UNIVERSITÀ
DI TRENTO

Department of Information Engineering and Computer Science

Bachelor's Degree in
Computer Science

Final Dissertation

Predictive Queue Management in Supermarkets

Supervisors
Prof. Alberto Montresor
Iacopo Carreras
Andrei Tamilin

Student
Matteo Destro

Academic year 2019/2020

Contents

Abstract

Long waiting queues at the checkouts are one of the major causes of customers' dissatisfaction in retailing. The time spent waiting for service greatly affects the perceived service quality and therefore can have a significant impact over customers' loyalty and, eventually, sales. In the context of supermarkets, the only way for store managers to have control over the queues length is by adjusting staffing levels, in an attempt to provide a uniform level of service at all times.

The decision of either increasing or decreasing staffing levels is typically taken on the spot by looking at the current queues' state. This method is not efficient since it is not responsive enough to sudden changes in traffic levels and does not take in account the near-term evolution of customers flow. Therefore, a more efficient way of improving customers' satisfaction may be to prevent waiting queues before they appear, increasing the number of open terminals according to the predicted inflow of people at the checkouts. However, keeping a checkout open has a cost in terms of manned staff, since it reduces staffing levels for others store operations, and the benefits of decreasing waiting times could be easily overcome by the resulting costs. An optimization which takes in account both cost and customers' satisfaction is therefore necessary.

The work presented in this thesis was developed as part of RetailerIN by Thinkinside, an advanced in-store analytics and engagement solution, that tracks in real-time the movements of the supermarket's customers and provides insightful information regarding their shopping behavior, such as identifying the high-traffic areas or the most recurrent shopping journeys. With the support of these performance indicators, retailers are able to optimize the store layout, improve the overall shopping experience and engagement, and receive suggestions on how to adjust staffing levels based on shoppers volume. The main objective of this thesis is to improve the latter feature by developing a predictive model of the queueing behavior in retail stores, with the aim of forecasting near-term arrival rates at the checkouts, supporting the management in a more efficient pre-positioning of the staff to avoid, or at least reduce, waiting queues. The system shall suggest an optimal checkouts configuration, minimizing the staffing costs while maintaining the customers' waiting times under a maximum threshold. The main hypothesis behind this approach is that it is possible to decrease the total manned time of the staff at the counters without any loss of customer satisfaction, i.e. without increasing the waiting time past an acceptable value.

A predictive model is developed in order to get an accurate forecast of the queue length and to determine the best checkouts configuration. First, a model for forecasting the number of customers entering the store, based on the data of past weeks, is used in order to achieve a multi-step forecast. Next, the measured and predicted inflow rates are combined with the dwell time distributions to get a near-term forecast of the rates of customers arriving at the checkouts. Finally, a predictive model for the expected value of the queue length and waiting time is defined by applying queueing theory. With these forecasts, once a maximum acceptable queue length has been defined, different checkouts configurations can be evaluated to determine the optimal one. The methods used to implement these models include: artificial neural networks, ARIMA models, distribution analysis and different queueing theory techniques.

The model thus designed can be easily integrated into RetailerIN, offering store managers near-term forecasts of the traffic levels and suggesting in real-time the optimal staffing levels in order to support operational decision making. It is also possible to adapt the implementation to meet potential store's specific requirements, in order to take in account different layouts and any eventual pre-existing staffing constraint or queue management policy. Finally, all the methodologies and results discussed in this thesis are applicable, with the appropriate adjustments, to other contexts besides supermarkets, as long as they can provide some kind of traceable assets and have at least one waiting queue with adjustable service levels.

1 Introduction

This chapter consists of three sections. In the first one, the research problem behind this thesis as well as the initial hypotheses are presented. The next section gives an overview of the RetailerIN system, that collected and provided all the data used in this research. The last one describes the general structure of the thesis.

1.1 Research problem and initial hypotheses

This thesis was written as part of an internship at Thinkinside[1], a company located in Trento specialized in location intelligence for indoor spaces. Thinkinside flagship product, RetailerIN[2], is designed to help stores obtain knowledge of the customers' behavior inside their shop. Typically, those stores only have information about the purchases made, but have no visibility on what the customers actually do while shopping: how they move and respond to promotions, how different displays configuration impact sales, where are the bottlenecks in the shopping flow, what are the under-visited areas... By tracking the position of the stores' physical assets, e.g. carts and baskets, with high-precision localization sensors, and by analyzing the shoppers' movements, RetailerIN is able to derive all these valuable information and present them in a visual dashboard for the store managers.

From these analyses, the system developed by Thinkinside is also able to suggest to adjust staffing levels based on customers volume, for example by repositioning sales associates across departments or by prioritizing the activities to perform in the store in order to maximize store operations while preserving an adequate customer support. In this context, a great impact on the customers' shopping experience is given by waiting queues. Various researches have shown that long queues and waiting times have significant influence over the perceived service quality and therefore over the customers' loyalty and conversion rate [5, 14, 18]. Hence, avoiding, or at least reducing, the waiting queues should eventually lead to an increase on sales. In the context of supermarkets, the only way to achieve this kind of optimization is by opening or closing checkouts according to the customers flow. Typically, this is done directly by the staff, by observing the current queue length and deciding on whether a terminal shall be opened or closed. While this approach can be quite effective if the adjustments are responsive enough to sudden changes in the rate of people arriving at the checkouts, it necessarily introduces a certain delay given by the checkouts' opening time and by some of the customers leaving a queue and joining the new one. Moreover, it does not take in account the possible traffic spikes: for example, a new checkout get opened to reduce the current queue length, but it is left unused after a short time due to the incoming traffic returning to a normal level. These issues could be prevented by defining a predictive model for the arrival rate at the checkouts, that would allow to prevent the opening of additional counters if the increases in traffic levels are not persistent. The previous scenario also introduces a second factor that must be considered to achieve the aforementioned optimization: the cost in terms of manned staff at the checkouts. Naturally, the minimum possible queue length can be achieved by maximizing the service capacity, that is leaving all the available terminals opened. This however would lead to overstaffing in most of the cases, since a lower number of checkouts would still be able to satisfy every customer. Moreover, it would decrease the overall store performance and increase the total time spent in idle by the staff, since the cashiers would remain at the counters without any actual customer to serve, preventing them from focusing on the right activities at the right time. The number of open counters should therefore be minimized in order to maximize store operations while preserving an adequate customer support.

Both presented requirements, maximizing customers satisfaction by preventing queues and maximizing store operations by reducing the number of available terminals, conflict with each other,

[1] www.thinkin.io

[2] www.retailerin.com

meaning that an optimal compromise must be designed. The purpose of this research is to find such compromise, that should allow to control and reduce the working time beyond the checkouts without creating long waiting queues. This problem could be approached in three different ways:

- Minimizing the number of open counters, such that a predetermined maximum threshold for the queue length (or waiting time) is not exceeded.
- Minimizing the queue length, such that a predetermined maximum threshold for the number of open counters, and therefore for the staffing level, is not exceeded.
- Defining a "cost function" that quantizes both factors by assigning them a value in terms of costs. By taking in account the inflow traffic, the optimal values for the two factors can be determined such that the cost function is minimized.

While the last approach could give the best results, it is difficult to assign a value in terms of costs to the length of a queue or to the number of open terminals, since it may depend on other external factors and specific store policies. Considering that it is easier to manipulate the number of open counters, the first approach was therefore chosen and researched. This approach is based on two main assumptions:

- Optimizing the number of open checkouts in a supermarket decreases the total staff's idle time during the shifts at the counters.
- It is possible to decrease the number of open checkouts without any loss on the overall customers satisfaction, represented by the waiting queues length.

As explained before, a predictive model for the expected queue length is necessary for this optimization to be effective. This requirement introduces another hypothesis that must hold:

- It is possible to forecast the waiting queue length by analyzing the customers' shopping behavior in the previous periods.

The next chapters explore and validate these hypotheses by the analysis of real-world data from two different supermarkets.

1.2 The RetailerIN system

All the data used in this research is provided by the RetailerIN system. RetailerIN is an in-store analytics and engagement solution, able to measure and analyze in real-time and with high precision (sub-meter accuracy) how shoppers move and interact with products in a physical store. RetailerIN relies on an indoor location system based on BLE tags attached to baskets and carts and on receiver antennas mounted on the ceilings. Since only these assets are tracked, all the data collected is anonymous and not traceable to specific customers. For every asset, the RetailerIN system is able to build the movements path and to segment it into each different customer's shopping session, constructing a detailed "shopping journey" report. This data is then stored and processed in real time to extract relevant indicators and metrics on the customers' behavior. These indicators are used to build visual analytics dashboards for store and marketing managers, helping them to optimize the placement of goods, to measure the effectiveness of store layouts and to improve staff management.

1.3 Structure of the thesis

Chapter 2 explores the available literature and introduces different possible approaches to the research problem. Chapter 3 describes all the analyses conducted in order to obtain a better understanding of the data as well as identify the most appropriate solutions to achieve the research goals. In Chapter 4 the final prediction model is introduced and justified. Chapter 5 discusses the models' prediction accuracy and the choices made for the final implementation. The final chapter provides a conclusion derived from the results and points out some limitations and possible future improvements of the system.

2 State of the art

This chapter discusses the available related literature. First, various solutions to similar problems from other researches are described. All these researches falls into the broader discipline of *Operational Research* (OR), and mainly into two specific sub-fields: *Time Series Forecasting* and *Queueing Theory*. Since this thesis is also focused on these types of approach to the problem, the state-of-the-art techniques for those fields are discussed in the following two sections.

Some of the terminology that will be used in this chapter:

- *Inflow/outflow rate*: number of customers that enter/exit the supermarket in a given time interval.
- *Arrival rate*: number of customers that arrive at the checkouts in a given time interval.
- *Service rate*: maximum number of customers that can be served by each checkout in a given time interval.
- *Dwell time*: time spent by a customer inside the store.
- *Basket size*: total number of items bought in a single shopping session.

2.1 Literature review

There are various studies that try to analyze and predict the queue length in different settings, e.g. airports or hospitals, but little was found in the specific context of supermarkets or regarding staff optimization. While these unrelated researches can still give useful insight, it is difficult to adapt the proposed solutions to the context of this thesis, since every distinct setting introduces different constraints. Therefore, this section focuses on the findings of the supermarket-focused studies.

In 2004, Berman et al. [11] developed a system that manages the switching of workers between a "front room", where the checkouts are located, and a "back room", that is the rest of the shop. This system processes real-time data about the count of customers either in front and back rooms, with the goal of minimizing the customer's waiting time while maintaining an adequate level of staffing in the back room for store operations. To accomplish this performance objective, they first defined a maximum *waiting time threshold* that shall not be exceeded, and a minimum *time-average worker complement threshold* to complete all the back room work. They used a $M/M/\infty$ queueing model to determine the performance of the system with a given number of front room workers. With this method, they were able to suggest in real-time the optimal strategy of switching workers from one room to another, minimizing the average waiting time while satisfying the threshold constraints.

Aksu H. [6] main goal was to optimize the idle time of the staff operating at the checkouts while maintaining the waiting queue length under a predetermined threshold. In 2008, he proposed a model based on the inflow of customers at the entrance and at the checkout area, the current waiting queue length and the number of available checkouts. He divided the supermarket area into a shop and a checkout area, and used a video-based system to count the customers moving between them. He then developed a prediction model to calculate a realistic forecast of customer's dwell times. The inward flow into the checkout area was predicted using the inward flow into the shop, since the inward flow into the shop appears with a delay at the checkout area, and this delay is the dwell time. To consider non-standard events and increase the accuracy, the predictions were adjusted with real-time measurements. Queueing theory was then applied to predict the average waiting time at the checkouts. With this information, the system was able to suggest to either open or close counters to meet a predefined minimum and maximum acceptable queue length. To avoid closing and opening too frequently, the system was also able to decide in how many cases the defined waiting queue length

could be exceeded, e.g. it was acceptable to exceed the target value for a short period of time. By following the model's suggestion, the supermarket used for testing decreased the staff's time spent in idle by 62.17% on average, while increasing the queue length by only 0.2 (from 1.9 to 2.1 customers in queue on average).

2.2 Time series forecasting

A *time series* is a collection of observations made sequentially through time. Since the customers traffic levels can be seen as points in time, many time series analysis techniques and prediction models can be used. *Time series forecasting* is the use of a model to predict future values based on previous observations. A time series can be decomposed into three main components:

- *Trend* (T_t): linear/nonlinear, increasing/decreasing behavior of the series over time.
- *Seasonality* (S_t): repeating patterns or cycle behavior over time.
- *Residual* (R_t): variability of the observations not explainable by the model.

There are two types of decomposition: *additive*, where the time series would be written as $y_t = T_t + S_t + R_t$, and *multiplicative*, where $y_t = T_t \cdot S_t \cdot R_t$. The additive decomposition is the most appropriate if the magnitude of the seasonal fluctuations, or the variation around the trend-cycle, does not vary with the level of the time series. When the variation in the seasonal pattern appears to be proportional to the level of the time series, then a multiplicative decomposition is more appropriate.

Once the main components of the time series have been determined, a predictive model can be defined. There are many different time series forecasting methods available, the most popular and effective are presented in the next sections.

2.2.1 Exponential smoothing

In the *Simple Exponential Smoothing* (SES) method, the forecasted values are based on a weighted average of the previous values, where the most recent observations are given more importance using larger weights [3].

A more complex implementation of the same principle is the *Holt-Winters Exponential Smoothing* method, which is able to decompose the time series into a level, trend and seasonal component, giving a more precise forecast [1]. However, it is not possible to model more complex series with multiple seasonality patterns, as in a supermarket inflow rate which presents daily and weekly seasonalities.

Taylor's *Double Seasonal Exponential Smoothing* method was developed to forecast time series with two seasonal cycles: a short one that repeats itself many times within a longer one [19].

2.2.2 Artificial Neural Networks

Artificial Neural Networks (ANN) allow to model complex nonlinear relationships between the target variable and its predictors. With time series data, subsequent lagged values can be used as inputs, in what is called an *Autoregressive Neural Network* (AR-NN) [20]. With seasonal data, it is also useful to include in the input the last observed values from the past seasons [7]. In general, to compute a forecast value $\hat{y}_{t+1}$ for time $t+1$, the corresponding input sample can be written as:

$$(y_t, y_{t-1}, ..., y_{t-p}, y_{t-m}, y_{t-2m}, ..., y_{t-Pm})$$

The notation AR-NN$(p, P, k)_m$ is used to indicate an autoregressive neural network, with:

- p: number of lagged observations in input;
- P: number of seasonal lagged observations in input;
- k: number of neurons in the hidden layer;
- m: number of time steps in a seasonal period.

For forecasting two steps ahead, the result of one-step forecast can be used in input. Therefore, once $\hat{y}_{t+1}$ has been determined, $\hat{y}_{t+2}$ can be computed by giving in input the sample:

$$(\hat{y}_{t+1}, y_t, y_{t-1}, ..., y_{t-p-1}, y_{t-m}, y_{t-2m}, ..., y_{t-Pm})$$

This method can be then applied recursively to obtain a three-step, four-step, etc... forecast.

2.2.3 ARIMA models

The *AutoRegressive Integrated Moving Average* (ARIMA) models provide another technique for time series forecasting. Exponential smoothing and ARIMA models are the two most widely-used time series forecasting methods, and provide complementary approaches to the problem. While exponential smoothing models are based on a description of the trend and seasonality in the data, ARIMA models aim to describe the autocorrelations in the data [12]:

- *AutoRegressive* means that the model uses the relationship between an observation and some number of lagged previous values to generate a linear regression model;
- *Integrated* refers to the use of differencing of raw observations (e.g. subtracting an observation from the previous time step value) in order to make the time series *stationary*. A stationary time series has constant and time-independent mean, variance and autocorrelation;
- *Moving Average* means that the model uses the dependency between an observation and the residual error from a moving average model applied to lagged observations.

Accordingly, the parameters of the ARIMA model are defined as follows:

- p: autoregressive order, the number of lagged observations included in the model;
- d: degree of differencing, the number of times that the raw observations are differenced;
- q: moving average order, the size of the moving average window.

This simple version of ARIMA can only be used to forecast non-seasonal time series, it can however be extended to model seasonal patterns with the Seasonal ARIMA (SARIMA) [7]. There are four additional seasonal parameters not part of ARIMA that must be configured:

- P: seasonal autoregressive order;
- D: seasonal degree of differencing;
- Q: seasonal moving average order;
- m: number of time steps in a seasonal period.

A SARIMA$(p, 0, 0)(P, 0, 0)_m$ model is equivalent to an AR-NN$(p, P, 0)_m$ model, but with additional parameters restrictions to ensure stationarity.

However, only a single seasonal effect can be modelled with SARIMA. To model more seasonalities, a SARIMAX model must be used, since it supports the use of additional *exogenous variables*: the multiple seasonal effects can be captured by Fourier terms and used as external variables. By doing this the overall forecast accuracy is increased and additional exogenous variables can be added to the model to take in account other external factors (e.g. weather or holidays).

2.2.4 TBATS models

An alternative approach proposed in 2011 by De Livera et al. [10] uses a combination of Fourier terms with an exponential smoothing model and a *Box-Cox transformation* [13], in a completely automated manner. The main advantage of a TBATS model over a SARIMAX model with Fourier terms is that the seasonality is allowed to change slowly over time, while the harmonic regression terms force the seasonal patterns to repeat periodically without changing. However, one drawback of TBATS models is that they are very slow to estimate, especially when the number of available observations is large, since they will consider various alternatives and fit different models in order to find the best one.

2.3 Queueing theory

Queueing theory is the mathematical study of waiting lines, or queues. A queueing model is constructed so that queue lengths and waiting times can be predicted by the traffic and service levels. There are various types of queueing models, with different constraints and properties. The *Kendall's notation*, proposed in 1953 by Kendall [4], is the standard system used to describe and classify these models using four factors, written as $A/S/c/K$, where:

- A denotes the time distribution between arrivals to the queue;
- S denotes the service time distribution;
- c denotes the number of available servers;
- K denotes the maximum capacity of the queue. If not specified it is assumed $K = \infty$.

The best representation of a supermarket queue is given by the $M/M/c$ model, where M denotes a *Markovian process*, meaning that the inter-arrival times and the service times of the c servers are exponentially distributed. The customers are served in FCFS order (First Come First Served).

The *utilization factor* $\rho = \lambda/c\mu$ describes the proportion of total service capacity being used in the system, where:

- λ is the average arrival rate;
- μ is the average service rate of a single server;
- c is the number of available servers.

If $\rho \geq 1$, i.e. the arrival rate exceeds the total service capacity, the queue will grow indefinitely, but if $\rho < 1$, the system is considered stable and the steady-state average queue length can be calculated.

The main issue with this stationary approach in the context of supermarkets is that the arrival rate and the number of available terminals cannot be expressed by probability distributions with constant mean, since they are strongly time-dependent and therefore non-stationary. We can denote such time-varying model as $M(t)/M/c(t)$, where the service rate is instead considered to be time-independent. Moreover, in a real situation it is possible for the utilization factor to exceed 1 for a short period of time (called *overloading*), in which the queue length increases, and after that, when $\rho < 1$, the system returns slowly to its equilibrium.

A common approach for dealing with time-varying rates is the *Stationary Independent Period by Period* (SIPP) approach, where the analysis is conducted on small intervals considered independently. For each interval, a different $M/M/c$ model with constant arrival rates and constant number of available servers is created and solved with the stationary approach. However, in 2001 Green et al. [17] showed that the commonly used SIPP approach is inaccurate for parameter values corresponding to many real situations, even when the time-dependent variations are small and especially for systems which operate near the critical load. Moreover, this stationary analysis requires the arrival rate to be strictly smaller than the service rate, i.e. $\rho < 1$ must hold for every interval, while many real systems can be temporarily overloaded.

Green et al. [16] proposed an easy-to-compute approximation for determining long run average performance measures for multi-server queues with periodic arrival rates: the *Pointwise Stationary Approximation* (PSA). This approximation was obtained by integrating over time, that is taking the formula for the stationary performance measure with the arrival rate that applies at each point in time. They empirically showed that this approximation is a tight upper bound for the true number of customers in the queue and that it is very accurate under certain parameters values. However, again, the PSA approach can only be applied to systems where $\rho < 1$ hold, i.e. temporal overloading in some intervals is strictly forbidden [15].

In 2008, Stolletz [15] proposed an improvement of the SIPP approach, the *Stationary Backlog-Carryover* (SBC) approach. This method was designed for systems with temporal overloading, thus the models constructed for every consecutive period are no longer independent from each other. Contrary to the SIPP approach that independently applies a $M/M/c/\infty$ model to each interval, SBC utilizes

a $M/M/c/c$ model, also known as the *Erlang's B model*, in which a maximum number of c customers can be in the queue at any time, and any further arrival to the queue is considered *blocked* (i.e. lost). These blocked customers are then carried over into future periods. This is obtained by defining an *artificial arrival rate* $\widetilde{\lambda}(t)$ that consists of both the original arrival rate $\lambda(t)$ and a *backlog rate* $b(t-1)$ of the previous period, that is the rate of blocked customers leaving the $M/M/c/c$ system at time $t-1$. This artificial arrival rate can be then used with a modified $M/M/c/\infty$ model to determine the average queue length.

Once the average queue length has been determined with one of the aforementioned methods, the average waiting time in the queue can be calculated using *Little's Law*, proposed by Little in 1961 [2]. This law states that the average number of customers in a stationary system Ls is equal to the arrival rate λ multiplied by the average time spent by the customers in the system Ws:

$$Ls = \lambda \cdot Ws \tag{2.1}$$

This equation is applicable only with stationary systems, but it can be extended to non-stationary systems with the SBC approach by using the artificial arrival rate $\widetilde{\lambda}$ instead of λ.

3 Data analysis

This chapter illustrates the analyses conducted to get a better understanding of the nature of the problem and the available data. The first section describes the two datasets used in these analyses and the differences between them. The following sections present a series of considerations on the distributions and patterns found in the data and on the possible applications of these findings for reaching the objectives of this thesis.

3.1 Datasets description

The datasets were generated by the RetailerIN system, already described in Section 1.2. RetailerIN is able to track, collect and analyze the movements of carts and baskets utilized by the customers. For every shopping session of a customer using a tracked asset, different information are collected and organized:

- a unique *session identifier*;
- an *asset identifier* that can be used to trace the physical asset used. There are two types of available assets: *carts* and *baskets*;
- the *start* and *end* time of the session, i.e. when the customer enters and leaves the shop;
- the total *dwell time* of the session (in seconds), i.e. the total time spent by the customer in the shop;
- an estimate of the total *movement time* and *stationary time*, such that movement time + stationary time = dwell time;
- the total *distance* covered while shopping (in meters);
- the *terminal number* in which the payment is made. There are two types of terminal: classical *assisted rolling belt* terminals and *self-checkouts*.

The datasets are exported in CSV format (Comma Separated Values). All the collected information are completely anonymous and not traceable to a specific customer.

Two distinct datasets were analyzed from two distinct supermarkets in which RetailerIN is running, that we would call *supermarket A* and *supermarket B*. Supermarket A is a large-sized North American retail store, while supermarket B is a medium-sized store in northern Italy. Each dataset covers different time periods and have some additional information that are not available for the other:

- *Dataset A* covers a six-month period, from the 1st of September 2019 to the 29th of February 2020, for a total of 235788 unique shopping sessions. Additional data about the products bought in each session was available, in particular:
  - the *total value* of the items bought (in dollars);
  - the *total number* of items bought, also called the *basket size*.
- *Dataset B* covers a two-month period, from the 6th of February to the 25th of March 2020, for a total of 32685 unique shopping sessions. Additional data about the length of every waiting queue and the average waiting times for each counter was available with a one-minute frequency. Specifically, for each interval, the measures available were:
  - the *average number of customers in queue*, that is the average number of customer simultaneously waiting in the queue;
  - the *total number of customers in queue*, that is the total number of customers that have joined the queue;
  - the *total number of completed sessions*, that is the total number of customers that completed the purchase and left the queue;
  - the average *time in queue* spent by the customers.

The session dataset was binned in a 10-minute interval. This interval size was chosen because it gave a good compromise between accuracy and frequency: smaller intervals would have included too much noise and randomness to obtain significant results, while larger intervals would have not offered the granularity needed to implement a fast response to sudden changes in traffic levels. In this chapter only the analyses conducted on the dataset A are shown, since the greater amount of available data gave more insightful results. The dataset B is used when the forecast accuracy of the predictive models is discussed in Chapter 5.

3.2 Dwell time, inflow rate and service time distributions

First, the measured inflow rates and dwell times were grouped by day and hour to get an overview of how the customers' behavior changes during the week.

Since carts and baskets are used by different types of customers with different shopping behaviors, they were considered separately. The heatmaps in Figure 3.1 and 3.2 show the obtained results, confirming that carts and baskets are indeed used in very different ways. Moreover, it is already possible to identify an initial cyclic behavior: for example, the carts present an higher usage during weekends. These seasonal cycles are further analyzed in the next section.

As explained in Section 2.3, a lot of queueing theory models assume that the parameter values follow different specific probability distributions. The following distributions analyses were conducted to verify the applicability of these models.

First, the distribution of the dwell time, shown in Figure 3.3, was investigated and approximated with an *Erlang distribution*. Again, carts and baskets were considered separately for the same reasons stated previously.

For the inflow rate, the distribution of the *inter-entry time* shown in Figure 3.4 was calculated. The inter-entry time is the time passed between the entry of two consecutive customers. In this case, the distribution can be approximated by an *exponential distribution*.

Finally, the distribution of the checkouts' service time was investigated. Since there are no available measurements of the actual service time, the *basket size* n, i.e. the total number of products bought in each session, was used as an approximation. The service time S can be defined as a combination of:

- a *service time per item* S_{item};
- an *extra service time* S_{extra}, to take in account possible additional time spent at the checkouts, for example to complete the payment;

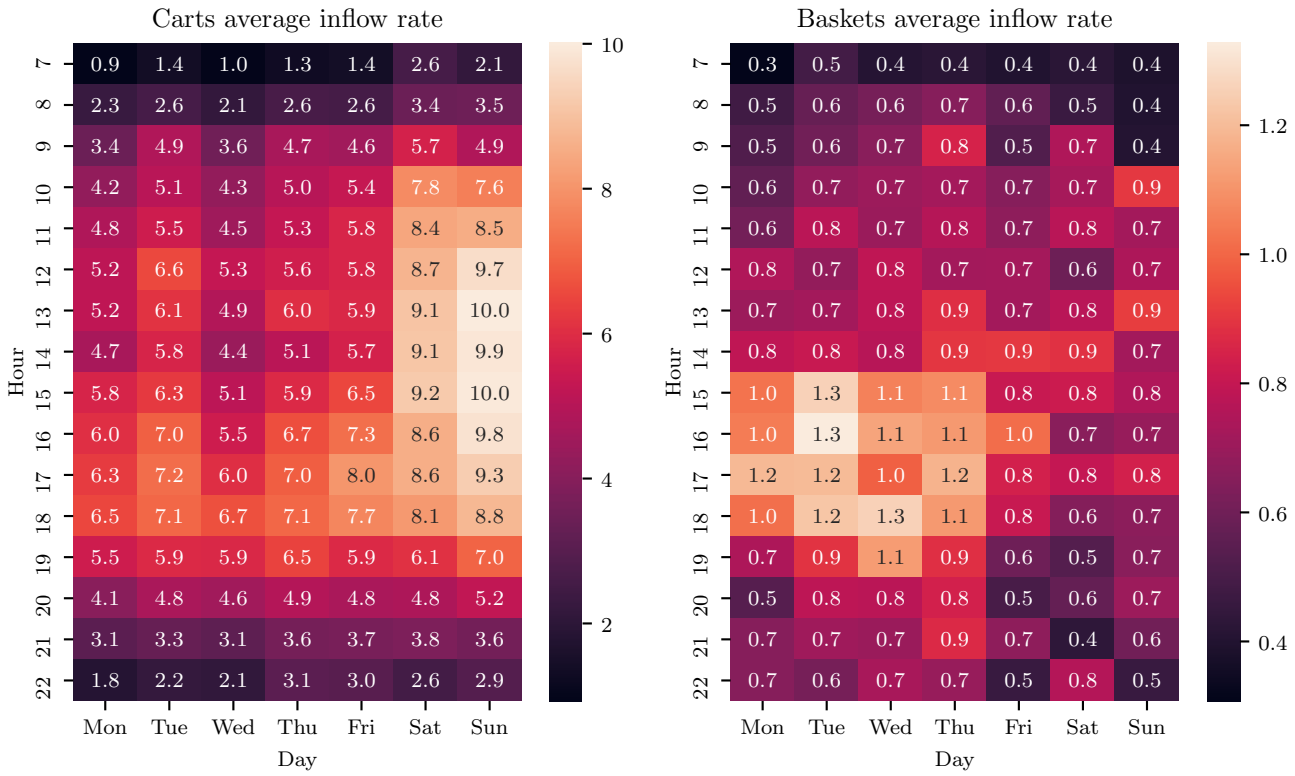

Figure 3.1: The heatmap showing the distribution of the inflow rates, in customers/10min.

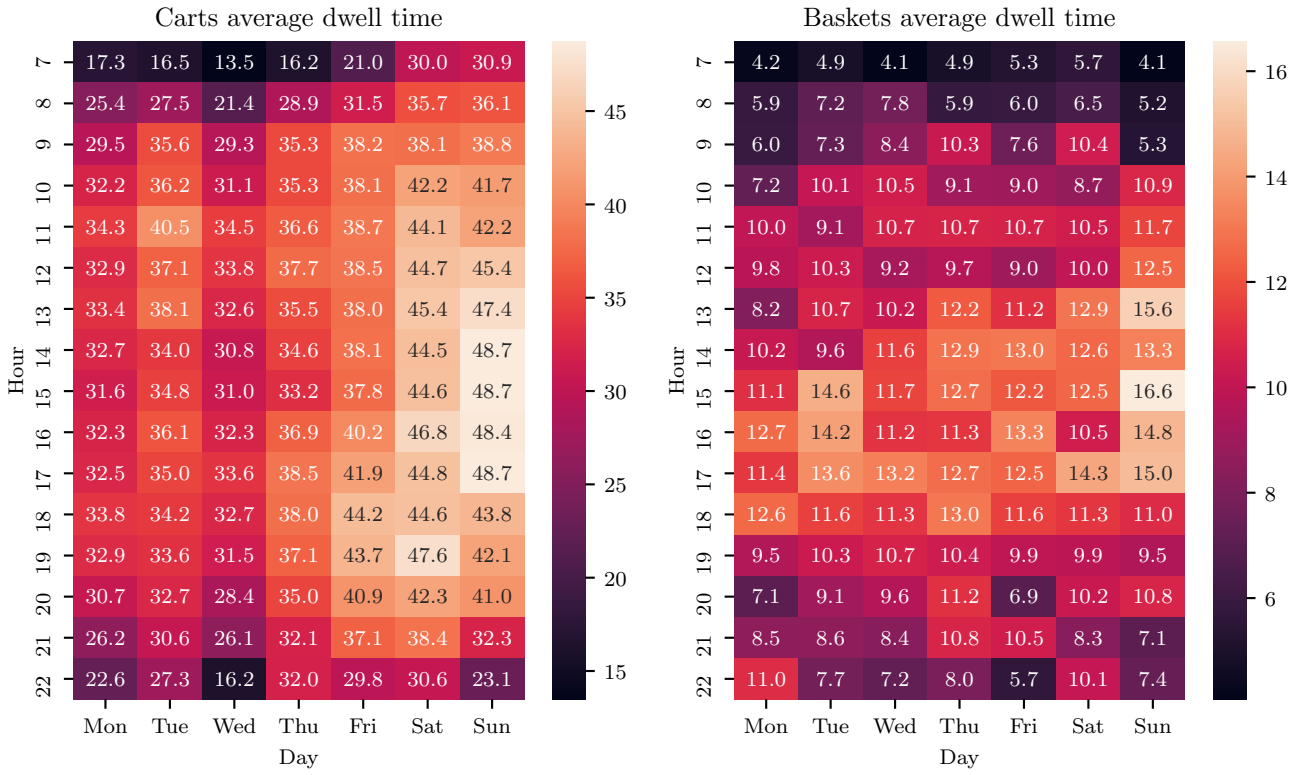

Figure 3.2: The heatmap showing the distribution of the dwell times, in minutes.

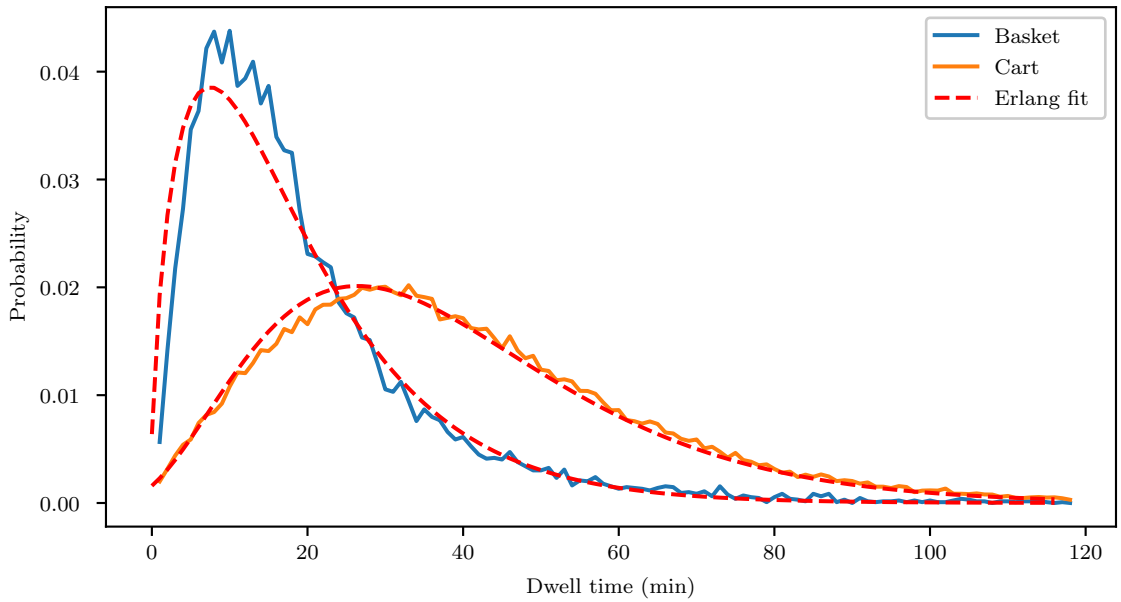

Figure 3.3: The probability density function of the dwell time, with the corresponding Erlang distribution fit (in red).

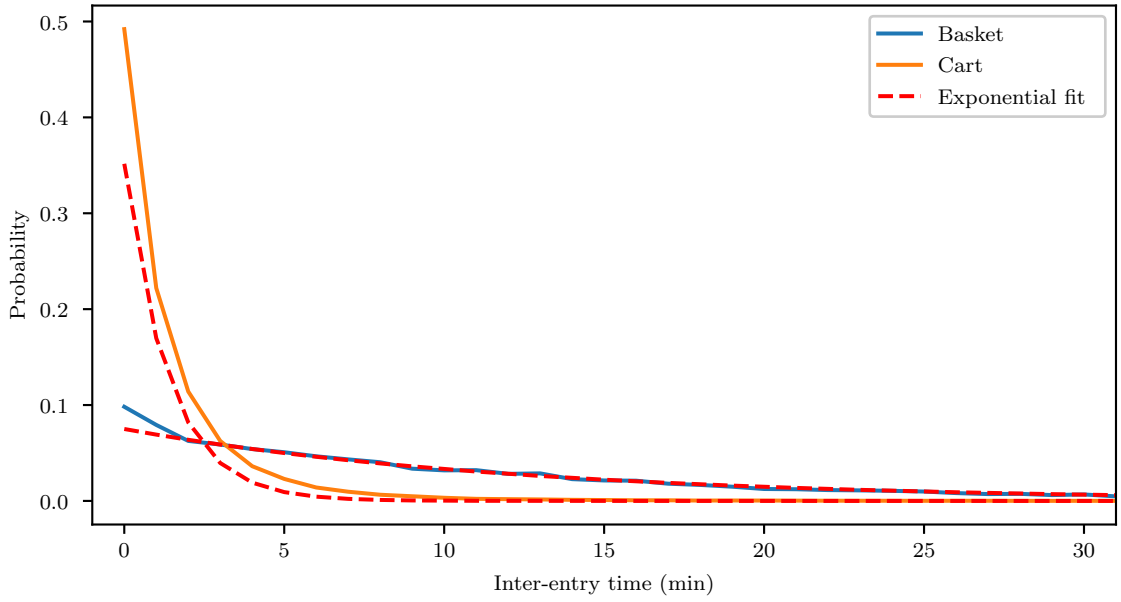

Figure 3.4: The probability density function of the inter-entry time with the corresponding exponential distribution fit (in red).

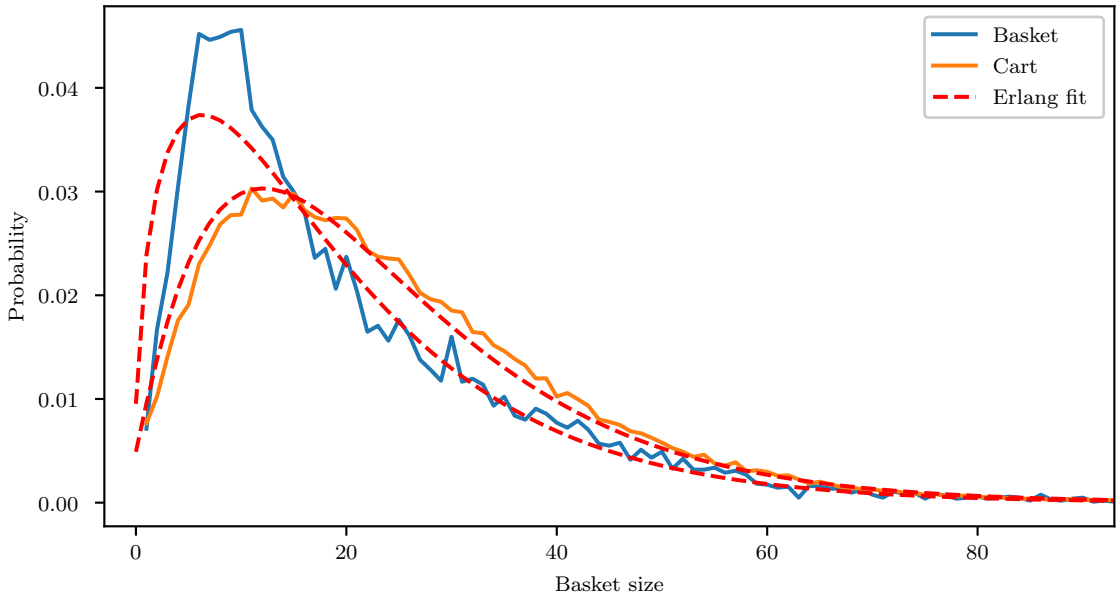

Figure 3.5: The probability density function of the basket size with the corresponding Erlang distribution fit (in red).

and we can write:

$$S = nS_{item} + S_{extra} \tag{3.1}$$

How the approximated values for S_{item} and S_{extra} are calculated is described in the next chapters. Since we considered constants S_{item} and S_{extra}, the service time distribution follows the basket size distribution shown in Figure 3.5.

3.3 Time series analysis

Different time series analysis techniques were used to extract the main components of the time series, such as trends and seasonalities. With this decomposition it was possible to verify whether additional data preprocessing were needed and to identify specific patterns and features of the time series that could be exploited by the forecasting process.

3.3.1 Inflow rate autocorrelation

In order to identify the number of previous values that directly influence the current value, the *autocorrelation* of the inflow rate was calculated. This measurement is useful to identify a first set of relevant features that could be used in a predictive model.

Autocorrelation refers to the degree of correlation between the values of the same variable across different past observations in the data, and it measures the linear relationship between a variable's current value and its past values. The autocorrelation values can range from -1 to +1, where +1 represents a perfect positive linear relationship (an increase in one of the values corresponds to a proportional increase in the other), -1 represents a perfect negative linear relationship, and 0 represents the absence of any linear relationship.

When data have a trend, the autocorrelations at small lags tend to be large and positive because observations nearby in time have also similar values. Therefore, the *autocorrelation function* (ACF) of a trended time series tend to have positive values that slowly decrease as the lags increase. The inflow rate and dwell time trends are investigated in the next section.

When data have a seasonality, the autocorrelations will be larger for the seasonal lags (multiples of the seasonal frequency) than for the other lags [7]. With the analysis of the inflow rate's ACF shown in Figure 3.6, it is clear that the time series presents two main seasonalities:

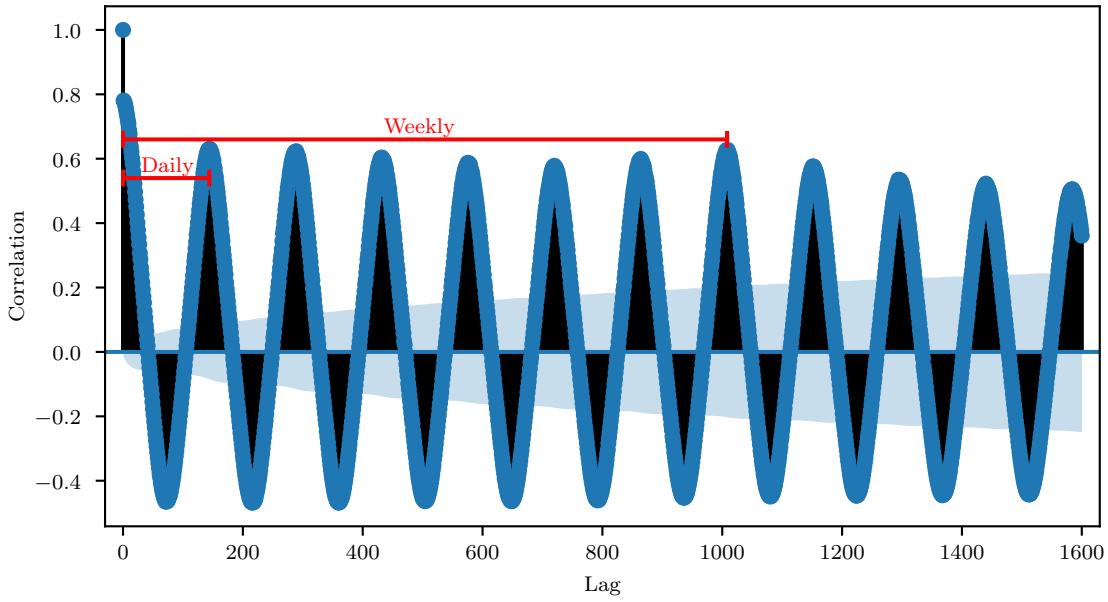

Figure 3.6: The inflow rate's ACF with the daily and weekly seasonalities.

- a *daily seasonality*, with a cycle length of ∼144 lags (∼24 hours), meaning that the inflow's traffic peaks are repeated every day at approximately the same hours;
- a *weekly seasonality*, with a cycle length of ∼1008 lags (∼7 days), meaning that each day of the week has a distinct traffic pattern that is repeated every week.

The *partial autocorrelation* (PACF) is the autocorrelation between two values after removing any linear dependency on the values between them. For example, the autocorrelation at lag 1 is the coefficient of correlation between y_t and y_{t-1}, which is probably also the correlation between y_{t-1} and y_{t-2}. But if y_t is correlated with y_{t-1}, and y_{t-1} is equally correlated with y_{t-2}, then we should expect to find correlation also between y_t and y_{t-2}. The partial autocorrelation excludes this mutual relation from the correlation coefficient, and, at lag k, it can be described as the correlation between y_t and y_{t+k} after removing the effects of $y_{t+1}, ..., y_{t+k-1}$. The analysis of the partial autocorrelation function, shown in Figure 3.7, is useful to get an insight about the number of lagged values that should be considered by a predictive regression model. Moreover, a positive autocorrelation denotes a *non-stationary* time series, that is a time series whose statistical properties change over time, i.e. the trend and seasonality affect the value of the time series at different times. More precisely, if $\{y_t\}$ is a stationary time series, then for all s, the distribution of $(y_t, ..., y_{t+s})$ does not depend on t [7]. The majority of the forecasting methods presented in Section 2.2 are based on the assumption that the time series is stationary, so the data must be *differenced* before training the model to ensure stationarity. The differencing process used is explained in details in Section 4.1.3.

3.3.2 Time series decomposition

Decomposing the time series into its main components is useful to find additional patterns and to validate the previous results. While in the previous section two different seasonal cycles were determined, the main goal of this section is to identify any possible trend variation among different seasons that should be taken in account when designing the final predictive model. As described in Section 2.2, there are two types of decomposition: *additive* and *multiplicative*. Since the variations around the seasonal components are not proportional to the level of the time series, the additive decomposition was chosen. First, the *trend* of the inflow rate was calculated using a moving window average with a size of 1008 intervals (that is equal to one week of observations). As shown in Figure 3.8, the general trend seems to be stable among different months, except for September where there seems to be a greater inflow traffic. With more data available, it should be possible to identify additional monthly or yearly seasonal cycles that could be exploited to further improve the forecasting model accuracy.

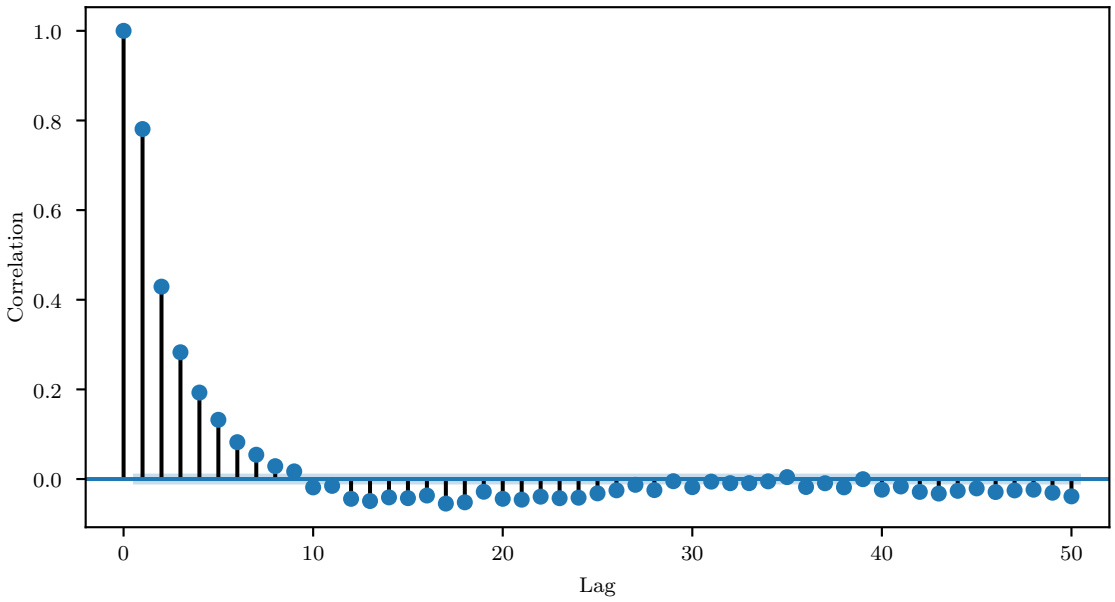

Figure 3.7: The inflow rate's PACF. The positive correlation values denote a non-stationary time series.

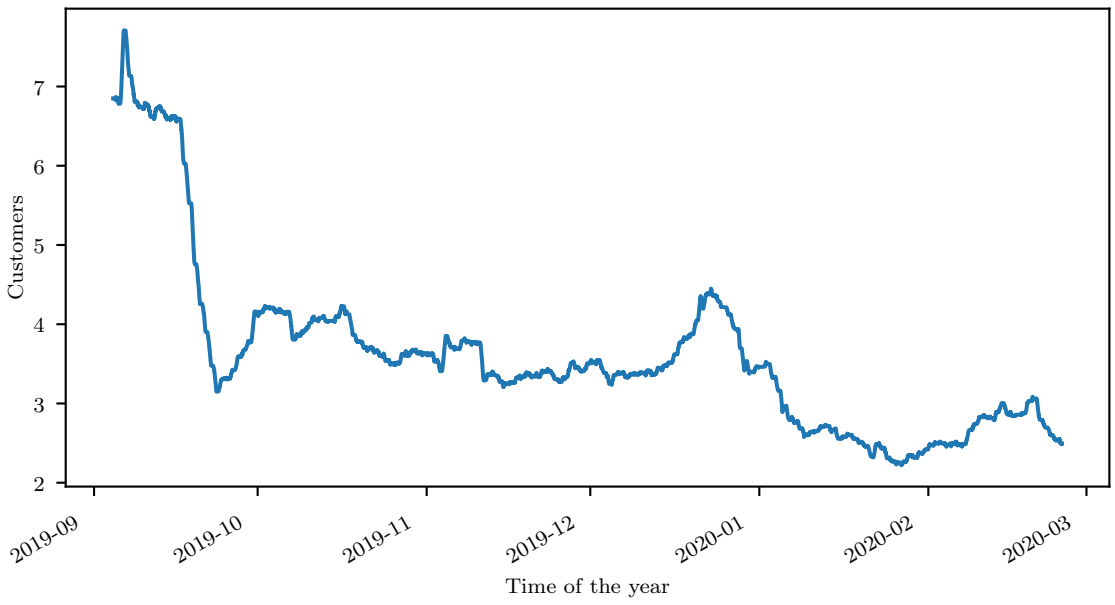

Figure 3.8: The inflow rate's trend.

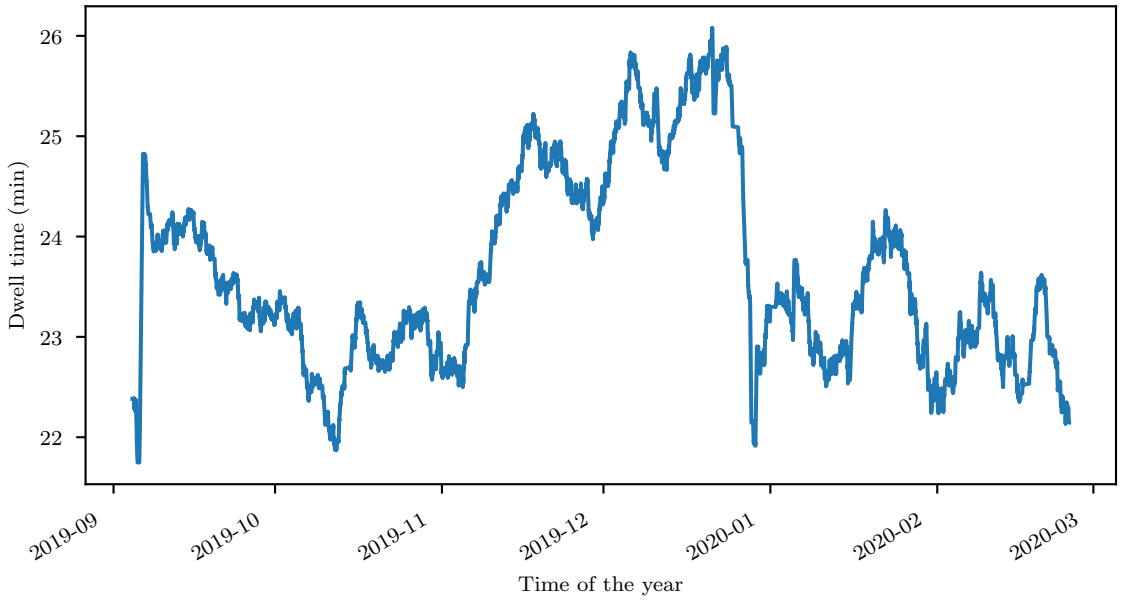

Figure 3.9: The dwell time's trend.

The same analysis was then conducted for the dwell time (see Figure 3.9). The dwell time's trend is not as stable as the inflow rate's, since a slightly increase is visible during November and December.

3.3.3 Service time approximations

As described in Section 3.2, an approximation of the checkouts' service time S can be obtained using the basket size n and Eq. (3.1): $S = nS_{item} + S_{extra}$. However, this approximation is feasible only if the number of products bought in each session is available to RetailerIN. Unfortunately, that is not the case for every store, so other approximations were investigated.

An alternative approach is to use the average number of customers served by each terminal. Since the service rate should represent the maximum service capacity of a terminal, the periods where the counters spent some time in *idle*, i.e. without serving any customer, had to be excluded from this approximation. Therefore, only the intervals with a measured queue length greater than a predetermined threshold were considered. This threshold was determined by analyzing the measured *outflow rate* and comparing it to the *total service rate* (where total service rate = service rate·number open terminals). When the two measurements have similar values, the maximum service capacity has been reached and the checkouts have probably spent no time in idle, thus the number of customer served in that interval should give an accurate approximation of the actual service rate. However, again, this approach is implementable only if the dataset contains the queues' status.

In order to get an approximation that could be used in every setting, with data that is available independently from the store, the relation between the service and dwell times was researched. As shown in Section 3.2, both measurements follow an Erlang distribution. Intuitively, a longer dwell time should correspond to a greater amount of products bought (i.e. a greater basket size) and therefore a longer service time. However, this hypothesis was confuted by Berman et al. [11], which showed that there is no such relation between these two measurements, and by the correlation coefficient, that resulted in a value of approximately 0.24. This is probably because the majority of the dwell time is spent looking and deciding, and there could be other unrelated factors that influence the service time, such as the use of coupons and credit cards during the payment.

The last approach presented was given by the analysis of the time spent in idle (t_i) by each terminal. To do so, the *inter-exit times* Δt_e were calculated, that are the times passed between the consecutive exits of two customers served by the same counter:

- if $\Delta t_e > S$, the checkout has probably spent some time in idle, in particular $t_i = \Delta t_e - S$. If $\Delta t_e \gg S$, the checkout is probably closed for that interval;

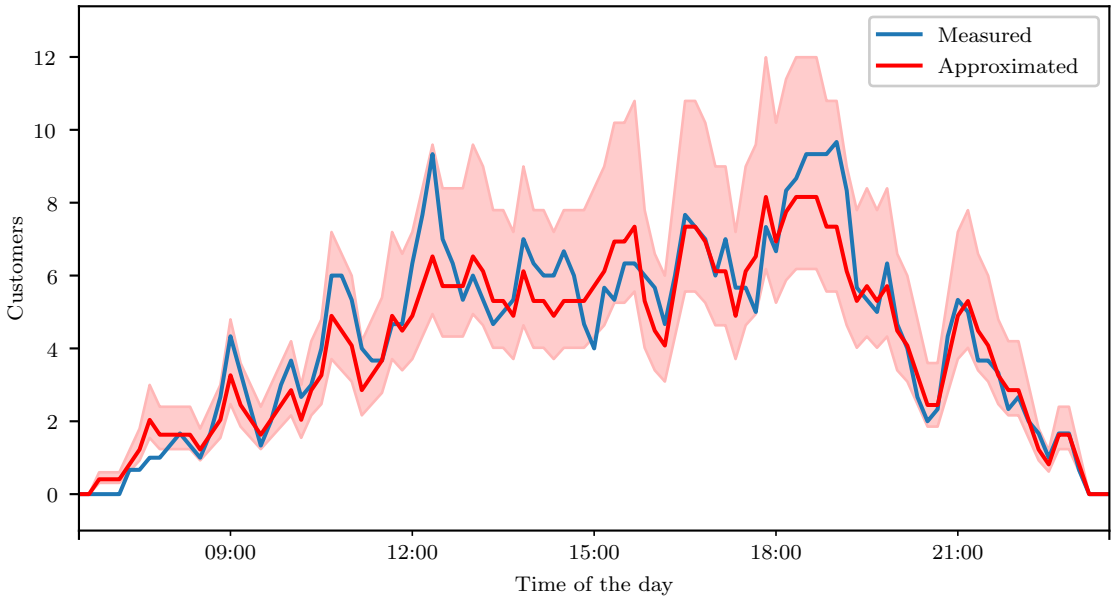

Figure 3.10: Example of the measured outflow rate with the corresponding service-rate-based approximation.

- otherwise, if $\Delta t_e \leq S$, by the time the session's purchase has been processed by the checkout, another customer is ready to be served, and therefore $t_i = 0$. In this case there is probably a waiting queue for that counter.

The hours with peak outflow traffic, i.e. with minimum inter-exit times, were taken in consideration to approximate the value of S. This is because when the inter-exit times are low, there is probably a queue and consequently the checkouts should have spent no time in idle, meaning that $t_i = 0$ and $S = \Delta t_e$. If the data about the products bought in each session is available, the approximated service time S thus obtained could be used to determine the values of S_{item} and S_{extra} introduced previously.

3.3.4 Service rate and open terminals count approximation

Once an approximation of the service time S has been determined, the *service rate* μ, that is the number of customers served by a single terminal in a time interval Δt, can be obtained as:

$$\mu = \frac{\Delta t}{S} \tag{3.2}$$

Intuitively, the total service rate should be directly influenced by the number of open terminals, considering that with more terminals the total service capacity increases. For this reason, the relation between these two measures was investigated. Since a direct measurement of the total service rate was not available, and it is equal to the total number of customers that leave the checkout area and then after a small delay exit the store, the *outflow rate* was used as approximation. The correlation between the two measures resulted in a value of approximately 0.92, meaning that there is a strong linear relationship between them. To further verify this relation, the mean of the service rates per terminal $\bar{\mu}$ was used to obtain the outflow rate $o(t)$, as:

$$o(t) = c(t) \cdot \bar{\mu} = c(t) \cdot \frac{\Delta t}{\bar{S}} \tag{3.3}$$

where $c(t)$ is the number of open terminals at time t. A comparison between the approximated and real outflow rate can be seen in Figure 3.10. The accuracy of the results shows that a time-independent service time can be considered, since a constant value for S gives a good approximation of the actual outflow.

4 Solution

This chapter describes the methods used to build the final predictive model. As stated previously, the main goal of this research is to optimize the number of open checkouts while maintaining the waiting times under a predetermined threshold. In order to reach it, the final model is composed of different submodules that cooperates sequentially, so that the results of a submodule are used as input to the next one, as shown in Figure 4.1. First, the *measured inflow rate* of the recent past is combined with a *forecasted inflow rate* of the immediate future. These values, combined with a *dwell time forecast*, are used to obtain a prediction of the *arrival rate at the checkouts*. The arrival rate, the number of *open counters* and the approximated *service rate* are then processed with queueing theory techniques to calculate the expected queue length for a given interval. Finally, different values for the number of available checkouts can be tested to obtain the minimum staff allocation that ensures that the predefined maximum queue length limit is not exceeded.

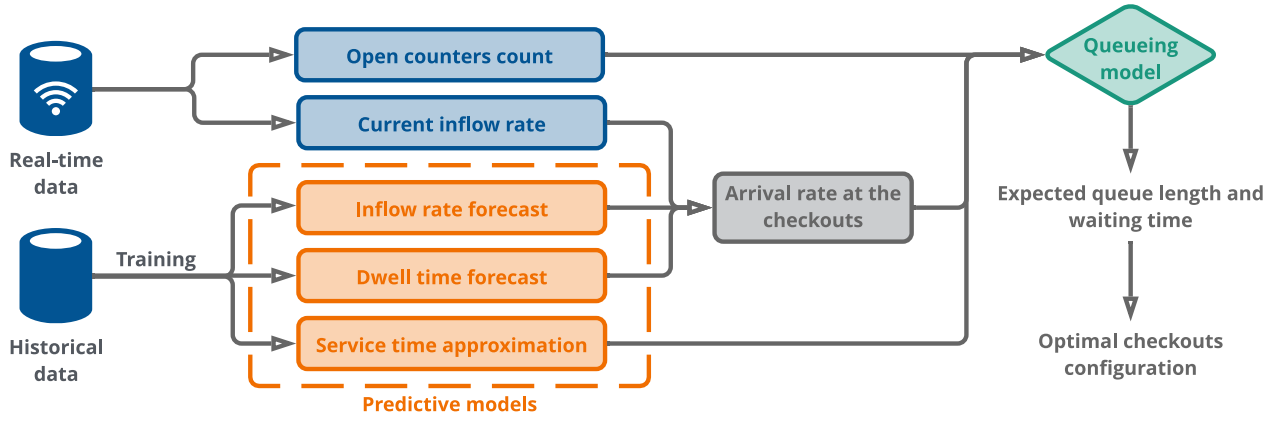

Figure 4.1: The final architecture of the predictive model.

4.1 Inflow rate forecast

Since the inflow rate can be expressed as a time series, different time series predictive models, already presented in Section 2.2, were tested. This section discusses the implementation details of these models, while a comparison of the obtained results and performances is presented in Section 5.2

4.1.1 Persistence model

The first model presented was used only to define a performance baseline. This gave an idea of how well the other models could perform on the data and set a minimum accuracy level: if a model's performances were worse than this baseline, it was discarded and not considered. This baseline was obtained by using a *persistence model*, where the last measured value y_{t-1} is simply used as forecast for the next time interval t:

$$\hat{y}_t = y_{t-1} \tag{4.1}$$

4.1.2 Drift model

As seen in Chapter 3, the inflow rate time series presents a strong seasonal component, so basing the forecast on this repeating pattern could results in a good accuracy. The forecast values were calculated as the average of the values in the previous weeks from the same week day and time, as:

$$\hat{y}_t = \hat{f}(t) = \frac{1}{N}\sum_{i=1}^{N} y_{t-iW} \tag{4.2}$$

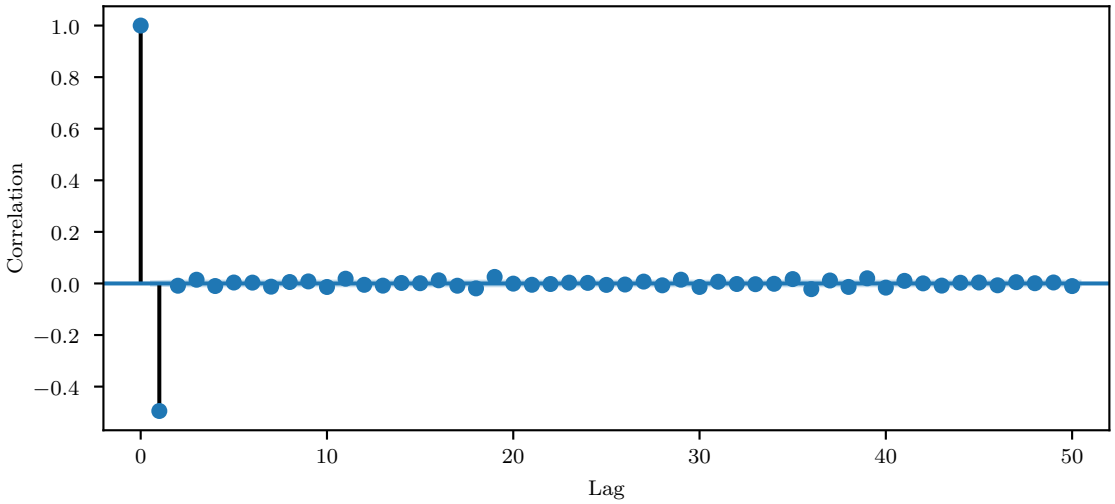

Figure 4.2: The ACF of the stationarized inflow rate time series. In comparison to the non-stationary ACF (see Figure 3.6), there is no visible seasonality.

where W is the number of intervals in a week and N is the number of previous weeks considered. Since we used 10-minute intervals, $W = 1008$. This method is accurate only if the cyclic patterns maintain the same levels every day, without any random variation.

To take in account these variations in the inflow rate, that can be caused by external factors like holidays or special promotions, a *local drift* was calculated using the errors of the forecasted values for the immediate past:

$$\hat{y}_t = \hat{f}(t) + \frac{1}{M}\sum_{i=1}^{M} y_{t-i} - \hat{f}(t-i) \tag{4.3}$$

were M is the number of previous steps to be considered by the drift. We can denote this model as Drift(N, M). By doing this, the forecasts are progressively adjusted to the current traffic level while still taking in consideration the seasonal cyclic behavior.

4.1.3 Autoregressive Neural Network model

As stated in Section 2.2.2, artificial neural networks are a popular and effective technique for time series forecasting. In the same section, *autoregressive neural networks* were introduced with the notation AR-NN$(p, P, k)_m$.

Since the inflow rate's values present a strong correlation with the previous values, as seen in Section 3.3.1, the time series is non-stationary, and therefore the dataset must be rendered stationary before training the model. This was done by *differencing* the time series, i.e. by calculating the difference between consecutive observations (also called *first-order differencing*). Each value of the differenced time series can be written as $y'_t = y_t - y_{t-1}$. This transformation can help stabilize the mean of the time series, reducing the effects of trend and seasonalities. Since the time series present a strong seasonal component, a seasonal differencing was also applied, written as $y'_t = y_t - y_{t-m}$, where m is the number of lags in a seasonal cycle. A daily seasonal cycle was taken in account, so $m = 144$. Both differentiations were applied to obtain stationarity, thus each observation of the final time series used to train the model was calculated as:

$$y''_t = y'_t - y'_{t-1} = (y_t - y_{t-m}) - (y_{t-1} - y_{t-m-1}) \tag{4.4}$$

The ACF analysis of the differenced time series in Figure 4.2 shows that this approach is effective in making the time series stationary.

As stated in Section 2.2.2, given a target value y_{t+1}, each sample in input can be written as:

$$(y_t, y_{t-1}, ..., y_{t-p}, y_{t-m}, y_{t-2m}, ..., y_{t-Pm})$$

The model used an MSE (Mean Squared Error) loss function and a *sigmoid* activation function. The ANN was trained over a batch size of $24 \cdot 6 = 144$ samples, that is the number of observations in a day. The training was then repeated for a total of 100 epochs. Various parameters values for p, P and k were tested and the model with the best performance was used. The obtained results and accuracy are discussed in Section 5.2.

4.1.4 SARIMA model

In Section 2.2.3 the ARIMA model and some of its evolutions were introduced. Since the inflow rate presents a strong daily seasonal pattern, as seen in Section 3.3.1, the SARIMA model was chosen as a possible forecasting method. This model has different parameters that must be configured correctly in order to achieve a good prediction accuracy:

- p: autoregressive order;
- d: degree of differencing;
- q: moving average order;
- P: seasonal autoregressive order;
- D: seasonal degree of differencing;
- Q: seasonal moving average order;
- m: number of time steps in the seasonal period.

The parameters configuration can be written as SARIMA$(p, d, q)(P, D, Q)_m$. For some of these parameters, the optimal values can be determined by the analysis of the time series, while the only possible approach for the others is to try different values and select the ones that minimize the forecasting errors. By the results obtained in Chapter 3, we can set:

- $d = 1$ and $D = 1$ for respectively first-order and seasonal differencing. By doing this, we achieve the same type of differencing explained in the previous section;
- $q = 6$ and $Q = 3$, obtained by the PACF analysis of the differenced time series. Specifically, the plot shows an exponential decay on the seasonal lags of the PACF (e.g at lags 144, 288, ...). For this reason, the number of significant autocorrelation coefficients gives a good approximation for q and Q [7];
- $m = 144$, that is the daily seasonal cycle length, i.e. the number of 10-minute intervals in a day.

In 2007, Hyndman et al. [8] proposed the *Hyndman-Khandakar algorithm*, which combines unit root tests, minimization of the *Akaike's Information Criterion* (AIC) and *Maximum Likelihood Estimation* (MLE) to automatically determine the best parameters for the SARIMA model. While this method is very computational expensive and does not always returns the best configuration, it removes a lot of the steps and trials involved in the parameters determination, simplifying the whole process. By using this technique, the parameters values presented previously were confirmed to be the optimal ones. Moreover, the results showed that by using $p = 0$ and $P = 0$ the best accuracy was obtained.

4.2 Arrival rate forecast

Once a predictive model was defined, the measured and forecasted inflow rates were combined and used to get a prediction of the *arrival rate* at the checkouts. Since the inflow rate into the shop appears with a delay at the checkout area, and this delay is the customer's dwell time, a time-dependent probability density function $p_t(\tau)$ is required, where τ is the expected dwell time of a customer [6].

In Section 3.2 the general distribution of the dwell time was presented and, with the analysis of Section 3.3, it is clear that it is strongly time-dependent, thus it can be seen as a stochastic process $\{X_t\}$ with $X_t \sim \text{Erlang}(k_t, n_t)$, where $k_t = \text{E}[X_t]^2/\text{Var}[X_t]$ and $n_t = \text{E}[X_t]/\text{Var}[X_t]$. Given the weekly and daily seasonality of the time series, different X_t distributions were defined for each interval on

every week day, for a total of $7 \cdot 24 \cdot 6 = 1008$ distributions, since 10-minute intervals were used. Each aforementioned distribution was obtained by calculating the values for $\mathrm{E}[X_t]$ and $\mathrm{Var}[X_t]$ from the observations of the previous weeks on the same interval and week day.

Let $\lambda(t)$ be the *predicted arrival rate* for the time interval t, Δt the intervals size, $p_t(\tau_i)$ the probability of having a dwell time of τ_i, with $\tau_i =](i-1)\Delta t,\ i\Delta t]$, and y_{t-i} the *measured inflow rate* at time $t - i$. We can approximate $\lambda(t)$ by:

$$\lambda(t) = \sum_{i=1}^{\infty} y_{t-i} \cdot p_t(\tau_i) \tag{4.5}$$

Since after a certain τ_i value, $p_t(\tau_i)$ becomes irrelevant, we can limit the number of values to be considered by defining a maximum dwell time τ_{max}, that can be set either by a customizable parameter or by analyzing the dwell times distribution (e.g. by using the *95th percentile*). In any case, τ_{max} should be such that $p_t(\tau_{max} + \Delta t)$ tends to zero. We can then rewrite the previous formula as:

$$\lambda(t) = \sum_{i=1}^{K} y_{t-i} \cdot p_t(\tau_i) \tag{4.6}$$

where K is such that $\tau_K \leq \tau_{max} \leq \tau_{k+1}$. This equation is however applicable only if the measured values for $y_{t-1}, y_{t-2}, ..., y_{y-K}$ are known, thus only when $t \leq t_{now}$, where t_{now} is the current interval when those calculations are executed. If $t > t_{now}$, a forecast of the inflow rate, as described in Section 4.1, must be used to provide the missing information. For this reason, we define a function $\mathrm{in}(t)$ as:

$$\mathrm{in}(t) = \begin{cases} y_t & \text{if } t < t_{now} \\ \hat{y}_t & \text{otherwise} \end{cases} \tag{4.7}$$

where y_t and $\hat{y}_t$ are respectively the measured and predicted inflow rate at time t. It shall be noted that the real measurements for an interval are available only once the said interval is ended, hence the reason for using y_t only when $t < t_{now}$. The Equation (4.6) is therefore rewritten as:

$$\lambda(t) = \sum_{i=1}^{K} \mathrm{in}(t-i) \cdot p_t(\tau_i) \tag{4.8}$$

To obtain a *multi-step forecast* of the next N intervals, we can directly solve a linear system of equations:

$$\begin{bmatrix} \lambda(t) \\ \lambda(t+1) \\ \vdots \\ \lambda(t+N) \end{bmatrix} = \begin{bmatrix} \mathrm{in}(t-1) & \mathrm{in}(t-2) & \cdots & \mathrm{in}(t-K) \\ \mathrm{in}(t) & \mathrm{in}(t-1) & \cdots & \mathrm{in}(t-(K-1)) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{in}(t+(N-1)) & \mathrm{in}(t+(N-2)) & \cdots & \mathrm{in}(t+(N-K)) \end{bmatrix} \begin{bmatrix} p_t(\tau_1) \\ p_t(\tau_2) \\ \vdots \\ p_t(\tau_K) \end{bmatrix} \tag{4.9}$$

Section 5.3 describes the results obtained and the accuracy of this predictive model.

4.3 Queue length forecast

This section describes the model implemented for forecasting the queue length at the checkouts. As described in Section 2.3, the required variables are:

- $\lambda(t)$: arrival rate at the checkouts;
- $\mu(t)$: service rate of a single checkout;
- $c(t)$: total count of available checkouts (servers), also referred to as the *configuration* of the checkouts.

A forecast of the arrival rate $\lambda(t)$ has been proposed in the previous section and an approximation

of the service rate $\mu(t)$ is presented in the next section. The total number of open checkouts $c(t)$ is the tunable parameter that can be used to evaluate the performance of different checkouts configurations.

4.3.1 Service rate approximation

In Section 3.3.3, various possible approximations for the service time were introduced. In Section 3.3.4, it was shown that a constant, and therefore time-independent, service rate could be used without losing precision. This constant service rate was obtained by investigating the time spent in idle by each terminal, which was determined by the analysis of the inter-exit times.

As described in Section 3.3.3, when the time spent in idle by the counters' staff becomes insignificant, i.e. when it approaches a value of zero, the customers' inter-exit times can give a good approximation of the actual time spent serving each customer in the queue. Since the idle time is not an available measurement, this was achieved by using the hours with peaks in the arrival rate at the checkouts as an indicator of low idle times. A threshold value λ_{max} was defined, such that when it is exceeded by the actual arrival rate, i.e. when $\lambda_t \geq \lambda_{max}$, the time t can be considered as a period with high traffic density, and therefore included in the aforementioned approximation. An average of the measured inter-exit times during these high traffic periods was then calculated and used as *service time* S of a single counter. The *service rate* μ was then obtained from S using Eq. (3.2). In Section 5.4 the results of this approximation are presented.

4.3.2 Queue length approximation

As described in Section 2.3, the standard queueing theory does not support time-varying arrival rates with temporal overloading. The only approach that seemed to fit the context of this research was the *Stationary Backlog-Carryover* (SBC) approach, proposed by Stolletz in 2008 [15]. SBC is based on a $M/M/c/c$ model, in which a maximum number of c customers can be in the queue at any time and any further arrival is considered *blocked* (i.e. lost), and uses an artificial arrival rate $\widetilde{\lambda}_t$ to take in account the temporal overloading of the queue, allowing standard queueing models to be used. $\widetilde{\lambda}_t$ consists of both the average arrival rate λ_t and the *backlog rate* of the previous period b_{t-1}, that is the rate of customers leaving the system due to blocking in the former period.

Let $P_t(B)$ be the steady-state probability of blocking for the $M/M/c/c$ model in period t with artificial arrival rate $\widetilde{\lambda}_t$. The backlog rate b_t is given by:

$$b_t = \widetilde{\lambda}_t \cdot P_t(B) \tag{4.10}$$

These blocked customers are carried over into the period $t+1$, which results in the artificial arrival rate $\widetilde{\lambda}_{t+1}$. Starting with $\widetilde{\lambda}_1 = \lambda_1$ and $b_0 = 0$, we can define the artificial arrival rate recursively through:

$$\widetilde{\lambda}_t = \lambda_t + b_{t-1} = \lambda_t + \widetilde{\lambda}_{t-1} \cdot P_{t-1}(B) \tag{4.11}$$

By doing this, the customers not served in period $t-1$ are evenly spread over period t. The probability $P_t(B)$ is obtained by the *Erlang's loss* formula:

$$P_t(B) = \frac{(\widetilde{\lambda}_t/\mu_t)^{c_t}}{c_t! \sum_{k=0}^{c_t} \frac{(\widetilde{\lambda}_t/\mu_t)}{k!}} \tag{4.12}$$

The expected servers utilization ρ_t can be then calculated as:

$$\rho_t = \frac{\widetilde{\lambda}_t(1 - P_t(B))}{c_t \mu_t} = \frac{\widetilde{\lambda}_t - b_t}{c_t \mu_t} = \frac{\lambda_t + b_{t-1} - b_t}{c_t \mu_t} \tag{4.13}$$

Stolletz proposed two different approximations for the expected number of customers in the queue, the first based on a *Modified Arrival Rate* (MAR) and the second based on ρ_t and b_t [15].

The MAR approach uses an $M/M/c/\infty$ queueing model, with the same utilization factor ρ_t of the $M/M/c/c$ model analyzed in the previous step. To do so, a modified arrival rate λ_t^{MAR} is chosen such that the model reach the approximated ρ_t obtained by Eq. (4.13), in particular:

$$\lambda_t^{MAR} = \rho_t c_t \mu_t \tag{4.14}$$

Applying Eq. (4.13) results in:

$$\lambda_t^{MAR} = \lambda_t + b_{t-1} - b_t \tag{4.15}$$

The expected number of customers in the system Ls_t can be then obtained by solving the specific formula for the $M/M/c/\infty$ model [9]:

$$Ls_t^{MAR} = c_t\rho_t + \frac{\rho_t}{1-\rho_t}\pi_{c_t^+} \tag{4.16}$$

where $\pi_{c_t^+}$ is the probability of having all the c_t servers occupied:

$$\pi_{c_t^+} = \frac{(c_t\rho_t)^{c_t}}{c_t!(1-\rho_t)}\pi_0 \tag{4.17}$$

and π_0 denotes the probability of having 0 customers in the system:

$$\pi_0 = \left[\sum_{k=0}^{c_t-1}\frac{(c_t\rho)^k}{k!} + \frac{(c_t\rho)^{c_t}}{c_t!}\frac{1}{1-\rho_t}\right]^{-1} \tag{4.18}$$

The second proposed approach approximates the expected queue length with the number of backlogged customers (*A1 approximation*). The backlog rate b_t is multiplied by the period length Δt to obtain the number of waiting customers at the end of period t. The expected queue length Lq_t^{A1} is then given by:

$$Lq_t^{A1} = b_t\Delta t \tag{4.19}$$

and the total number of customers in the system Ls_t^{A1} by:

$$Ls_t^{A1} = Lq_t^{A1} + c_t\rho_t \tag{4.20}$$

However, they showed that the expected queue length is often overestimated by the A1 approximation. An improvement was obtained by reducing the approximated Lq_t^{A1} value by the expected number of non-busy servers $c_t(1-\rho_t)$ (*A2 approximation*), such that:

$$Lq_t^{A2} = max\{0, Lq_t^{A1} - c_t(1-\rho_t)\} \quad \text{and} \quad Ls_t^{A2} = Lq_t^{A2} + c_t\rho_t \tag{4.21}$$

Once the number of customers in queue has been determined, Little's Law (2.1) [2] can be used to calculate the expected queue and system waiting times, respectively Wq_t and Ws_t, as:

$$Wq_t = Lq_t/\lambda_t^{MAR} \quad \text{and} \quad Ws_t = Ls_t/\lambda_t^{MAR} \tag{4.22}$$

The results and performances of each approximation are discussed in Section 5.4.

4.4 Checkouts optimization

In the previous section, various approximations for the average queue length were discussed. In this section, these approximations are used to compute the optimal number of checkouts that should be opened to maintain an acceptable service level while reducing the manned staff's idle time.

4.4.1 Optimal checkouts configuration

The optimal number of open checkouts c for the next future period $t+1$ should be as low as possible, but still enough to maintain the queue length under a predetermined acceptable threshold Ls_{max}. The problem can be alternatively expressed as "find the minimum value for c such that $Ls(c) \leq Ls_{max}$ and $1 \leq c \leq c_{max}$", where $Ls(c)$ is the expected queue length having c servers, calculated using one of the methods presented in the previous section, and c_{max} is the total number of available checkouts in the supermarket. Given the complexity of the model, different values for c are tried with the queue length forecast equations until the best value is found, starting with $c = c_{now}$, where c_{now} is the current number of open counters. Algorithm 1 shows a possible implementation of this process.

Alternatively, a maximum waiting time Ws_{max} can be used as threshold, instead of Ls_{max}. In

Algorithm 1: Determine the optimal checkouts configuration.

```
c ← c_now;
if Ls(c) > Ls_max then
    while c ≤ c_max and Ls(c) > Ls_max do
        c ← c + 1;
else
    while c ≥ 1 and Ls(c) < Ls_max do
        c ← c − 1;
    c ← c + 1;
```

that case the algorithm remains the same but with $Ls(c)$ and Ls_{max} respectively replaced by $Ws(c)$ and Ws_{max}.

4.4.2 Opening/closing fluctuations control

In order to avoid opening and closing checkouts too frequently, an additional control has to be implemented. For example, it should be possible to exceed the maximum waiting time limit for a short period of time, to avoid opening a new counter that would not be fully used after this short high-traffic period is ended. The system should therefore evaluate if it is necessary to change the checkouts configuration in a new state which is persistent for a relevant amount of time, or if the increased traffic is transient and would last for just a short period. If such trend is not persistent, there should be no changes in the actual number of open checkouts.

To do so, a multi-step forecast must be implemented. For example, let $\hat{c}_{t+1}$ be the forecasted optimal configuration for the next interval and c_t the current configuration: if $\hat{c}_{t+1} > c_t$, the values for $\hat{c}_{t+2}, \hat{c}_{t+3}, ..., \hat{c}_{t+n}$ must also be computed to make a decision. If the trend is not persistent, e.g. $\hat{c}_{t+2} = c_t$, there should be no changes in the configuration. We can thus define two store-based customizable parameter:

- n: the number of future steps that shall be analyzed, with $n \geq 2$;
- m: the minimum number of steps that should present the new trend level, with $m \leq n$.

If at least m of the n predicted values present a persistent trend, the checkouts configuration shall be adjusted accordingly. Otherwise, the trend variation can be considered temporary and therefore ignored.

5 Results

The first section of this chapter briefly discusses the implementation of the prototypes used for testing and for performance evaluation. The next sections present the results obtained by the forecasting techniques described in the previous chapter and justify the choices made for the final model.

5.1 Prototype implementation

Different prototypes for the models presented in Chapter 4 were implemented in Python to be able to evaluate the prediction accuracy. In order to simplify the whole evaluation process and to prevent a model's results to influence the others', each model was implemented and tested separately. When the final architecture of the complete predictive model was defined by choosing the prototypes that obtained the best performances, a final implementation, with all the submodules working sequentially, was written. This implementation was designed to be able to observe the customers' behavior and produce forecasts in real-time, and to be updated with newly collected observations once a day. With

this structure, the prototype can be easily ported to the Scala programming language and integrated into the RetailerIN platform without needing much refactoring.

5.2 Inflow rate forecast

This section discusses the results obtained by the forecasting methods for the inflow rate presented in Section 4.1, in particular:

- Persistence model
- Drift model
- Autoregressive Neural Network model
- SARIMA model

Given the greater amount of available data, the *dataset A* (see Section 3.1) was chosen for the models' evaluation. Each model was trained on the first half (from September to November) and tested on the second half (from December to February) of the dataset.

The models' prediction accuracy was evaluated by analyzing the *one-step forecast* errors on the test set. A *forecast error* is the difference between an observed value and its forecast, and it can be written as $e_t = y_t - \hat{y}_t$. Only the forecasts for the store's opening hours (from 6:00 to 23:00) were considered for the evaluation, since the values outside these periods are equal to zero and therefore not relevant. Different measures were used to summarize the forecast errors:

- *Mean Absolute Error*: $\text{MAE} = \text{mean}(|e_t|)$
- *Root Mean Square Error*: $\text{RMSE} = \sqrt{\text{mean}(e_t^2)}$
- *Mean Absolute Percentage Error*: $\text{MAPE} = \text{mean}(|100e_t/y_t|)$

Since the final implementation on the RetailerIN system will work in a real-time environment, the computational performances are another important aspect that must be taken in consideration when choosing the forecasting method. This metric is composed by two measurements: the *forecast time* and the *update time*, that respectively indicate the average time spent by the model computing the forecast for the next interval and updating the model with newly collected data. The forecasting time is the most significant of the two, since new predictions are computed with high frequency, given the intervals' short length, while the model updates are executed once every day, preferably when the store is closed. The time spent for the initial training, when the system is started for the first time, is not significant and therefore not considered.

Table 5.1 summarizes the results obtained by each model. The residuals analysis for the AR-NN model can be seen in Figure 5.1.

Table 5.1: Accuracy and performance results of the inflow rate predictive models.

Model name	MAE	RMSE	MAPE	Forecast time (ms)
Persistence	5.0355	7.1982	49.2727%	0.0001
Drift	2.6586	3.5376	15.1712%	0.0023
AR-NN	2.6027	3.4878	14.0093%	2.1529
SARIMA	3.1986	4.2620	19.8370%	3.3107

5.2.1 Conclusions

The best performances in terms of forecast accuracy were obtained by the AR-NN model, and the analysis of the residuals shows that there is no information left out from the approximation. However, the Drift model was chosen to be implemented in the final architecture, since it obtained very similar results and the calculation of a new forecast is faster. Moreover, the implementation of the Drift model is much simpler than a neural network, thus it can be implemented directly without needing any additional library.

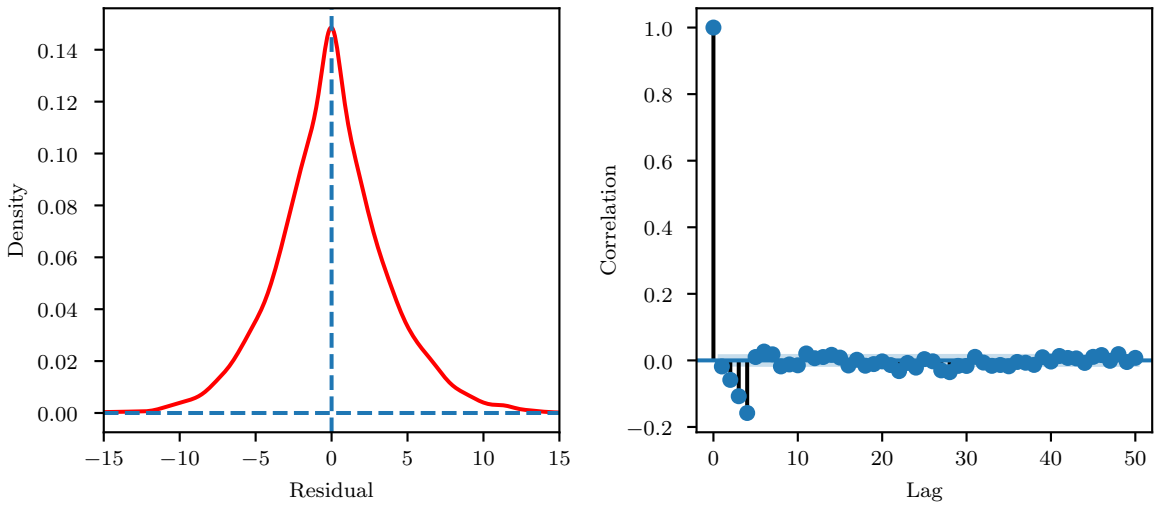

Figure 5.1: Residuals analysis of the AR-NN model.

5.3 Arrival rate forecast

In Section 4.2 a forecasting method for the arrival rate was described. To evaluate its accuracy, a measurement of the actual arrival rate was necessary. Unfortunately this information was not directly available in neither of the two datasets, so an approximation based on the queue behavior was established, in particular on the total number of customers in queue, available with one-minute frequency. When the total number of customers in queue increases, a new customer must have joined the queue. Therefore, the positive difference between the values of consecutive intervals should represent the number of new customers that have just arrived at the checkouts. The arrival rate λ_t can be thus written as:

$$\lambda_t = Lq_t^{TOT} - Lq_{t-1}^{TOT} + \theta_{t-1} \tag{5.1}$$

where Lq_t^{TOT} is the total number of customers in queue at time t and θ_{t-1} is the total number of customers served at time $t-1$. The obtained results were then aggregated in a 10-minute interval to obtain the same frequency as the forecast. With this approach, the *dataset A* could not be used for the accuracy evaluation since it does not contain data regarding the total number of customers in queue. Instead, the evaluation was conducted over the smaller *dataset B*.

Since this model is still part of the time series forecasting field, the same errors measures presented in the previous section were used. The accuracy for various multi-step forecasts is shown in Table 5.2. The number of steps indicates the number of future intervals forecasted, e.g. a three-step forecast gives prediction values for λ_{t+3} having the measured inflow rate data till interval t.

Table 5.2: Accuracy results of the arrival rate predictive model.

Steps	MAE	RMSE	MAPE
1	1.3220	1.8464	20.4156%
2	1.4769	1.9971	21.8374%
3	1.5559	2.1961	24.2532%
4	1.7508	2.4305	27.0171%

5.3.1 Conclusions

The results show that the model's prediction accuracy decreases as the number of steps ahead for which the forecast is computed increases. This is probably caused by the error introduced by the store's inflow rate forecast: for example, the three-step predictions are based on the one-step and two-step inflow rate's predictions, as described in Section 4.2. Therefore, improving the inflow rate forecasting model introduced previously would also increase the overall performances of this model.

Another important factor that must be taken in consideration when evaluating the obtained results is the period in which the data have been collected. As stated in Section 3.1, the *dataset B* used in

the evaluation covers a two-month time span, from the 6th of February to the 25th of March 2020, and includes data from a store in northern Italy. In that period the COVID-19 emergency greatly impacted the customers' shopping behavior and imposed various security measures that affected the supermarkets' traffic levels: at the beginning of the pandemic, a lot of people started panic buying groceries; then, when the lockdown began in March, many customers were at home and had more time to go shopping, thus the usual traffic trend changed, while grocery stores limited the number of clients that could be in the shop at the same time, affecting the inflow rates and the waiting queues. The results shown in this section, while still valid for the traffic levels of this situation, are therefore not representative of the usual performances that would be obtained by the model in production.

A further accuracy improvement would be given by including in the model the analysis of the customers' movements during the shopping sessions. This would require to process in real-time the position of every customer currently in the shop, with the aim of identifying some recurrent shopping behavior that could give a forecast of the actual time that each customer will spend inside the store. This forecast could take in consideration the walking speed, the entry time, the pauses, the path followed, and any other significant information that would allow to assign each customer to a specific group of people with similar shopping behavior, and therefore with similar dwell times. The dwell times such determined would probably be much more accurate, since they would not be based only on the day and hour of the shopping session as the approximation used in this research, and would therefore improve the arrival rate forecast. However, these types of advanced analyses would require high-frequency data about the asset's movements in each session and different forecasting techniques that are out of the scope of this research.

5.4 Queue length forecast

The results obtained by the arrival rate forecast presented in the last section were used in this section to calculate a forecast of the waiting queue length. In Section 4.3.2 three approximations of the average queue length based on the SBC approach were presented:

- A1
- A2
- MAR

Besides the arrival rate, these models also requires an approximation of the service rate in order to be used. With the approach presented in Section 4.3.1, a realistic time-independent service rate μ was calculated, with a value of:

$$\mu = 2.12 \frac{\text{customers}}{\text{interval}} = 0.21 \frac{\text{customers}}{\text{min}}$$

To evaluate the accuracy of each method, a measurement of every counter's queue length was needed. The RetailerIN system is capable of inferring these information from the customers' movement in the store. However, this inference is susceptible to accuracy issues, due to the nature of the problem, that must be taken in account when evaluating the forecasting performance. These issues includes:

- *Tracking of baskets*: when a customer is using a basket and his turn of being served arrives, the basket is left at the beginning of the rolling belt terminal. It is therefore impossible to determine when the customer has been served and left the queue.
- *Customers leaving the queue*: it is possible that a customer leaves the queue to get additional products that he forgot or to join a different shorter queue. This kind of behavior is difficult to correctly track and can lead to inaccurate measurements.
- *Sensors accuracy*: if the counters or the waiting queues are too close from each other, the system may have some difficulties distinguishing every separate queue.

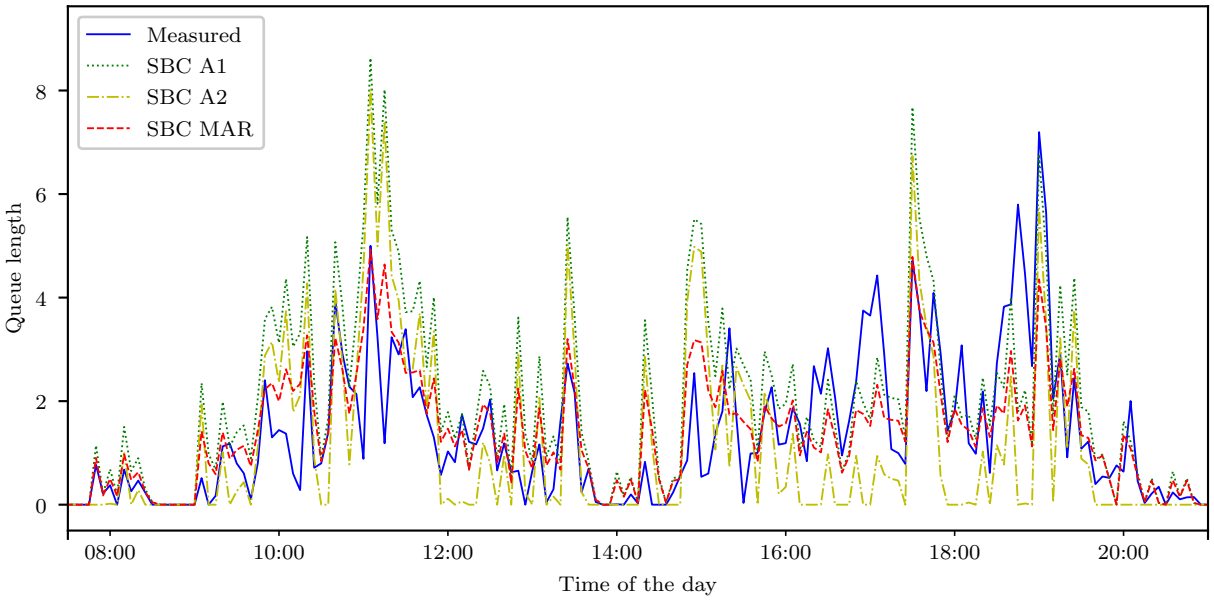

Figure 5.2: Example of the queue length prediction results with the A1, A2 and MAR methods.

As stated in the previous section, the queues length data was available only for the *supermarket B*, but it covered just one week of observations, from the 2th to the 8th of March, instead of two months as the sessions' dataset. Again, since the training data can be seen as a time series, the same accuracy measures from the previous section were used. The results are shown in Figure 5.2 and Table 5.3.

Table 5.3: Accuracy results of the queue length predictive model.

Approximation	MAE	RMSE	MAPE
A1	0.8539	1.6681	28.7874%
A2	0.8049	1.6088	26.5083%
MAR	0.4919	0.9646	19.5708%

5.4.1 Conclusions

The MAR approximation obtained the best forecast performance. As explained before, the accuracy of the model is greatly affected by the accuracy of the queue length measurements. Moreover, the samples used for the evaluation were limited to only one week of data and the prediction accuracy may vary significantly depending on the considered period. There is also the store's size factor that has a significant impact on the results: with a higher traffic level, the effects of random and unpredictable queueing behavior, as well as the errors in the measurements, are diminished. The store used for the evaluation is small-sized, therefore the traffic levels are quite low and the accuracy is thus affected.

6 Conclusions and future works

First, this chapter briefly describes the conclusions this work sought to address along with the research goals and the methods used to reach these goals. The next section discusses potential improvements and future researches based on the insights of this thesis.

6.1 Conclusions

The results presented in the previous chapter supports one of the initial hypotheses introduced in Section 1.1, that claims that it is possible to forecast the waiting queue length by analyzing the

customers' shopping behavior during the previous periods. By designing a predictive model for the customers' inflow rates and dwell times, it is possible to forecast the arrival rates at the checkouts, that give an idea of the traffic levels that the counters will have to handle in the near future. Once the arrival and service rates are established, it is possible to determine the evolution of the waiting queues, as well as the average time each customers will have to wait before being served, by applying queueing theory. The store's management can then take staff allocation decisions based directly on these forecasts, or can rely on the suggestions given by the system, that automatically determines the optimal staffing levels based on the future development of the customers' traffic. This thesis discusses a queue-length-based optimization, that minimizes the staff at the counters while maintaining the queue length under a maximum threshold, but the algorithm can be easily adapted to reflect custom internal store's queue management policies and staff scheduling constraints.

Considering the randomness of the customers' behavior, the forecasts obtained with this approach are accurate enough to represent a useful tool to support decision making and resource planning. Given that RetailerIN is focused on layout and staffing optimization to increase store's profitability, this model provide an additional value to the predictive capacity of the Thinkinside's system, and it is easily implementable, since it was designed to cooperate with the already-existing components and data formats. Moreover, the analyses conducted on this thesis are generic enough to be adapted to other contexts besides supermarkets, for example airports and banks, but also call-centers, as long as measurements for the traffic and service levels are available and the setting is characterized by at least one waiting queue with adjustable service capacity.

6.2 Future works

As stated previously, the main objectives of this thesis were to develop a predictive model for the evolution of the waiting queues in the near future and to support management in adjusting staffing levels in order to maximize store operations while maintaining an adequate service level. While the accuracy of the queue length forecast could be evaluated, as seen in Chapter 5, it was not possible to verify the effectiveness of the suggested checkouts optimizations due to time constraints. This could be done by implementing the final model into RetailerIN and testing it with different supermarkets layout. By ensuring that the suggestions provided are followed by the staff and by collecting the performance metrics over a time span of different months, it would be possible to compare the performance of the influenced and uninfluenced system by analyzing the average queues' length and the amount of time the staff spent at the counters. However, this is impractical for several reasons, the first of which is that it is impossible to guarantee that the checkouts configuration suggestions will always be followed, and therefore it is difficult to obtain performance metrics that are truly representative of the effectiveness of the model. A less significative but simpler evaluation method, without the aforementioned issues, would be to implement a simulation by using realistic arrival and service rates based on the approximations described in this thesis.

Another possible improvement would be to refine the dwell times distributions by excluding the time spent in queue from the dwell time of each session. Moreover, as already described in Section 5.3.1, the arrival rate forecast, and therefore the overall accuracy, could be further improved by implementing a more advanced dwell time prediction by considering in real time the movements of every customers in the store.

Bibliography

[1] Holt C. C. Forecasting seasonals and trends by exponentially weighted moving averages. *ONR Research*, 52(1):5–10, 1957.

[2] Little J. D. C. A proof for the queuing formula: L=λW. *Operations Research*, 9(3):383–387, 1961.

[3] Brown R. G. Exponential smoothing for predicting demand. *Operations Research*, 9(5):673–687, 1956.

[4] Kendall D. G. Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded markov chain. *The Annals of Mathematical Statistics*, 24(3):338–354, 1953.

[5] Tom G. and Lucey S. Waiting time delays and customer satisfaction in supermarkets. *Journal of Services Marketing*, 9(5):20–29, 1995.

[6] Aksu H. *Dwell time forecast and checkout optimisation in supermarkets.* PhD thesis, Charles Sturt University, 2018.

[7] Hyndman R. J. and Athanasopoulos G. *Forecasting: Principles and Practice.* OTexts, 2nd edition, 2018.

[8] Hyndman R. J. and Khandakar Y. Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software*, 26(3), 2008.

[9] Kleinrock L. *Queueing Systems Volume 1: Theory.* Wiley-Interscience, 3rd edition, 1975.

[10] De Livera A. M., Hyndman R. J., and Snyder R. D. Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association*, 106(496):1513–1527, 2011.

[11] Berman O. and Larson R. C. A queueing control model for retail services having back room operations and cross-trained workers. *Computers and Operations Research*, 31(2):201–222, 2004.

[12] Box G. E. P. and Jenkins G. M. *Time Series Analysis: Forecasting and Control.* Wiley, 1st edition, 1970.

[13] Box G. E. P. and Cox D. R. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 26(2):211–252, 1964.

[14] Jones P. and Peppiat E. Managing perceptions of waiting times in service queues. *International Journal of Service Industry Management*, 7(5):47–61, 1996.

[15] Stolletz R. Approximation of the non-stationary M(t)/M(t)/c(t)-queue using stationary queueing models. *European Journal of Operational Research*, 190(2):478–493, 2008.

[16] Green L. V. and Kolesar P. J. The pointwise stationary approximation for queues with nonstationary arrivals. *Management Science*, 37(1):84–97, 1991.

[17] Green L. V., Kolesar P. J., and Soares J. Improving the SIPP approach for staffing service systems that have cyclic demands. *Operations Research*, 49(4):549–564, 2001.

[18] Riel A. C. v. R., Semeijn J., Ribbink D., and Bomert P. Y. Waiting for service at the checkout: Negative emotional responses, store image and overall satisfaction. *Journal of Services Marketing*, 23(2):144–169, 2012.

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

[20] Tang Z. and Fishwick P. A. Feedforward neural nets as models for time series forecasting. *INFORMS Journal on Computing*, 5(4):374–385, 1993.