://doi.org/10.3390/w9100796

Logility. (2016). Eight Methods that Improve Forecasting Accuracy Eight Methods that Improve Forecasting Accuracy, 12.

Logility. (2017). Practical Tips to Improve Demand Planning, 8. Retrieved from www.logility.com

Maravelakis, P. E., Panaretos, J., & Psarakis, S. (2004). EWMA chart and measurement error. *Journal of Applied Statistics*, *31*(4), 445–455.

https://doi.org/10.1080/02664760410001681738
Montgomery, D. C. (2016). Time Series Analysis and Forecasting. https://doi.org/10.1007/978-3-319-28725-6
Schöneberg, T., Koberstein, A., & Suhl, L. (2010). An optimization model for automated selection of economic and ecologic delivery profiles in area forwarding based inbound logistics networks. *Flexible Services and Manufacturing Journal*, *22*(3–4), 214–235. https://doi.org/10.1007/s10696-011-9084-5
Sedgewick, R., & Wayne, K. (2011). *Algorithms*. (P. Education, Ed.) (4th Editio). Boston: Addison-Wesley.
Shan, H., Hu, E., Wang, L., & Liu, G. (2014). Forecasting Electric Vehicles Demand in USA. *Pakistan Journal of Statistics*, *30*(5), 939–955. https://doi.org/10.1007/978-3-662-43871-8
Smith, J. ., & Wallis, K. F. (2009). A simple explanation of the forecast combination puzzle. *Oxford Bulletinn of Economics and Statistics*, *71*, 331–355.
Sorensen E., B. (2012). ECONOMICS 7344. Houston, Texas: University of Houston. Retrieved from http://www.uh.edu/~bsorense/arma-intro2012.pdf
Syntetos, A. A., Babai, Z., Boylan, J. E., Kolassa, S., & Nikolopoulos, K. (2016). Supply chain forecasting: Theory, practice, their gap and the future. *European Journal of Operational Research*, *252*(1), 1–26. https://doi.org/10.1016/j.ejor.2015.11.010
Taylor, S. J., & Letham, B. (2017). Forecasting at Scale. *PeerJ Preprints*, 1–17. https://doi.org/10.7287/peerj.preprints.3190v1
Tsay, R. S. (University of C. (2014). *Multivariate Time Series Analysis*. Chicago, IL.
Vidal Holguín, C. J. (2010). *Fundamentos de Control y Gestión de Inventarios* (2da Edició). Cali, Colombia: Universidad del Valle.

10 ATTACHMENT 1: FLOW CHART FORECAST SOFTWARE