ERASMUS UNIVERSITY ROTTERDAM
Erasmus School of Economics

MSc. Econometrics and Management Science
Master Thesis in Business Analytics & Quantitative Marketing

Quantitative Analysis of Traffic Flows with Improved Seasonal Naive Forecasting

Author:
S.F.W. Pouwelse
Supervisor Erasmus University:
dr. M. Zhelonkin
Second Assessor:
dr. A. Alfons

July 8, 2022

Abstract

The Westerscheldetunnel is one of the largest traffic structures in the Benelux. Hence, it is important that road traffic can be precisely predicted. We estimate these traffic volumes by improving seasonal naive forecasting methods. First, a quantile regression model estimates base probabilistic forecasts with several predictive variables. These predictors are designed to capture the difference in external circumstances between the traffic intensity now and the intensity one seasonal cycle ago. Next, with this base model, we use robust peak detection with the median of absolute deviations. After analysing these peaks, we find new predictors to use for the probabilistic forecasts. Furthermore, we use a Long Short-Term Memory network and a Bayesian linear regression model to estimate point forecasts for the traffic volumes. The best model is 23% more accurate relative to simple seasonal naive forecasting.
Keywords: Traffic Volume, Seasonal Naive Forecasting, Improved Seasonal Naive Forecasting, Probabilistic Forecasting, Point Forecasting, Quantile Regression, Square Root Lasso, Robust Peak Detection, LSTM Network, Bayesian Regression

Contents

1 Introduction

After completion in 2003, the Westerscheldetunnel became the largest traffic tunnel in the Benelux with a length of 6.6 kilometers. Every day, around 21 000 vehicles drive through the tunnel[1]. This tunnel, situated in the Dutch province of Zeeland, connects the province to its region Zeeuws-Vlaanderen and to the neighbouring country Belgium. In this paper we investigate variants of seasonal naive forecasting to predict the traffic flows that occur in the Westerscheldetunnel.

To analyse these traffic flows, we use data provided by the government agency of the province of Zeeland, which is called *Province Zeeland*. This data contains the amount of vehicles that drive trough the tunnel per hour for the years 2017-2020. We primarily use the data of the years 2017 till 2019 to predict the traffic, as this is the most recent data that is not influenced by the covid-19 pandemic[2].

The main research question of this paper is:

How well can we predict the traffic flows in the Westerscheldetunnel by improving seasonal naive forecasting methods?

To answer our main research question, we first answer the following sub-questions:

What are external factors that increase the accuracy of the prediction of the traffic flows in the Westerscheldetunnel?

Which moments have a significant peak in the prediction error and what causes this?

To answer the first sub-question we perform a quantile regression with a square root variant of the Least Absolute Shrinkage and Selection Operator (Lasso) penalty. With this model, we find predictors for the traffic flow. We also decompose the seasonality into harmonic frequencies, to account for the cycling pattern. For the second sub-question, we compare the predictions of the model to the true observations. Then, we identify the peaks that significantly differ from their estimate using robust peak detection, following the Median of Absolute Deviations (MAD).

Knowing this, we try to find new external causes which might impact the traffic flow at these moments. Such that, we do not overlook any important predictors. Now that we have a good approximation of all influential external factors, we perform two different types of forecasting. First, we use the quantile regression to achieve probabilistic forecasting. Second, we perform point forecasting with a Long Short-Term Memory (LSTM) network and a Bayesian Linear Regression. So, with these models, we predict the traffic volumes and answer our main research

[1] N.V. Westerscheldetunnel. (n.d.). *Westerscheldetunnel*. Retreived from https://www.westerscheldetunnel.nl/nl/westerscheldetunnel/

[2] Centraal Bureau voor de Statistiek. (2022). *Mobiliteit in coronatijd*. Retreived from https://www.cbs.nl/nl-nl/visualisaties/welvaart-in-coronatijd/mobiliteit

question.

In Section 2, we explain more in depth why we choose our methods and models. In Section 3, we explain the data alterations we make and the predictors we use. Then, in Section 4, we elaborate on the methods to answer our research questions. The results of our research are displayed in Section 5. The conclusion and discussion of these results follow in Section 6.

2 Literature

For this paper, we use quantile regression, we perform a robust method for peak detection and we construct two models for the point forecasting. In this section we evaluate these methods.

The goal of this research is to predict the traffic flow as good as possible. The importance of the distinction between explaining and predicting in statistical research was put on the agenda by Shmueli (2010). He defined predictive modelling as the use of statistical methods to predict future observations. Meanwhile, he stated that explanatory modelling is used to test causal hypotheses. Hence, in this research, we do not talk about the causal relations between the traffic flows and the explanatory variables. We research and discuss the impact of the explanatory variables on the prediction performance of the model. Therefore, we can not make any causal conclusions based on our variable analyses. However, we can make conclusion based on the prediction accuracy of the model.

2.1 Previous Research

Analyzing and predicting road traffic is not a common subject for econometric research. Das & Tsapakis (2020) researched to predict traffic volumes by using several linear regression models and machine learning techniques. Sekuła et al. (2018) estimate the traffic volume by training neural networks to learn the relation between these volumes and several influential factors. These types of models are called volume profiles. However, we believe that these volume profiles could be improved.

To find methods for the proposed improvements, we first look at seasonal naive forecasting. Seasonal naive forecasting is a commonly used and is a simple technique for estimating time series. The core principle is to predict the dependent variable by using the value of the dependent variable of one seasonal cycle ago (Hyndman & Athanasopoulos, 2018). For further improvements, we look into the field of internet traffic. The predictive methods for internet traffic are also usable for road traffic. As both fields predict the same: the passage trough a certain point, whether this point is a website or a tunnel. Papagiannaki et al. (2004) use a baseline to predict standard behaviour for internet traffic. Then, they use anomaly detection to look for diverging

observations. The same holds for the research of Gu et al. (2005). They detect anomalies by comparing the network traffic against a baseline distribution.

Furthermore, we do not want to only use point forecasting. Therefore, we also use probabilistic forecasting. Dawid (1984) encouraged the world of statistics to use more probabilistic forecasting, where we predict the probability of an event instead of the event itself. For example, we predict the probability that a certain amount of vehicles passes through the tunnel, rather than simply predicting the number of expected vehicles (Gneiting, 2008; Gneiting & Katzfuss, 2014). Hence, our research can be used to predict traffic flows in two different ways. First, we can simply predict the traffic volume for a certain point in the future. Second, we can predict traffic volumes with an estimation of the probabilities of occurrence.

So by bringing all these fields together, we get a renewed insight in the behaviour of traffic data. We believe that this approach can yield new insights for both road traffic and for online traffic. Thus, we get an approach that is widely usable. For example, within the government agency *Province Zeeland*, it can be used to analyse traffic flows on all roads under supervision of the Province. Furthermore, it can be extended to estimate activities on several websites of the organisation.

2.2 Quantile Regression with Square Root Lasso

So to estimate the traffic volumes as well as possible, we obviously need to have adequate predicting variables. Therefore, we first construct a simple base model. Then, we can make predictions with this base model and investigate the outliers. Because we analyse outliers, we need statistical robust estimations. Hence, for this cause, we use the quantile regression with a variant of the Lasso error function.

2.2.1 Quantile Regression

The foundation for quantile regression lies around 1750 in the research of Ruer Bošković. He used the conditional median to perform a regression analysis (Koenker & Hallock, 2001). Then around 1886, Francis Edgeworth published several papers that translated Bošković's work into the plural median, which was a geometric regression approach set to compete with the Ordinary Least Squares (OLS) regression (Koenker, 2000). Later, this work would be made into the quantile regression model, as proposed by Koenker & Bassett Jr (1978).

Koenker & Bassett Jr discussed several problems with the robustness of OLS-estimation. They introduced quantile regression as a robust alternative for the linear model. This is done by modeling the quantiles of the dependent variable, without assuming a specific conditional

distribution (Waldmann, 2018). Later Koenker & Hallock (2001) extent Koenker & Bassett Jr's research with conditional quantile functions. These functions allow for modelling the conditional distribution of the dependent variable as a linear expression of the explanatory variables. This is also seen in Gneiting et al. (2006), where the true observations are displayed against the quantiles of the estimation distribution. Which should yield a uniform distribution if the model is robust (Gneiting et al., 2006).

In terms of seasonality, Fourier (1822) introduced a method to describe a harmonic series into a set of harmonic functions. A variant of this method was also used in the research by Belloni, Chernozhukov, & Fernández-Val (2011), where several sinusoids were used as explanatory variables in a quantile regression.

As our data contains outliers and several cycling and seasonal patterns (daily, weekly, seasons), we need a regression model that is robust in these situations. Furthermore, our goal for this model is to make probabilistic forecasts and to detect anomalies. Hence, these predictions need to be robust. Otherwise, we are not be able to get a good understanding of the outliers and the factors that might cause them. Therefore, we choose the quantile regression model with harmonic frequencies to analyse the characteristics of the traffic flows.

2.2.2 Lasso Error Term

The first variant of the Lasso error term was used in the paper by Santosa & Symes (1986), to account for the white noise in seismograms. Later, Breiman (1995) derived a similar method to produce lower prediction errors for real and simulated data sets, which he called the nonnegative garrote. This method can also set parameters to zero, if the predicting variable does not improve the prediction quality substantially. This was not possible for the Ridge regression model. However, Breiman did state that Ridge regression was more stable compared to the nonnegative garrote.

The models and ideas of these two papers where then combined by Tibshirani (1996). He proposed the Lasso error term for linear estimation, as it provided better forecasting accuracy and simplified the interpretation by setting unnecessary coefficients to zero. Furthermore, in contrast to the nonnegative garrote, Tibshirani stated that Lasso showed the stability of a Ridge regression.

However, as years went by, doubts were raised about the ability of Lasso to specify the correct sparsity pattern. The sparsity pattern is the subset of variables for which the parameter is nonzero (Hastie et al., 2015). Hence, these are the variables that are deemed to improve the prediction quality of the model. Meinshausen & Yu (2009) acknowledge these doubts, but also

state that Lasso can still provide a good approximation of the sparsity pattern. Furthermore, they state that Lasso is still a useful tool for model identification when dealing with high-dimensional data. Lasso is also useful when the data is highly correlated, as long as λ is chosen to match the design of the data (Dalalyan et al., 2017; Hebiri & Lederer, 2012). Hence, this allows the inclusion of lagged dependent variables in the model. Belloni, Chernozhukov, & Wang (2011) constructed a quadratic error term called Square Root Lasso, which outperforms the regular Lasso and the Cross-Validation Lasso methods.

Hence, we use the square root variant of the Lasso error term for our base model, as it has the ability to rule out predicting variables that are not important for the predictive performance. Furthermore, the method can approximate the correct sparsity pattern when dealing with high-dimensional and highly correlated data sets.

2.3 Peak Analysis

In this research, we use robust peak detection to find anomalies in our prediction errors. This method is based on statistical dispersion. Statistical dispersion is the extent to which the data values differ from each other (Siegel, 2016). Rousseeuw (1991) stated that based on the dispersion of a univariate variable, it is possible to calculate z-scores. These z-scores are a test statistic to detect if an observation is an outlier. Later, Rousseeuw & Hubert (2011) reviewed several new outlier-detection methods, instead of using the classical standard deviation. This is because they state that a single outlier can make the standard deviation arbitrarily large. They measured this concept in terms of the breakdown value. The breakdown value indicates the smallest fraction of observations necessary to break the estimator due to being arbitrarily large (Hubert & Debruyne, 2009). As a single outlier can break the standard deviation estimator, they give it a breakdown value of 0%.

Rousseeuw & Hubert suggest the MAD as the main test statistic for univariate data, which has a breakdown value of 50%. As our prediction error is a univariate variable, we use this statistically robust method to look for anomalies.

2.4 Prediction Analysis

After finding predictors and detecting anomalies, we perform point forecasts. For the prediction analysis we use two widely different methods. First, we use a LSTM network. Second, we use Bayesian linear regression.

2.4.1 LSTM Network

Our first method is based on machine learning. The LSTM network origins from Hopfield Networks, which were proposed by Hopfield (1982). Later, Rumelhart et al. (1986) added recurrence to the Hopfield Network to construct a Recurrent Neural Network (RNN). A special type of RNN is a LSTM network. They were first proposed by Hochreiter & Schmidhuber (1997). It was developed as a computational lighter and faster version of a RNN, which could sometimes outperform the RNN itself (Hochreiter & Schmidhuber, 1997).

Gers et al. (2002) proposed a variant of the LSTM network especially developed for estimating time series (Gers et al., 2002; Heaton, 2017). According to Yu et al. (2019), LSTM networks have achieved all the interesting results that were previously achieved by RNN's. They also state that LSTM networks are better in modeling long-term dependencies than a RNN. However, Zhang & Qi (2005) state that a regular neural network can not handle seasonality within the data set. Meanwhile, Gao et al. (2020) see a RNN as a great method to model such long-term dependencies. Therefore, it will be interesting to see how the LSTM network deals with the seasonality.

Lastly, to prevent overfitting, Gal & Ghahramani (2016b) proposed the variational dropout technique. Dropout is the probability that a node and its connections are removed from a training computation. With variational dropout, we drop the same node-mask at each time step. Meanwhile, regular dropout discards random node-masks at each dropout. Furthermore, Gal & Ghahramani (2016b) also used weight decay. Weight decay adds a Ridge penalty term to the weights of the network to help prevent overfitting (Hastie et al., 2009).

The computational advantages make the LSTM network very useful in our research. In addition, all neural networks can model the dependent variable as a non-linear function of the variables (Hastie et al., 2009). This differentiates this method from our quantile regression and our Bayesian analysis, where we use linear combinations of variables to estimate the dependent variable. However, we can not clearly derive the computed weights and output signals of the predictive variables. As output, we only get the value of the predicted dependent variable.

2.4.2 Bayesian Linear Regression

Our second method for the prediction analysis is a Bayesian linear regression. Bayesian analysis was named after Thomas Bayes, as he was the first to suggest methods to limit the probabilities of uncertain events (Bayes, 1763). Many decades later, his work would be made into mathematical methods by Laplace (1829). However, it would take till around 1980 - when computational power increased - for the Bayesian methods to become widely implemented (Fienberg, 2006).

This was due to the fact that for a Bayesian analysis, it is necessary to simulate several 'realities'. This is done through generating sequences by sampling from statistical distributions (Greenberg, 2012). Geman & Geman (1984) proposed the Gibbs-sampler to be used for this. The Gibbs-sampler is a Markov Chain Monte Carlo (MCMC) algorithm to approximate the exact posterior distribution of the parameters (Greenberg, 2012). Furthermore, Fernández & Steel (1998) state that properties like skewness and kurtosis have no impact on the posterior distribution.

A benefit of Bayesian analysis is that it is exact for small sample sizes (Greenberg, 2012). Hence, Bayesian inference is a proper method to analyse our data, which is a relatively small data set. As all dates only occur thrice till 2019, we get a total of 26 544 observations for the first prediction analysis. Furthermore, this method is interesting for our research as it provides more predictive insights. This is because the Bayesian analysis also gives us information about the weights and impact of the predictors on the estimation of the dependent variable. This contrasts with the LSTM network, which only gives the estimated value.

2.4.3 Mean Absolute Scaled Error

Commonly, the prediction accuracy of point forecasts are measured with the Mean Absolute Percentage Error (MAPE) (Kim & Kim, 2016). However, Hyndman & Koehler (2006) were not satisfied with the MAPE and proposed the Mean Absolute Scaled Error (MASE) due to several reasons. First, the MAPE becomes undefined when the true observation is zero. Furthermore, Hyndman & Koehler state that the MAPE puts more weight on observations with a low real value. This is due to the fact that a small error has a relatively larger effect on a small real value than it has on a large real value. In addition, Makridakis & Hibon (1979) explain that the MAPE is skewed as it has no upper bound for the errors. So the MAPE can go far above hundred percent. However, it can never go below zero percent. Later, Hyndman & Athanasopoulos (2018) extended the use of the MASE by creating a MASE suitable for seasonal time series.

Therefore, in this research, we choose the seasonal MASE over the commonly used MAPE. For example, our data has relatively low values of traffic at night (Figure 3), which sometimes even equal zero. Furthermore, the data contains multiple large peaks. Hence, all these factor would result in a skewed MAPE. Thus, we measure the prediction accuracy of our models with the MASE suitable for seasonal time series.

3 Data and Assumptions

The data contains all passing traffic in the Westerscheldetunnel per hour. These hourly observations are split into the two driving directions and per three vehicle length classes: 0-560 cm, 561-1220 cm and 1221+ cm. This research focuses on the total amount of traffic passing through the Westerscheldetunnel in both directions per hour. This value is denoted by *TOT*. The data ranges from 2017/01/01 till 2020/12/31. However, as the data of 2020 is heavily affected by the covid-19 outbreak, we first use the data from 2017 till 2019. Later, we include the data of 2020. In Figure 1 we see the traffic flow for all hours in January and February 2019, which gives 1 416 observations:

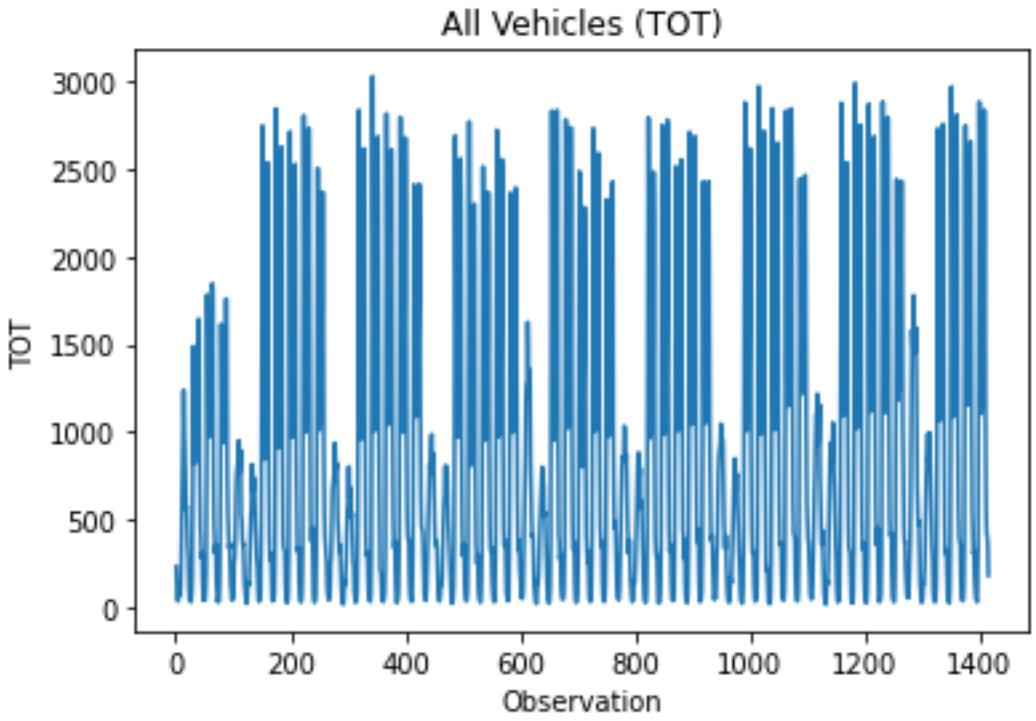

Figure 1: Hourly amount of vehicle passing from both directions through the Westerscheldetunnel in January and February 2019

3.1 Covid-19

In February 2020, the first Dutch people got covid-19. In March 2020, the Netherlands went into their first intelligent lockdown[3]. As we know today, the rest of 2020 was heavily impacted by the virus. The pandemic also had a large impact on the traffic activities. In Figure 2 we see the Dutch traffic activity in 2020 compared to traffic activity in 2019[2].

[3]Rijksoverheid. (n.d.). *Coronavirus Tijdlijn.* Retrieved from https://www.rijksoverheid.nl/onderwerpen/coronavirus-tijdlijn

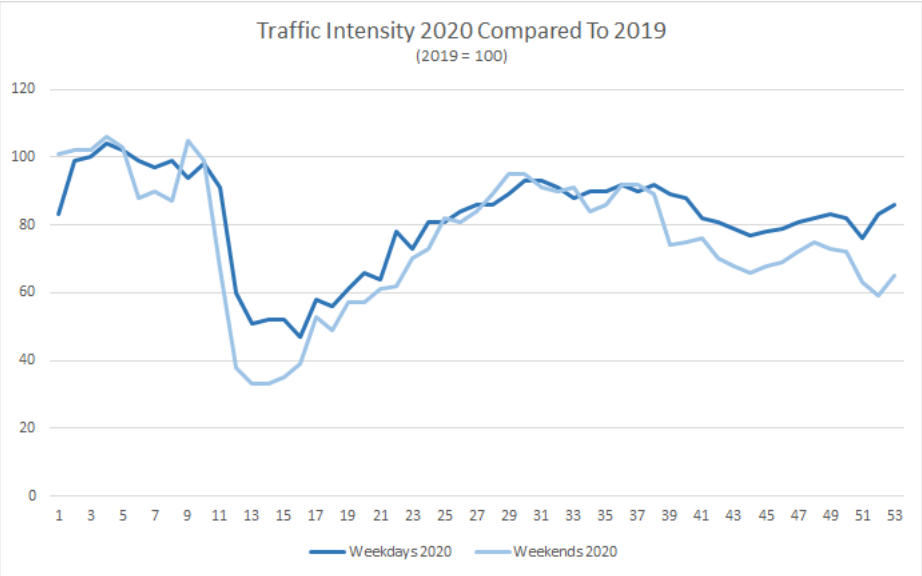

Figure 2: Traffic intensity of the Netherlands in 2020 in comparison to 2019

We can clearly see that the traffic activities in 2020 are negatively affected by the covid-19 virus. Therefore, we only use the data before 2020 in the first part of our research. Later, we use the full data set to predict the traffic flows in the Westerscheldetunnel by including covid-19 related variables.

3.2 External Predictors

For our research, we need to find external factors that help predict the traffic activity in the Westerscheldetunnel. To do this, we incorporate possible influential predictors in our model. We use two methods to find these variables. First, we simply discuss such factors with the traffic experts at *Province Zeeland*. For example, the weather data per hour and day-specific variables like holidays and toll free days (Appendix A.1). Our second method to discover influential predictors is based on our peak analysis. For the large peaks in our prediction error, we try to explain it with new impactful predictors.

For most of these predictive variables, we need to make an important alteration. As seen in Figure 1, every hour behaves similar throughout the weeks. As we include the seasonal lagged dependent variable into our model, it is very important that we incorporate these external predictors in terms of differences. For example, we do not simply include the hourly amount of rain. We include the difference in the amount of rain with the observation of 168 hours (one seasonal cycle) ago. Otherwise, we would have a form of serial correlation for our parameters (Cameron & Trivedi, 2005). This is due to the fact that the lagged observation is dependent on the amount of rain then. Furthermore, the current observation would be calculated with the lagged variable and the current amount of rain. Hence, we would indirectly use the amount of

rain multiple times in the estimation.

In the last part of our research, we perform probabilistic and point forecasts for the years 2017 till 2020. We include the same variables as for the regular analysis and add several covid-19 indicators suggested by the experts. For example, the difference in the number of daily covid-19 infections, the number of daily hospitalizations due to covid-19 and a variable stating if there is a lockdown (Appendix A.4).

3.3 Trend and Seasonality

In Figure 3 we can clearly see an additive seasonal pattern consisting of one week (168 observation). Furthermore, we also see small differences between weekdays themselves and between the two weekend-days. For example, based on Figure 1, the peaks on Friday seem consistently lower than those on Tuesday. Hence, our data set contains a seasonal pattern of one full week. It is important for our quantile regression that this seasonality is preserved throughout the whole data set (as explained later in Section 4.1). Therefore, we complete the weeks at the beginning and at the end of our data set.

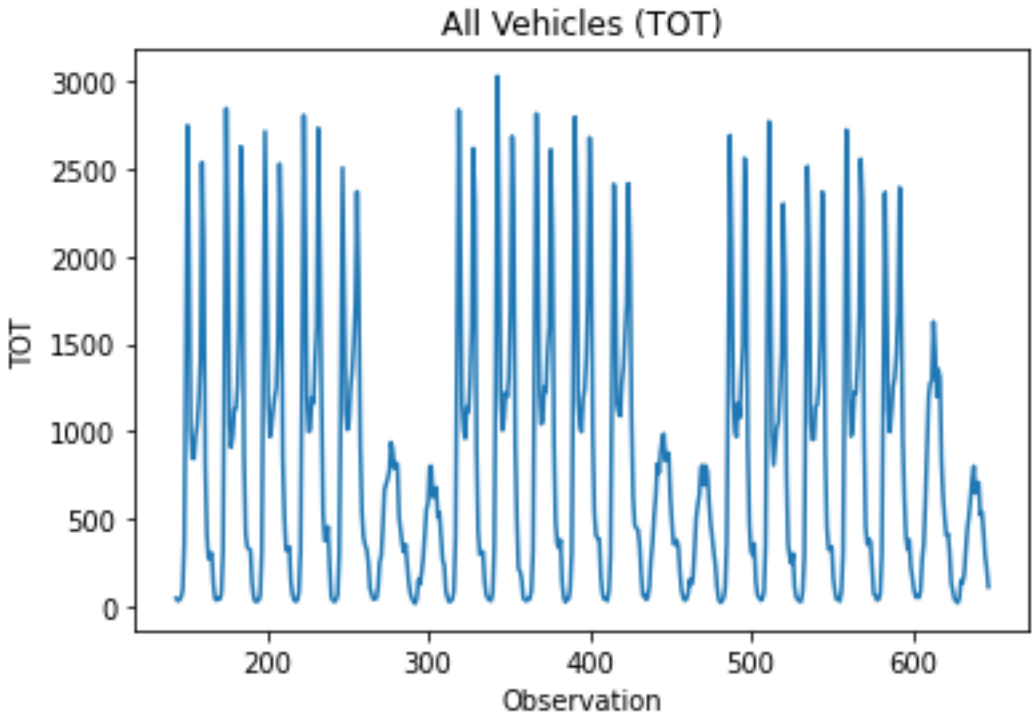

Figure 3: Hourly amount of vehicle passing from both directions through the Westerscheldetunnel from Monday January 7^{th} till Sunday January 27^{th} 2019

After removing the seasonality from the time series, we look for a possible trend. In Figure 4 we see a very fluctuating trend for the data from 2017 till 2019. Due to the lack of a clear slope, we state that there is no trend present for TOT in the years 2017 till 2019. As the changes in

trend can easily be caused by external factors like weather and holidays.

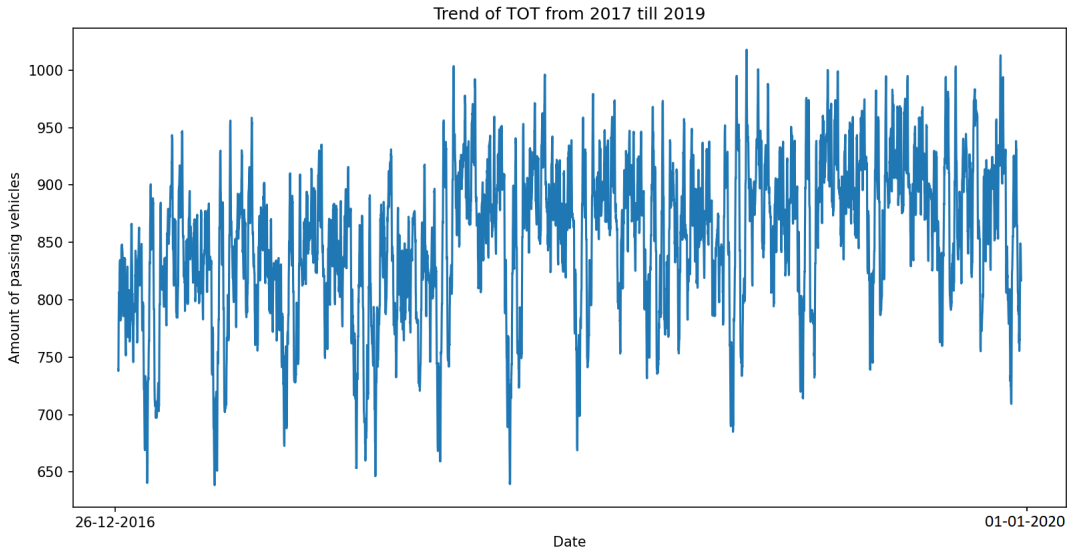

Figure 4: Trend after removing the seasonality from Monday December 26^{th} *2016 till Sunday January* 1^{th} *2020*

4 Methodology

In this section we explain our methods and models more in depth. First, we explain our quantile regression. Then, we elaborate on our robust peak detection method. At last, we tell more about the LSTM model and the Bayesian linear analysis. All models are constructed based on a random training set, consisting of seventy percent of all observation. The remaining thirty percent is used to measure the prediction accuracy of the models in terms of the MASE.

4.1 Quantile Regression with Square Root Lasso

With quantile regression we can estimate the conditional quantile functions (Koenker & Hallock, 2001):

$$\min_{\beta \in R^p} \sum \rho_\tau (TOT_i - \xi(x_i, \beta)), \tag{1}$$

where $\xi(x_i, \beta)$ is a linear function of parameters and $\rho_\tau(\cdot)$ is a tilted absolute value function that yields the τ^{th} quantile of $[TOT_i - \xi(x_i, \beta)]$ as the solution. This allows the model to learn a specific quantile, instead of the mean. In this research, we use $\tau = 0.05, 0.1, 0.15, ..., 0.9, 0.95$ for the probability forecasts.

Belloni, Chernozhukov, & Fernández-Val (2011) add sinusoidal functions to the minimization problem, such that we approximate a Fourier decomposition. Hence, we get:

$$\min_{\beta \in R^p} \sum \rho_\tau [TOT_i - \{g(t_i) + \xi(x_i, \beta)\}], \tag{2}$$

where $g(t_i) = \alpha_0 + \alpha_1 t_i + \alpha_2 sin(2\pi t_i) + \alpha_3 cos(2\pi t_i) + \alpha_2 sin(4\pi t_i) + \alpha_3 cos(4\pi t_i)$ for timestamp t_i. However, $g(t_i)$ can be arbitrarily long. Therefore, we use the following function for the harmonic frequencies (Zhu & Woodcock, 2014; Zhou et al., 2022):

$$g(t_i) = \sum_{s=1}^{S} \left(\alpha_{1s} cos\left(\frac{2\pi s t_i}{T}\right) + \alpha_{2s} sin\left(\frac{2\pi s t_i}{T}\right) \right). \tag{3}$$

Here, S is the number of seasonal patterns and T is the amount of observations per seasonal period. The variable t_i represents the place of the hour within the week. Therefore, we have $T = 168$ and $t \in \{1, ..., T\}$. Furthermore, $S = 158$ as this is amount of weeks that are included in our data set for 2017 till 2019. When we include 2020 in our data set, we get $S = 210$. Hence, it is important to complete the weeks at the beginning and at the end of our data set, as mentioned in Section 3.3. Otherwise, S would not be a whole integer.

After constructing the main part of the minimization problem, we add the square root Lasso error function. Therefore, we the Lasso error function and lambda as provided by Belloni, Chernozhukov, & Wang (2011):

$$\min_{\beta \in R^p} \left(\frac{1}{N} \sum_{i=1}^{N} \rho_\tau [TOT_i - \{g(t_i) + \xi(x_i, \beta)\}] \right)^{1/2} + \lambda ||\beta||_1, \tag{4}$$

where $||\beta||_1 = \sum_{j=1}^{p} |\beta_j|$, with p being equal to the number of parameters. A parameter is dropped from the sparsity pattern when $\beta < 0.0001$. Furthermore, the penalty level λ equals:

$$\lambda = cN^{1/2} \Phi^{-1}(1 - \alpha/2p), \tag{5}$$

for some constant $c > 0$ and probability level $0 < \alpha < 1$. The function $\Phi^{-1}(\cdot)$ corresponds to the inverse normal distribution. We choose $c = 1.1$ and $\alpha = 0.05$ as recommended by Belloni, Chernozhukov, & Wang (2011).

4.2 Peak Detection

To detect peaks, we compare the prediction errors for the quantile regression with $\tau = 0.5$. Whenever a prediction error is significantly large, we mark it as an outlier. For this research we

use the MAD as given by Rousseeuw & Hubert (2011):

$$MAD = c \text{ median}(|e_i - \tilde{e}|), \tag{6}$$

where $\tilde{e}$ is the median of all prediction errors e. Furthermore, c is a chosen constant, such that the MAD is consistent for the standard deviation of the distribution (Rousseeuw & Croux, 1993). This is achieved when this constant corresponds with the area of data between the 25^{th} and the 75^{th} quantile. As the distribution should be symmetrical, we only use the value that yields the 75^{th} quantile. As the value for the 25^{th} can easily be derived from this (Rousseeuw & Croux, 1993). For example, for the normal distribution we would have $c = 1.483$, due to:

$$c = \frac{1}{\Phi^{-1}(0.75)} \approx 1.483. \tag{7}$$

However, our prediction error does not follow a normal distribition, as the Jarque-Bera p-value is practically 0 due to a very high kurtosis (Heij et al., 2004). We do see that the error term moderately follows a symmetric distribution, as the skewness is -0.99 (Hair et al., 2022). So even though the variable is slightly skewed, we believe that their is enough symmetry to perform the outlier analysis. As our goals is only to get an indication of the days that are affected the most by outliers.

Furthermore, the mean is 2.82 for the prediction error. Then, based on a t-test, we see that this mean is significantly different from 0 (Heij et al., 2004). Therefore, we account for this by subtracting this mean from all error terms. Next, we scale the variance to 1, such that we standardize the variance. Then, we get a symmetric distribution centered around zero. Thus, now we can derive the value for the 75^{th} quantile and compute the constant by dividing one by this quantile value. The steps in this process are applied to all models.

Next, to compute the z-scores, Rousseeuw & Hubert propose to use the MAD to derive them:

$$z_i = \frac{|v_i - \tilde{v}|}{MAD}, \tag{8}$$

where v is the altered error term as described before. Furthermore, we use a critical value of 2.5 for the z-score (Rousseeuw & Hubert, 2011). So when $z_i \geq 2.5$, we mark observation i as an outlier.

Next, we take all outliers and sort them per day. Such that, we get an overview of the number of outlying hours per day. Hence, we get a global indication of how much a day is affected by outliers. Then, we use the peaks over threshold method to identify the most affected dates. The height of a threshold is a subjective decision. We choose a threshold based on the 70^{th} percentile

(Leadbetter, 1991; McNeil & Saladin, 1997). This yields the 90 most affected days, which we use to identify missing predictive variables. For all missing predictors, we constructed dummy variables.

With these new predictors, we once again perform our quantile regressions with square root Lasso to obtain new probability forecasts. However, if these new models do not show clear signs of improvement, we tune lambda to get the sparsity pattern that yields the lowest MASE. This will be done with the quantile regression for the 50^{th} percentile. Then, this found value for lambda is also used for all quantile regression based on other percentiles.

4.3 Point Predictions

The next step in our research is to predict the traffic volumes in the Westerscheldetunnel as good as possible. For the LSTM Network, we use all the predictors. This includes the variables added after the peak analysis. For the Bayesian regression we can not use all predictors, as this yields uninvertable matrices due to high singularity. Hence, for the Bayesian linear regression, we use the sparsity pattern as selected by the quantile regression with square root Lasso based on the 50^{th} percentile. After performing the point forecasts for 2017 till 2019, we predict the traffic flows till 2020 with the inclusion of several covid-19 related variables.

4.3.1 MASE

To compare our different models, we use the MASE for seasonal time series as proposed by Hyndman & Athanasopoulos (2018) to measure their accuracy. The MASE is always computed based on the out of sample test set.

$$MASE = mean(|q_t|) = mean\left(\left|\frac{TOT_t - \widehat{TOT}_t}{\frac{1}{T-m}\sum_{t=m+1}^{T}|TOT_t - TOT_{t-m}|}\right|\right), \tag{9}$$

where $TOT_t - \widehat{TOT_t}$ is the forecast error and m is the length of the seasonal period. Hence, we get that $m = 168$. Furthermore, the forecast error is scaled on the denominator, which is equal to the Mean Absolute Error (MAE) of the seasonal naive forecasts (Hyndman & Athanasopoulos, 2018). Thus, we get that the MASE < 1 when our model outperforms simple seasonal naive forecasting.

4.3.2 LSTM Network

This research uses the LSTM network as proposed by Gers et al. (2002). For the LSTM network, every computation exists of a Multilayer Perceptron (MLP). A MLP is a series of nodes connected

by weights and output signals (Gardner & Dorling, 1998). It consists of at least three layers: an input layer, one or more hidden layers and an output layer. This is displayed in Figure 5.

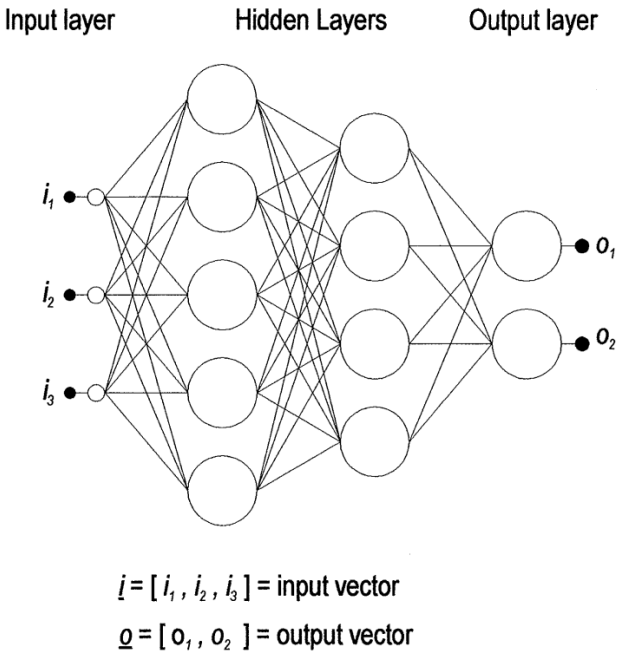

Figure 5: A MLP with two hidden layers from Gardner & Dorling (1998)

In the above figure we see that the input of this MLP consists of three values. For example, one lagged variable and two explanatory variables. Next, it passes through two hidden layers yielding two output values. This paper uses only one hidden layer with a size of $1 + \frac{2}{3}k$, where k is the number of parameters (Gers et al., 2002; Heaton, 2017). Furthermore, in contrast to the example in Figure 5, we only have one output value TOT_t.

By assigning different weights and output signals to nodes, the model trains itself in estimating more accurately. The error term that is used to find optimal estimations is the Normalized Root Mean Squared Error (NRMSE) (Gers et al., 2002):

$$NRMSE = \frac{\sqrt{(y_k - t_k)^2}}{\sqrt{(max(t_k) - min(t_k))^2}}, \tag{10}$$

where y_k is the output of the LSTM network and t_k is the target output. To improve the quality of our model, we standardize all inputs with a mean of zero and a standard deviation of one (Hastie et al., 2009). Thus, we standardize a variable x by subtracting its mean (μ) and dividing by its standard deviation (σ) (Heij et al., 2004):

$$\frac{x - \mu}{\sigma} \sim N(0, 1). \tag{11}$$

In our research we use the variational dropout method and weight decay method as given by

Gal & Ghahramani (2016a). We use recurrent dropouts rate of 0, 0.05, 0.1, 0.15, 0.2, 0.25 and 0.3 (Gal & Ghahramani, 2016b; Semeniuta et al., 2016; Srivastava et al., 2014). For the batch size, we use the values of 32, 64 and 128 (Kandel & Castelli, 2020; Goodfellow et al., 2016). We first look for the dropout rate that yield the lowest MASE. Therefore, we use a default batch size of 64. Next, we look for the batch size that gives the lowest MASE, with the best found dropout rate. When training a LSTM network, we stop the network when the loss of the validation set has not decreased for ten epochs (Dodge et al., 2020). As neural networks are a stochastic process, output can vary even when specifying a seed. Hence, we construct 50 LSTM networks for each parameter-tuning step and report the mean.

4.3.3 Bayesian Linear Regression

For the second variable analysis, we use a Bayesian linear regression as given by Greenberg (2012). We use a Markov Chain Monte Carlo (MCMC) algorithm to construct a linear regression with normal distributed error terms. Hence, our dependent variable TOT_t follows the multivariate normal distribution:

$$TOT_t \sim N_n(X_t\beta_t, \sigma_t^2 I_K), \tag{12}$$

where TOT_t is a size n vector of all observations and X_t is the set of the selected variables. Next, β_t is the set of corresponding parameters. We define n as the amount of observations and K as the number of parameters of the sparsity pattern. Hence, we get $n = 26\,544$ for the period till 2019 and $n = 35\,280$ for the period including 2020. Then, we use the following non-informative prior specification:

$$\beta|\sigma \sim N_K(b, \sigma^2 B) \text{ and } p(\sigma^2) \propto \sigma^{-2}, \tag{13}$$

with $b = 0$ and $B = I_K/\sigma^2$. Therefore, we have the prior:

$$\beta|\sigma \sim N_K(0, I_K). \tag{14}$$

Given the prior, we use the Gibbs sampler. We draw β from the multivariate normal distribution and σ^2 from the inverse gamma distribution:

$$\beta^{(g)} \sim N_K(\hat{\beta}^{(g)}, B^{(g)}) \tag{15}$$

$$\sigma^{2^{(g)}} \sim IG(n/2, \delta^{(g)}/2), \tag{16}$$

where $\hat{\beta}^{(0)} = b$, $B^{(0)} = B$ and

$$B^{(g)} = \sigma^{2^{(g-1)}}(X'X + B^{-1})^{-1} \tag{17}$$
$$= \sigma^{2^{(g-1)}}(X'X)^{-1}$$
$$\hat{\beta}^{(g)} = B^{(g)}(\sigma^{-2^{(g-1)}}X'y + B_0^{-1}\beta_0) \tag{18}$$
$$= (X'X + \sigma^{2^{(g)}})^{-1}(X'y)$$
$$\delta^{(g)} = (y - X\beta^{(g)})'(y - X\beta^{(g)}). \tag{19}$$

We do this until $g = 500\,000$. After all simulations, we remove a 10% burn-in sample and we use a thin-value of 10 to create the preferred sample (Greenberg, 2012). Then, we use the medians of parameters β and σ^2 to estimate $\widehat{TOT}_t$.

4.4 Comparison Lagged Differences

To further evaluate our point forecasts, we also construct a model where we include several lagged variants of the dependent variable. This to account for possible missing information or forms of serial correlation (Heij et al., 2004). Then, we can see if the addition of these predictors increases the accuracy of the point forecasts. For this, we choose the model that has the lowest MASE. To construct the variables in a similar method as our other external predictors, we use the lagged difference of the traffic volumes of 1, 2, 3, 4 and 5 hour(s) ago compared respectively to the traffic intensities of 169, 170, 171, 172 and 173 hours ago. Hence, we get that the variable TOT_j is equal to:

$$TOT\_j = TOT_{t-j} - TOT_{t-T-j}, \tag{20}$$

where $T = 168$ and $j = 1, 2, 3, 4, 5$. In this way we construct an indication if the general trend for traffic flow is lower compared to the traffic volumes one seasonal cycle ago.

5 Results

In this section, we state the results of our models and predictions. Further, more in-depth, conclusions are drawn in Section 6.

5.1 First Quantile Regression

An overview of the predicted quantiles from the quantile regressions is displayed in Figure 6. Here, we see the distribution of the true observations in the test set across the predicted quantiles. For example, it should hold that the 90 percent of these true observation should have a

value lower than the prediction of the quantile regression based on the 90^{th} percentile. Thus, when we use twenty quantiles, every quantile should approximately contain five percent of the observations.

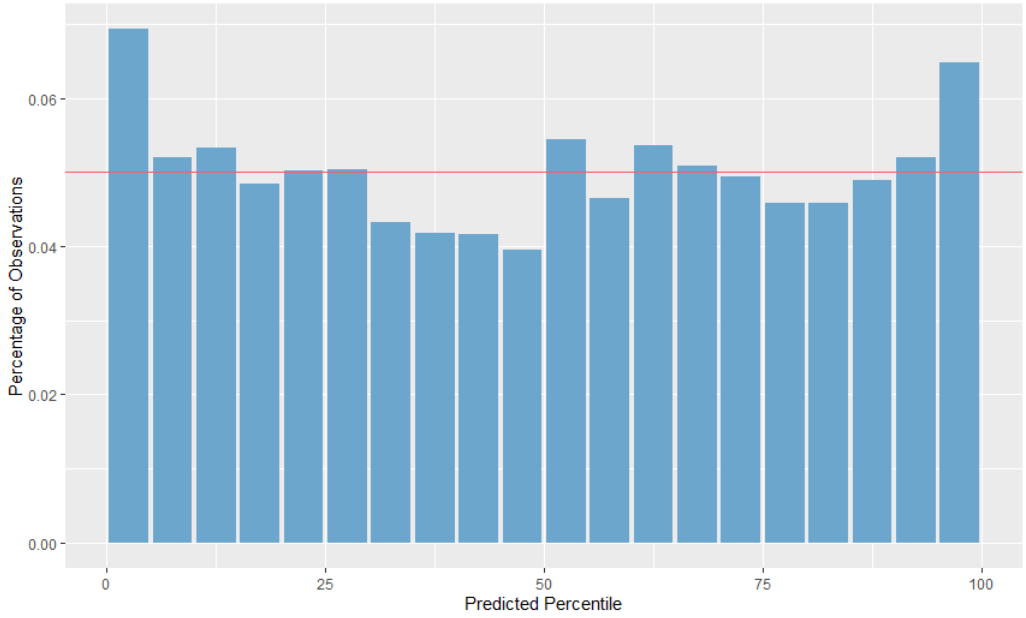

Figure 6: An overview of the distribution of the true observations of the test set across the predicted quantiles for the first quantile regressions

However, we see that we overpredict below the 5^{th} percentile and above the 95^{th} percentile. Which indicates that the outliers are not modelled correctly. Furthermore, we underpredict in the quantiles between the 30^{th} and 50^{th} percentile.

To analyse the outliers, we use the quantile regression based on the 50^{th} percentile. This model selects six variables for the sparsity pattern: *Wind_Force, Temperature, Air_Pressure, Holiday_Dutch, Cos_158* and *TOT_158*. These variables are explained in Table 10 (Appendix A.1). The coefficients of these variables are listed in Table 16 (Appendix B). Based on this model, we perform our peak analysis. This specific model has a MASE of 0.971.

5.2 Peak Detection

Based on the quantile regression for the 50^{th} percentile, we detect 1 000 outlying hours in the whole data set. This is equal to 3.8% of all observations. These anomalies are distributed across 275 days, which make up 24.9% of all days. In Table 19 (Appendix C), we have the 90 days that contain the most outliers. All these days are provided with an explanation on why they might diverge.

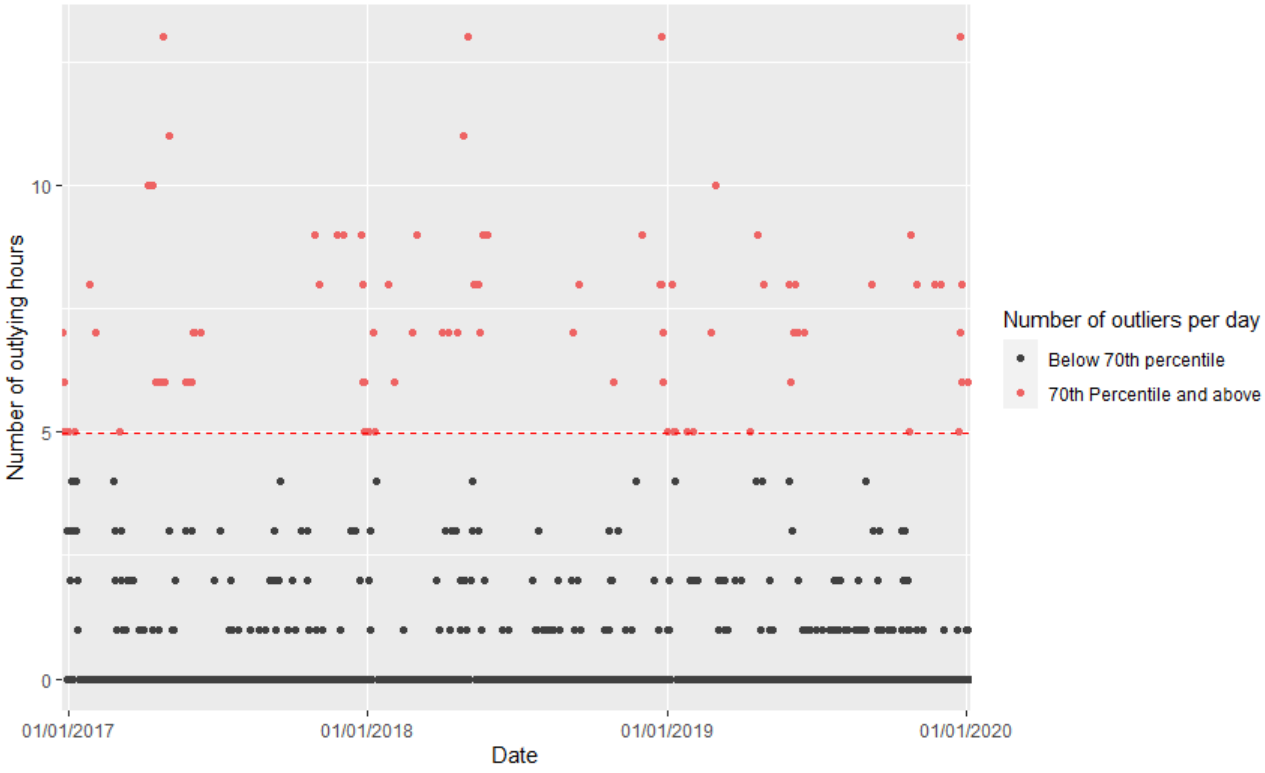

Figure 7: An overview of the number of outliers per day for all dates from 26/10/2016 till 05/01/2020

There seems to be a big influence of Christmas and the weekends around Easter, Ascension Day and Whit Sunday. The days before and after public holidays also occur frequently in these 90 days. Therefore, we also include these additional dates for other variables as a precaution. Furthermore, the dates which are a week after the holidays are also regularly labeled as an outlier. This is probably caused by the fact that special days can affect the next week through the lagged predictor *TOT_168*. We also find twelve days that are a toll free Saturday. In addition, we see fourteen Saturdays which are a week after a toll free Saturdays. The created predictors to account for these days are listed in Table 14 (Appendix A.3)

5.3 Altered Quantile Regression

For the second and 'altered' quantile regression, the predictors are added which are found in the peak analysis. Hence, the variable indicating if there is a holiday is now split into indicators for vacation periods and separate special days. In Figure 8, we see how the true observations in the test set are distributed across the predicted quantiles.

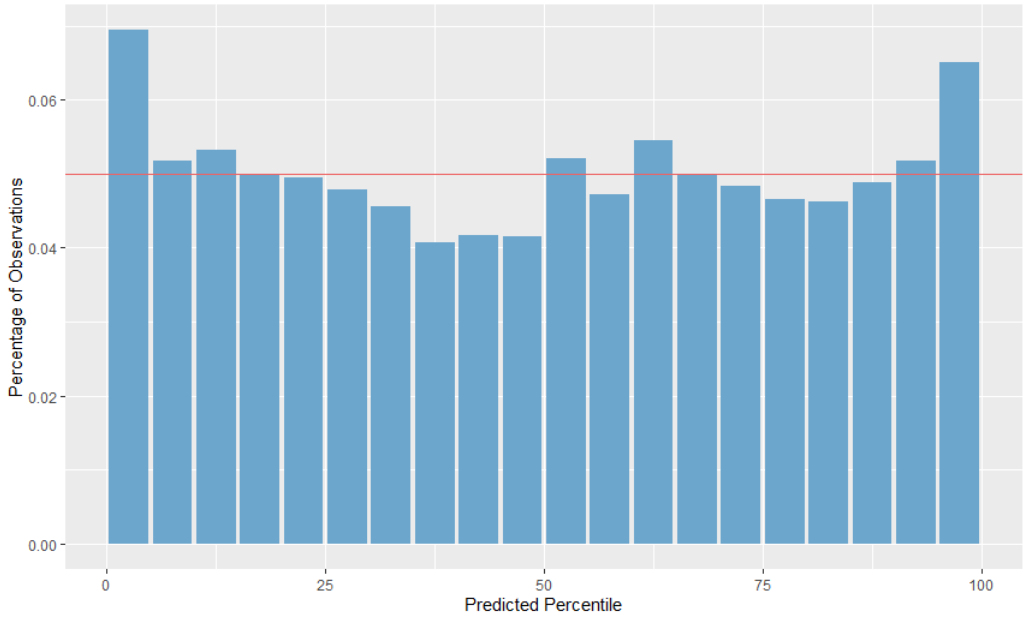

Figure 8: An overview of the distribution of the true observations of the test set across the predicted quantiles for the first quantile regressions

When we compare these results to the results of the first quantile regressions (Figure 6), we see two almost identical distributions. There is still overprediction in the quantiles below the 5^{th} percentile and above the 95^{th} percentile. Even more, we still underpredict between the 35^{th} and the 50^{th} percentile. Thus, the needed changes did not occur for the predicted quantiles.

This is illustrated by the quantile regression based on the 50^{th} percentile. This model selected the following variables for the sparsity pattern: *Wind_Force, Temperature, Air_Pressure, Cos_158* and *TOT_158*. The main difference with the previous model is the lack of a vacation or holiday indicating predictor. The coefficients for the other predictors stayed almost exactly the same. These coefficients are listed in Table 17 (Appendix B). In term of the outliers, the model yields 999 outlying observations spread throughout 275 days for the whole data set. Furthermore, the model has a MASE of 0.972, which is practically the same as for the first quantile regression.

5.4 Tuned Quantile Regression

Hence, the altered quantile regression does not show clear sings of improvement. Therefore, as mentioned in Section 4.2, we tune λ to obtain the quantile regression with lowest MASE. This is based on the quantile regression for the 50^{th} percentile. In Figure 9 we see the MASE for $\lambda = 1, 10, 20, 30, ..., 560$. The red observation is the MASE for $\lambda = 570.89$, which is the value for λ according to equation (5).

Here, we can see that the MASE is the lowest for $\lambda = 10$. The MASE for $\lambda = 10$ is

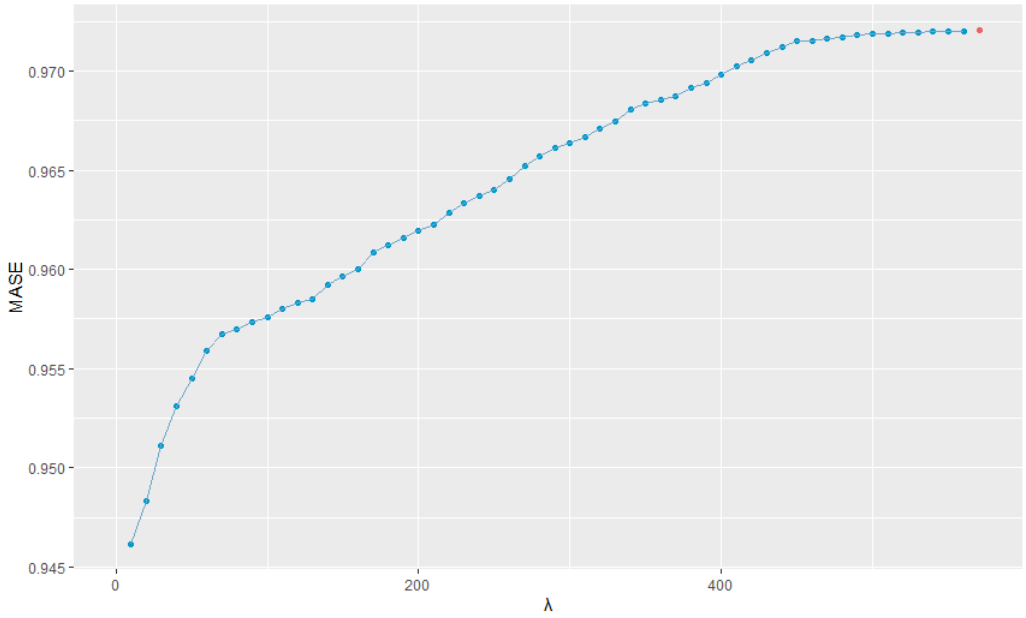

Figure 9: An overview of the MASE of the quantile regressions for $\lambda = 1, 10, 20, ..., 560, 570.89$

0.946, while the MASE is 0.972 for $\lambda = 570.89$. The MASE for $\lambda = 1$ is not showed, as it frequently yields singularity. Hence, due to the inconsistency for $\lambda = 1$, we choose $\lambda = 10$ for our new 'tuned' quantile regression. The sparsity pattern and coefficients for this model are listed in Table 18 (Appendix B). This tuned quantile regression yields 882 outliers across 276 days. Hence, for the tuned model, there is a decrease in the number of outliers. However, the number of days containing outliers remains similar.

Table 1: An overview of the three different quantile regressions with square root Lasso for the 50^{th} *percentile*

	First QR	**Altered QR**	**Tuned QR**
λ	568.18	570.89	10
Number of predictors	6	6	39
MASE	0.971	0.972	0.946
Number of outliers	1 000	999	882
Number of affected days	275	274	276

This means that tuning lambda has improved the point forecasting of the quantile regression based on the 50^{th} percentile. However, more importantly, we want to use the quantile regressions for probabilistic forecasting. In (Gneiting et al., 2006).

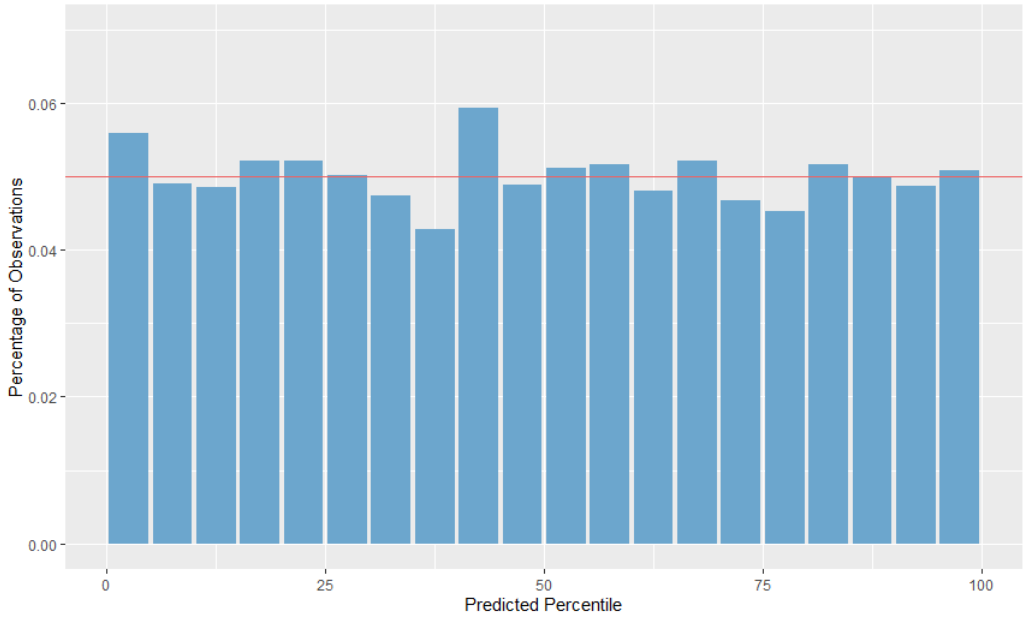

Figure 10: An overview of the distribution of the true observations of the test set across the predicted quantiles for the tuned quantile regressions

5.5 LSTM Network

For our point forecasts, we start with the LSTM network. For this model, we use all the predictors as found after the peak analysis. In Table 2, we see an overview of the average MASE and number of epochs per recurrent dropout rate.

Table 2: The average MASE and number of epochs per dropout rate for the period 2017-2019

Dropout Rate	MASE	Epochs
0	1.020	47.1
0.05	1.063	45.4
0.1	1.141	38.0
0.15	1.255	29.6
0.2	1.368	30.9
0.25	1.552	26.8
0.3	1.801	23.9

We see that the average MASE is the lowest when we use no recurrent dropout. However, the MASE is not below 1, which means that it does not outperform seasonal naive forecasting. Furthermore, without recurrent dropout, we see that the LSTM network is computationally heavier compared to the ones with recurrent dropout. As described in Section 4.3.2, we now choose our optimal recurrent dropout rate to tune the batch size. Hence, in Table 3, we see the average MASE and number of epochs per batch size with a recurrent dropout rate of 0.

Table 3: The average MASE and number of epochs per batch size for the altered sparsity pattern for the period 2017-2019

Batch Size	MASE	Epochs
32	1.031	36.2
64	1.020	47.1
128	1.010	64.8

The results show that the network with a batch size of 64 is slightly outperformed by the network with a batch size of 128. However, all models are less accurate than seasonal naive forecasting. Furthermore, we see a clear difference in computational power. As the LSTM network with a batch size of 32 only uses 36.2 epochs on average, compared to the 64.8 number of average epochs for a batch size of 128.

In summary, we see that for the tuned sparsity pattern, the LSTM network with a recurrent dropout rate of 0 and a batch size of 128 has the lowest MASE. Unfortunately, the MASE is still above 1. Hence, the model predicts less accurate than seasonal naive forecasting. Furthermore, the model also has a lower MASE than the point forecasts of the 50^{th} percentile quantile regression. With respect to the outliers, this model yields on average 2 017.9 outlying hours over 604.8 days. Which is surprising as the model with a batch size of 32 has 1 252.5 outliers across 422.5 days on average. Even though that model has a higher MASE.

5.6 Bayesian Linear Regression

For the Bayesian linear regressions, we use the predictors that are selected for the sparsity pattern by the tuned quantile regression. A statistical overview of the coefficients is shown in Table 23 (Appendix E). The model has a MASE of 1.062, which indicates that it is outperformed by seasonal naive forecasting. In terms of the outlying observations, the regression yields 959 outliers spread over 308 days.

5.7 Model overview

The results for the used models per sparsity pattern are stated in Table 4. For the LSTM network, the model was selected based on the lowest MASE.

In terms of the MASE, we see that the tuned quantile regression based on the 50^{th} percentile has highest accuracy. Furthermore, this model also yields the lowest number of outliers. Meanwhile, the LSTM network yields a much higher amount of outlying outliers. The Bayesian regression seems to be only a bit behind the quantile regression, in terms of the MASE and the number of outliers. Unfortunately, the MASE of the Bayesian regression and of the LSTM network are both higher than 1.

Table 4: An overview of the models for the two different sparsity patterns for the period 2017-2019

	QR for 50^{th} perc.	**LSTM**	**Bayesian**
Number of predictors	44	356	44
MASE	0.943	1.010	1.062
Number of outliers	884	2 017.9	959
Number of affected days	277	604.8	308

5.8 Covid-19

In this section we see the results of the models for the data till 2020, with the inclusion of the covid-19 related predictors. As described in Section 3.2, we perform the similar research steps as before. So, to recap briefly, first we perform a quantile regression with square root Lasso. Second, we tune the value of λ to get the lowest MASE and obtain probabilistic forecasts. Last, we use the predictor for our LSTM networks and Bayesian regressions.

5.8.1 Quantile Regression with Square Root Lasso (including data of 2020)

For the years 2017 till 2020, we see the distribution of the probabilistic forecasts in figure Figure 11. Here we see that we underpredict in the quantiles between the 35^{th} and 45^{th} percentile. Furthermore, we clearly overpredict above the 95^{th} percentile and below the 5^{th} percentile.

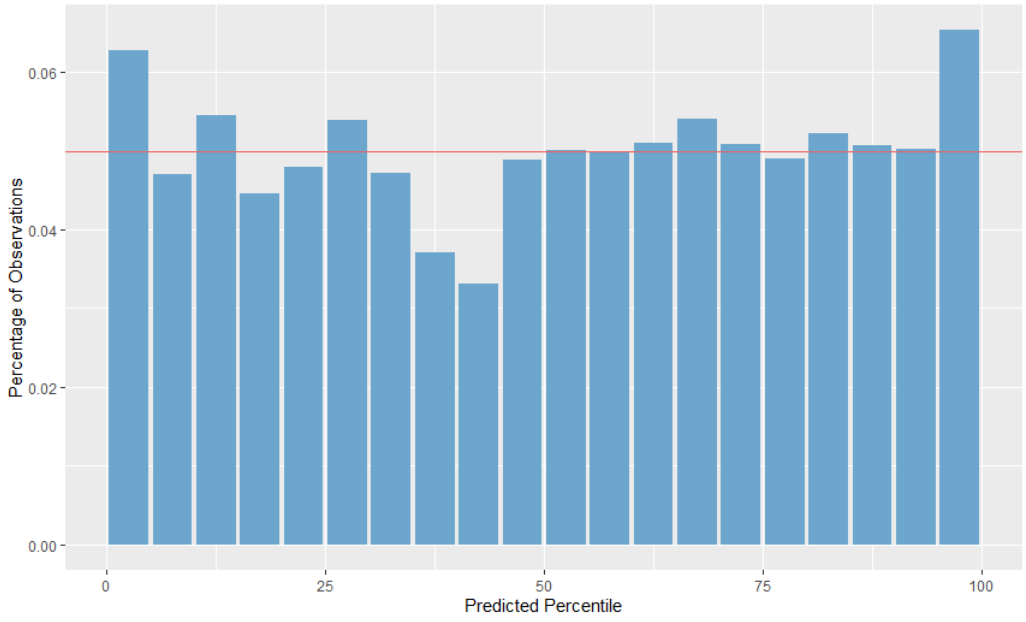

Figure 11: An overview of the distribution of test observations across the predicted quantiles for 2017 till 2020

The quantile regression for the 50^{th} percentile selected the following 7 predictors for the sparsity pattern: *Wind_Force, Temperature, Air_Pressure, Infections, Hospitalizations, Cos_210*

and *TOT_168*. This is based on $\lambda = 669.50$, which is the value for λ according to equation (5). Further explanation of these variables can be found in Table 10 and Table 15 (Appendix A). The coefficients of these seven variables are listed in Table 24 (Appendix F.1). This model has a MASE of 0.980, which indicates that it predict slightly better than seasonal naive forecasting. Furthermore, the model yields 1 307 outliers (3.7%), spread out over 357 different days (24.3%).

For the tuned sparsity pattern, an overview of the MASE for $\lambda = 1, 10, 20, ..., 660$ is given in Figure 12. This is based on the quantile regressions for the 50^{th} percentile. The red observation is the MASE for $\lambda = 669.50$.

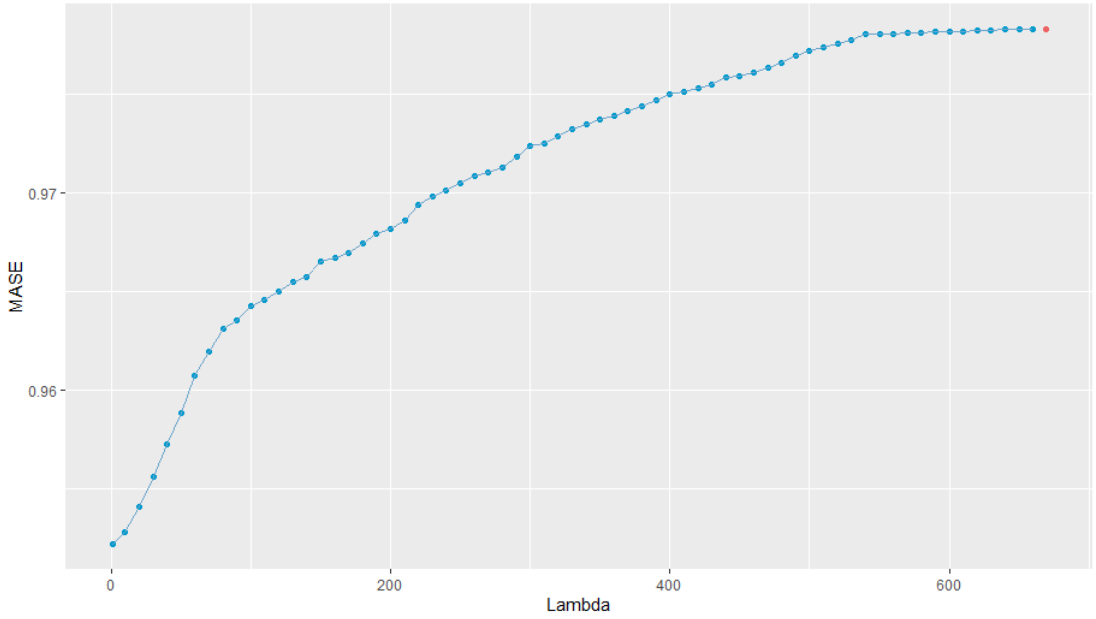

Figure 12: An overview of the MASE of the quantile regressions for 2017 till 2020 with $\lambda = 1, 10, 20, ..., 680, 669.50$

We clearly see that the MASE is the lowest for $\lambda = 1$. However, for several quantile regressions based on other percentiles, this model had problems with high levels of singularity. Therefore, we choose to use $\lambda = 10$. For this model the MASE is 0.953 and it yields 1 202 outliers spread out over 365 days. The model selected 45 predictors, for which the coefficients can be found in Table 25 (Appendix F.1). However, for the quantile regressions, we focus on probabilistic forecasting. In Figure 13 we see the distribution of the observations of the test set over the predicted quantiles. There, we see a distribution that seems to be uniform, which implies that we have found a robust model for probabilistic forecasting (Gneiting et al., 2006).

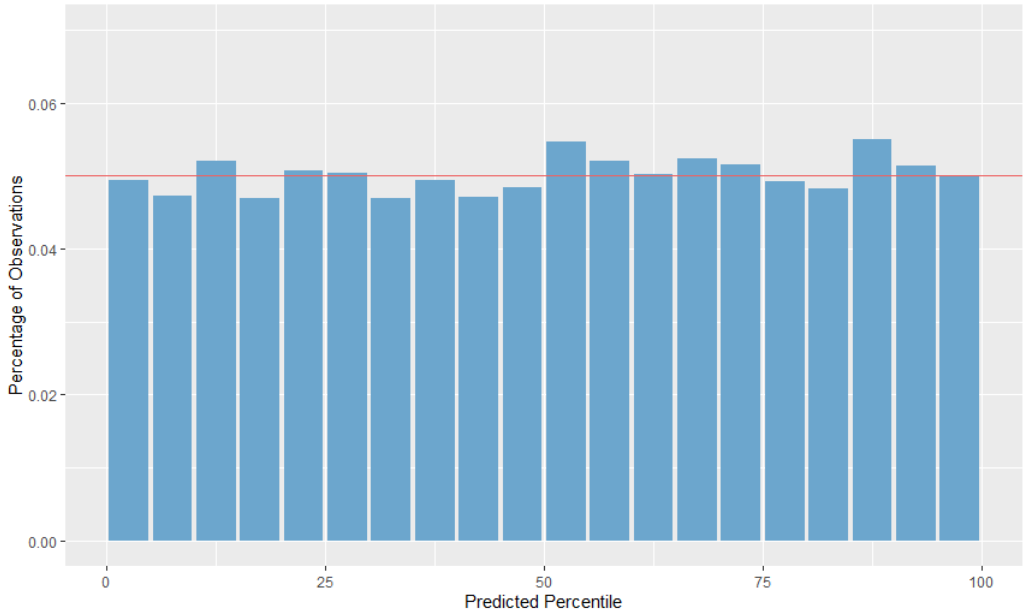

Figure 13: An overview of the distribution of test observations across the predicted quantiles for 2017 till 2020 with a tuned lambda

5.8.2 LSTM Network (including data of 2020)

In Table 5, we have the average MASE and number of epochs per recurrent dropout rate. For these LSTM networks, we used the same predictors as for the previous version. Furthermore, we included the five Covid-19 related variables as described in Table 15 (Appendix A).

Table 5: The average MASE and number of epochs per dropout rate for the period 2017-2020

	Mean	
Batch Size	MASE	Epochs
0	1.034	41.8
0.05	1.067	35.1
0.1	1.152	28.3
0.15	1.275	25.4
0.2	1.380	23.0
0.25	1.534	21.4
0.3	1.765	20.7

Once again, the MASE is the lowest when the network has no recurrent dropout. The average MASE for this model is not below 1. Hence, the model does not outperform seasonal naive forecasting. In addition, the average number of epochs is larger without recurrent dropout in comparison to the other networks. For the next step, we choose a recurrent dropout rate of 0 to tune the batch size. The average MASE and number of epochs per batch size are displayed in Table 6.

Table 6: The average MASE and number of epochs per batch size for the altered sparsity pattern for the period 2017-2020

	Mean	
Batch Size	MASE	Epochs
32	1.054	33.7
64	1.034	41.8
128	1.013	55.6

Again, the model with a batch size of 128 has on average the lowest MASE. However, none of the three models outperforms seasonal naive forecasting. Furthermore, the LSTM network with a batch size of 128 uses the most epochs on average. This model yields averagely 1 987.1 outliers across 593.5 days. This is slightly more than the 1 886.8 average outliers for a batch size of 64. With a batch size of 32, we get 2 769.2 outlying hours on average.

In summary, we see that the LSTM network with a recurrent dropout rate of 0 and a batch size of 128 has the lowest MASE. Unfortunately, the MASE is still above 1. Furthermore, it is also outperformed by the point forecasts of the quantile regression based on the 50^{th} percentile. In terms of the outliers, this model yields on average 1 987.1 outlying observation over 593.5 days. Which is slightly more than the network with a batch size of 32, while that model has a higher MASE.

5.8.3 Bayesian Linear Regression (including data of 2020)

For the Bayesian regression with the years 2017 till 2020, we once again used the sparsity pattern as selected by the tuned quantile regression based on the 50^{th} percentile. These coefficients are shown in Table 28 (Appendix F.3). The model has a MASE of 1.067, which is higher than for seasonal naive forecasting. The model marks 1 351 observations as outliers across 420 days.

5.8.4 Model Overview (including data of 2020)

In Table 7, the results per sparsity pattern are stated for the period of 2017 till 2020. The LSTM network with the lowest MASE was selected based on the tuning of the parameters.

Table 7: An overview of the point prediction models for the period 2017-2020

	QR for 50^{th} perc.	**LSTM**	**Bayesian**
Number of predictors	45	465	45
MASE	0.953	1.013	1.067
Number of outliers	1 202	1 987.1	1 351
Number of affected days	365	593.5	420

Once again, only the tuned quantile regression based on the 50^{th} percentile has a MASE

below 1. Furthermore, this model also yields the lowest number of outlying hours. The LSTM network is narrowly outperformed by seasonal naive forecasting. However, it has much more outliers than the Bayesian regression, while that model has a higher MASE.

5.9 Comparison Lagged Differences

At last, we have the results of the comparisons between our previous models and a model with the inclusion of lagged differences of the dependent variable. As mentioned in Section 4.4, we add lagged differences to our point prediction model with the lowest MASE. Hence, we do this to the LSTM network for both periods. However, as the point prediction of the quantile regression for the 50^{th} percentile unexpectedly has an even lower MASE, we also add the lagged differences to this quantile regression point forecast. The models with lagged differences use the same specifications as their counterparts without lagged differences.

Table 8: An overview of the 50^{th} percentile quantile regression and the LSTM Network with the inclusion of the lagged differences for the period 2017-2019

	Without Lagged Differences		*With Lagged Differences*	
	QR for 50^{th} perc.	**LSTM**	**QR for 50^{th} perc.**	**LSTM**
Number of predictors	44	356	38	361
MASE	0.943	1.010	0.726	0.833
Number of outliers	884	2 017.9	853	1 490
Number of affected days	277	604.8	332	455.0

In Table 9, we see an overview of the predictive performance of our LSTM network and quantile regression for the period 2017 till 2019. external predictors. We clearly see that there is a big improvement in the MASE when we include the special lagged differences. Furthermore, for the LSTM network, there is also a large decrease in the number of outliers. Hence, the model with the added lagged differences outperforms the model with only external predictors for the period 2017 till 2019. However, for the quantile regression, we see that the model indeed yields less outliers, but it does have more days containing outliers.

The coefficients for the 50^{th} percentile quantile regression are given in Table 29 (Appendix G). Here, we see a that the coefficient for *TOT_1* is 0.635. However, the other four lagged differences all have a coefficient close to zero, lying between -0.05 and 0.03.

In Table 30, we see the same for the period 2017-2020. Here, the models with lagged difference predictors have a lower MASE than the ones with only external predictors. Also, LSTM network yield fewer outliers and days containing outliers. Furthermore, the quantile regression yields once again fewer outliers, but also yields more days containing outliers.

Table 9: An overview of the 50^{th} percentile quantile regression and the LSTM Network with the inclusion of the lagged differences for the period 2017-2020

	Without Lagged Differences		*With Lagged Differences*	
	QR for 50^{th} perc.	**LSTM**	**QR for 50^{th} perc.**	**LSTM**
Number of predictors	45	465	44	470
MASE	0.953	1.013	0.734	0.823
Number of outliers	1 202	1 987.1	1 163	1 877.3
Number of affected days	365	593.5	444	554.0

The coefficients for the sparsity pattern of the quantile regression based on the 50^{th} percentile are listed in Table 30 (Appendix G). For this model the coefficient for *TOT_1* is 0.651. Once again, the coefficient for the other four lagged differences are close to zero and lie between -0.05 and 0.03.

6 Conclusion

In this paper, we predicted the traffic flows in the Westerscheldetunnel by improving naive forecasting methods. By modelling differences in possible influential external factors, we achieved statically robust probabilistic forecasts. This was accomplished after including dummy variables based on our peak analysis. Hence, we can conclude that we got statistically robust probability forecasting results for the quantile regressions for the period 2017 till 2019 and for the period 2017 till 2020.

Our two point forecasting methods did not outperform simple seasonal naive forecasting. For the point forecasts by the LSTM network it was therefore necessary to include lagged differences. However, for the point forecast, we can say that the quantile regression with square root Lasso based on the 50^{th} percentile was surprisingly the most accurate. This was true in terms of having the lowest MASE as well as yielding the least outliers. Thus, we can conclude that the 50^{th} percentile quantile regression was the most accurate point forecast.

For the point forecasts for the period 2017 till 2020, we see very similar results. However, the amount of outliers even decreased for the LSTM network. As these differences are relatively small, we can not conclude if this is caused by the new predictors or by an increase in the amount of observations. In addition, the LSTM network is thus outperformed by the quantile regression, which might indicate that it could not deal with the seasonality, as previously suggested by Zhang & Qi (2005).

Lastly, for the quantile regression with lagged differences, we saw that only the first lagged difference was greatly present in terms of the coefficients. Hence, the other four lagged differences

seem to contribute less to the prediction. More importantly, the inclusion of these lagged differences did make the models less usable. They do substantially improve the accuracy of our models, but they also limit the window in which the estimation could be made. Due to the fact that we need the data of the hour before, we can not make any long-term predictions with these models.

So, in conclusion, our improved naive forecasting did outperform simple seasonal naive forecasting in some cases. Though, the biggest improvement in accuracy was made when we included lagged differences alongside the external predictors. As shown, the models with the inclusion of these lagged differences easily outperform their counterparts with only external predictors. Furthermore, it seems that our method for probability forecasting was very suitable to predict the traffic volumes and the corresponding probabilities.

6.1 Executive Summary

Our suggested methods for improving seasonal naive forecasting, do in fact yield better prediction results. The best method is to use external variables and construct them in terms of the difference with one seasonal cycle ago. Then, we can make probabilistic forecasts based on quantile regressions with square root Lasso. When adding lagged differences of the dependent variable, we can also outperform simple seasonal naive point forecasting.

6.2 Discussion and Further Research

In the following section, we discuss several remarks that can be made about this research. Furthermore, we talk about some opportunities for further research.

Vacation Periods and Lockdowns

For the Netherlands and for Germany, different regions of the country have different vacation periods. In this research, these periods are combined. However, it can be argued that there is a difference in the impact of vacation periods when split across regions. This can be investigated further by including the vacation periods separately in future research. The same holds for the covid-19 situations in Belgium and Germany. For example, their lockdowns are not taken into account by our model.

Total Traffic Activity

In this research, we predict the total traffic activity in Westerscheldetunnel. However, for future research it might be interesting to split this into both driving directions and even into the

separate vehicle length classes. We do believe that the methods presented in this paper are very suitable for such research.

Coefficient TOT_168

An interesting topic for future research is the coefficient for TOT_168. Throughout the paper and the models, we see that this coefficient is always very close to 1. This is intuitively logical, as we model the difference in circumstances between the traffic volume now and 168 hours ago. Therefore, it is interesting to see if there is a statistical relation between the accuracy of a model and the coefficient of TOT_168.

Kurtosis and Skewness

It could be that the quantile regression had some problems with the kurtosis and skewness of several covid-19 predictors. Hence, it could be argued that the sparsity pattern is not correctly specified. However, the overall pattern of the MASE is clearly visible for the tuned sparsity pattern. Therefore, we believe that the tuned sparsity pattern for the period 2017-2020 is a correct estimation of the influential predictors. The same holds for the tuned sparsity pattern for the period 2017-2019.

References

Bayes, T. (1763). An essay towards solving a problem in the doctrine of chances. *Philosophical transactions of the Royal Society of London*, *53*, 370–418.

Belloni, A., Chernozhukov, V., & Fernández-Val, I. (2011). Conditional quantile processes based on series or many regressors. *Journal of Econometrics*, *213*(1), 4–29.

Belloni, A., Chernozhukov, V., & Wang, L. (2011). Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, *98*(4), 791–806.

Breiman, L. (1995). Better subset regression using the nonnegative garrote. *Technometrics*, *37*(4), 373–384.

Cameron, A. C., & Trivedi, P. K. (2005). *Microeconometrics: Methods and applications* (Vol. 1). Cambridge (UK): Cambridge University Press.

Dalalyan, A. S., Hebiri, M., & Lederer, J. (2017). On the prediction performance of the lasso. *Bernoulli*, *23*(1), 552–581.

Das, S., & Tsapakis, I. (2020). Interpretable machine learning approach in estimating traffic volume on low-volume roadways. *International Journal of Transportation Science and Technology*, *9*(1), 76–88.

Dawid, A. P. (1984). Present position and potential developments: Some personal views statistical theory the prequential approach. *Journal of the Royal Statistical Society: Series A (General)*, *147*(2), 278–290.

Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., & Smith, N. (2020). Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv: 2002.06305*.

Fernández, C., & Steel, M. F. (1998). On bayesian modeling of fat tails and skewness. *Journal of the American Statistical Association*, *93*(441), 359–371.

Fienberg, S. E. (2006). When did Bayesian inference become "Bayesian"?. *Bayesian Analysis*, *1*(1), 1–40.

Fourier, J. B. J. (1822). *The analytical theory of heat*. Cambridge (UK): Cambridge University Press.

Gal, Y., & Ghahramani, Z. (2016a). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning* (pp. 1050–1059).

Gal, Y., & Ghahramani, Z. (2016b). A theoretically grounded application of dropout in recurrent neural networks. *Advances in Neural Information Processing Systems*, *29*.

Gao, P., Zhang, R., & Yang, X. (2020). The application of stock index price prediction with neural network. *Mathematical and Computational Applications*, *25*(3), 53.

Gardner, M. W., & Dorling, S. (1998). Artificial neural networks (the multilayer perceptron) — a review of applications in the atmospheric sciences. *Atmospheric Environment*, *32*(14-15), 2627–2636.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*, 721–741.

Gers, F. A., Eck, D., & Schmidhuber, J. (2002). Applying LSTM to time series predictable through time-window approaches. In R. Tagliaferri & M. Marinaro (Eds.), *Neural nets wirn vietri-01* (pp. 193–200). New York City: Springer.

Gneiting, T. (2008). Probabilistic forecasting. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 319–321.

Gneiting, T., & Katzfuss, M. (2014). Probabilistic forecasting. *Annual Review of Statistics and Its Application*, *1*, 125–151.

Gneiting, T., Larson, K., Westrick, K., Genton, M. G., & Aldrich, E. (2006). Calibrated probabilistic forecasting at the stateline wind energy center: The regime-switching space–time method. *Journal of the American Statistical Association*, *101*(475), 968–979.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. Cambridge (USA): MIT press.

Greenberg, E. (2012). *Introduction to Bayesian econometrics*. Cambridge (UK): Cambridge University Press.

Gu, Y., McCallum, A., & Towsley, D. (2005). Detecting anomalies in network traffic using maximum entropy estimation. In *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement* (pp. 32–32). Berkely: USENIX Association.

Hair, J., Hult, G. T. M., Ringle, C., & Sarstedt, M. (2022). *A primer on partial least squares structural equation modeling (pls-sem)*. doi: 10.1007/978-3-030-80519-7

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2). New York City: Springer.

Hastie, T., Tibshirani, R., & Wainwright, M. (2015). Statistical learning with sparsity. *Monographs on Statistics and Applied Probability*, *143*.

Heaton, J. (2017). The number of hidden layers. *Heaton Research*.

Hebiri, M., & Lederer, J. (2012). How correlations influence lasso prediction. *IEEE Transactions on Information Theory*, *59*(3), 1846–1854.

Heij, C., De Boer, P., Franses, P., Kloek, T., & Van Dijk, H. (2004). *Econometric methods with applications in business and economics* (Vol. 1). Oxford: Oxford University Press.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, *79*(8), 2554–2558.

Hubert, M., & Debruyne, M. (2009). Breakdown value. *Wiley Interdisciplinary Reviews: Computational Statistics*, *1*(3), 296–302.

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice* (Vol. 2). Melbourne: OTexts.

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

Kandel, I., & Castelli, M. (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express*, *6*(4), 312–315.

Kim, S., & Kim, H. (2016). A new metric of absolute percentage error for intermittent demand forecasts. *International Journal of Forecasting*, *32*(3), 669–679.

Koenker, R. (2000). Galton, Edgeworth, Frisch, and prospects for quantile regression in econometrics. *Journal of Econometrics*, *95*(2), 347–374.

Koenker, R., & Bassett Jr, G. (1978). Regression quantiles. *Econometrica: Journal of the Econometric Society*, 33–50.

Koenker, R., & Hallock, K. F. (2001). Quantile regression. *Journal of Economic Perspectives*, *15*(4), 143–156.

Laplace, P. S. (1829). *Essai philosophique sur les probabilités*. H. Remy.

Leadbetter, M. R. (1991). On a basis for 'peaks over threshold'modeling. *Statistics & Probability Letters*, *12*(4), 357–362.

Makridakis, S., & Hibon, M. (1979). Accuracy of forecasting: An empirical investigation. *Journal of the Royal Statistical Society: Series A (General)*, *142*(2), 97–125.

McNeil, A. J., & Saladin, T. (1997). The peaks over thresholds method for estimating high quantiles of loss distributions. In *Proceedings of 28th International ASTIN Colloquium* (pp. 23–43).

Meinshausen, N., & Yu, B. (2009). Lasso-type recovery of sparse representations for high-dimensional data. *The Annals of Statistics*, *37*(1), 246–270.

Papagiannaki, K., Taft, N., & Lakhina, A. (2004). A distributed approach to measure ip traffic matrices. In *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement* (pp. 161–174). New york: ACM.

Rousseeuw, P. J. (1991). Tutorial to robust statistics. *Journal of Chemometrics*, *5*(1), 1–20.

Rousseeuw, P. J., & Croux, C. (1993). Alternatives to the median absolute deviation. *Journal of the American Statistical association*, *88*(424), 1273–1283.

Rousseeuw, P. J., & Hubert, M. (2011). Robust statistics for outlier detection. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *1*(1), 73–79.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.

Santosa, F., & Symes, W. W. (1986). Linear inversion of band-limited reflection seismograms. *SIAM Journal on Scientific and Statistical Computing*, *7*(4), 1307–1330.

Sekuła, P., Marković, N., Vander Laan, Z., & Sadabadi, K. F. (2018). Estimating historical hourly traffic volumes via machine learning and vehicle probe data: A maryland case study. *Transportation Research Part C: Emerging Technologies*, *97*, 147–158.

Semeniuta, S., Severyn, A., & Barth, E. (2016). Recurrent dropout without memory loss. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers.* Osaka: The COLING 2016 Organizing Committee.

Shmueli, G. (2010). To explain or to predict? *Statistical Science*, *25*(3), 289–310.

Siegel, A. (2016). *Practical business statistics* (Vol. 7). Cambridge (USA): Academic Press.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288.

Waldmann, E. (2018). Quantile regression: a short story on how and why. *Statistical Modelling*, *18*(3-4), 203–218.

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. *Neural Computation*, *31*(7), 1235–1270.

Zhang, G. P., & Qi, M. (2005). Neural network forecasting for seasonal and trend time series. *European Journal of Operational Research*, *160*(2), 501–514.

Zhou, Q., Zhu, Z., Xian, G., & Li, C. (2022). A novel regression method for harmonic analysis of time series. *ISPRS Journal of Photogrammetry and Remote Sensing*, *185*, 48–61.

Zhu, Z., & Woodcock, C. E. (2014). Continuous change detection and classification of land cover using all available landsat data. *Remote Sensing of Environment*, *144*, 152–171.

Appendix

A Variable Analysis

Most predictive variables are defined in terms of their difference between the current observation and the observation of a week (168 hours) earlier. It includes weather data[4], holidays and covid-19 related variables[5]

A.1 Regular Variables

Table 10: The predictive variables for the variables analysis with the data from 2017 till 2019

Variable	Description
Wind_Force	The difference in wind force measured in m/s
Temperature	The difference in temperature measured in ^{o}C
Sunshine	The difference in the fraction of the hour that the sun shine
Rain	The difference in the fraction of the hour that it rains
Rainfall	The difference in the amount of rainfall in mm[6]
Air_Pressure	The difference in airpressure measured in Pascal
Fog	The difference between the booleans indicating if there is fog[7]
Snow	The difference between the booleans indicating if there is snow[7]
Thunder	The difference between the booleans indicating if there is thunder[7]
Ice	The difference between the booleans indicating if there is ice[7]
Toll_Free	The difference between the booleans indicating if there is a toll free day
Holiday_Dutch	The difference between the booleans indicating if there is a Dutch holiday
Holiday_Belgian	The difference between the booleans indicating if there is a Belgian holiday
Holiday_German	The difference between the booleans indicating if there is a German holiday
Sin_s	The sinusoid $sin((2\pi t_i)/(sT))$, for $s = 1, ..., 158$
Cos_s	The sinusoid $cos((2\pi t_i)/(sT))$, for $s = 1, ..., 158$
TOT_168	The value of TOT for the previous week

A.2 Holidays and Toll Free Days

Below, we have the marked dates to account for the public holidays and vacation periods for the Netherlands[8], Belgium[9] and Germany[10]. Furthermore, we see the toll-free Saturdays.

[4]Koninklijk Nederlands Metereologisch Instituut. (2022). *Uurwaarden van Weerstations.* Retrieved from https://daggegevens.knmi.nl/klimatologie/uurgegevens

[5]Rijksinstituut voor Volksgezondheid en Milieu. (2022). *Covid-19 cumulatieve aantallen per gemeente.* Retrieved from https://data.rivm.nl/meta/srv/dut/catalog.search#/metadata/1c0fcd57-1102-4620-9cfa-441e93ea5604

[6]Observations with a rainfall of < 0.05 mm were given a value of -1. We adjusted this to a more representative value of 0.025.

[7]Missing observations are replaced with a zero.

[8]Kalender Nederland. (n.d.). *Kalender Nederland.* Retrieved from http://kalender-nl.nl/

[9]Kalender België. (n.d.). *Kalender België.* Retrieved from http://kalender-be.be/

[10]Schulferien.org. (n.d.). *Schulferien Deutschland.* Retrieved from https://www.schulferien.org/deutschland/ferien/

Table 11: The dates that are marked as a public holiday or vacation period per country, which are used in the quantile regression before the outlier analysis

Country	**Period**	**Dates** *(YYYYMMDD)*
Netherlands	19/12/2016 - 31/12/2017	20161224:20161231, 20170101:20170108, 20170218:20170228, 20170301:20170305, 20170414:20170417, 20170422:20170428, 20170505, 20170525:20170528, 20170602:20170605, 20170708:20170731, 20170801:20170831, 20170901:20170903, 20171014:20171029, 20171223:20171231,
	01/01/2018 - 31/12/2018	20180101:20180107, 20180217:20180228, 20180301:20180304, 20180330:20180331, 20180401:20180402, 20180427:20180430, 20180501:20180506, 20180510:20180513, 20180518:20180521, 20180707:20180731, 20180801:20180831, 20180901:20180902, 20181013:20181028, 20181222:20181231,
	01/01/2019 - 05/12/2020	20190101:20190106, 20190216:20190228, 20190301:20190303, 20190419:20190422, 20190427:20190430, 20190501:20190505, 20190530:20190531, 20190601:20190602, 20190607:20190610, 20190706:20190731, 20190801:20190831, 20190901, 20191012:20191027, 20191221:20191231,
Belgium	19/12/2016 - 31/12/2017	20161224:20161231, 20170101:20170108, 20170225:20170228, 20170301:20170305, 20170401:20170417, 20170501, 20170525:20170528, 20170602:20170605, 20170701:20170731, 20170801:20170831, 20170901:20170903, 20171028:20171031, 20171101:20171105, 20171111, 20171223:20171231,
	01/01/2018 - 31/12/2018	20180101:20180107, 20180210:20180218, 20180331, 20180401:20180415, 20180501, 20180510:20180513, 20180518:20180521, 20180630, 20180701:20180731, 20180801:20180831, 20180901:20180902, 20181027:20181031, 20181101:20181104, 20181111, 20181222:20181231,
	01/01/2019 - 05/12/2020	20190101:20190106, 20190302:20190310, 20190406:20190422, 20190501, 20190530:20190531, 20190601:20190602, 20190607:20190610, 20190629:20190630, 20190701:20190731, 20190801:20190831, 20190901, 20191026:20191031, 20191101:20191103, 20191111, 20191221:20191231, 20200101:20200105
Germany	19/12/2016 - 31/12/2017	20161219:20161231, 20170101:20170108, 20170128:20170131, 20170201:20170228, 20170301:20170319, 20170401:20170423, 20170501, 20170520:20170528, 20170602:20170618, 20170622:20170630, 20170701:20170731, 20170801:20170831, 20170901:20170911, 20170930, 20171001:20171031, 20171101:20171105, 20171122, 20171221:20171231,
	01/01/2018 - 31/12/2018	20180101:20180114, 20180201:20180225, 20180305:20180331, 20180401:20180407, 20180428:20180430, 20180501, 20180507:20180531, 20180601:20180603, 20180623:20180630, 20180701:20180731, 20180801:20180831, 20180901:20180910, 20180929:20180930, 20181001:20181028, 20181101:20181102, 20181219:20181231,
	01/01/2019 - 05/12/2020	20190101:20190113, 20190131, 20190201:20190228, 20190301:20190315, 20190404:20190430, 20190501, 20190513:20190517, 20190531, 20190601:20190602, 20190607:20190630, 20190701:20190731, 20190801:20190831, 20190901:20190910, 20190928:20190930, 20191001:20191031, 20191101, 20191120, 20191220:20191231, 20200101:20200105

Table 12: The dates that are a toll-free Saturday

Country	**Period**	**Dates** *(YYYYMMDD)*
Toll-Free	18/12/2016 - 31/12/2017	20170128, 20170225, 20170408, 20170909, 20171028, 20171125,
	01/01/2018 - 31/12/2018	20180127, 20180224, 20180414, 20180908, 20181027, 20181124,
	01/01/2019 - 05/12/2020	20190126, 20190223, 20190413, 20190831, 20190914, 20191026, 20191123
	01/01/2020 - 03/01/2021	20200125, 20200229, 20200912, 20201128

A.3 Vacations and Special Days

Here, we have the altered versions of these holiday variables, which are split into vacation periods and several special days. This renewed version of the old holiday variable is made after the outlier analysis, as stated in Section 5.2. The indicator for a toll-free Saturday is still used after these changes.

Table 13: The dates that are marked as a vacation period per country

Country	**Period**	**Date** *(YYYYMMDD)*
Netherlands	19/12/2016 - 31/12/2017	20161224:20161231, 20170101:20170108, 20170218:20170228, 20170301:20170305, 20170422:20170430, 20170708:20170731, 20170801:20170831, 20170901:20170903, 20171014:20171029, 20171223:20171231
	01/01/2018 - 31/12/2018	20180101:20180107, 20180217:20180228, 20180301:20180304, 20180428:20180430,20180501:20180506, 20180707:20180731, 20180801:20180831, 20180901:20180902, 20181013:20181028, 20181222:20181231
	01/01/2019 - 31/12/2019	20190101:20190106, 20190216:20190228, 20190301:20190303, 20190427:20190430, 20190501:20190505, 20190706:20190731, 20190801:20190831, 20190901, 20191012:20191027, 20191221:20191231
	01/01/2020 - 03/01/2021	20200101:20200105, 20200215:20200229, 20200301, 20200425:20200430, 20200501:20200510, 20200704:20200731, 20200801:20200830, 20201017:20201025, 20201219:20201231, 20210101:20210103
Belgium	19/12/2016 - 31/12/2017	20161224:20161231, 20170101:20170108, 20170225:20170228, 20170301:20170305, 20170401:20170417, 20170701:20170731, 20170801:20170831, 20170901:20170903, 20171028:20171031, 20171101:20171105, 20171223:20171231,
	01/01/2018 - 31/12/2018	20180101:20180107, 20180210:20180218, 20180401:20180415, 20180630, 20180701:20180731, 20180801:20180831, 20180901:20180902, 20181027:20181031, 20181101:20181104, 20181222:20181231,
	01/01/2019 - 31/12/2019	20190101:20190106, 20190302:20190310, 20190406:20190422, 20190629:20190630, 20190701:20190731, 20190801:20190831, 20190901, 20191026:20191031, 20191101:20191103, 20191221:20191231
	01/01/2020 - 03/01/2021	20200101:20200105, 20200222:20200229, 20200301, 20200404:20200419, 20200701:20200731, 20200801:20200831, 20201031, 20201101:20201108, 20201219:20201231, 20210101:20210103
Germany	19/12/2016 - 31/12/2017	20161219:20161231, 20170101:20170108, 20170128:20170131, 20170201:20170228, 20170301:20170319, 20170401:20170423, 20170520:20170528, 20170602:20170618, 20170622:20170630, 20170701:20170731, 20170801:20170831, 20170901:20170911, 20170930, 20171001:20171031, 20171101:20171105, 20171122, 20171221:20171231,
	01/01/2018 - 31/12/2018	20180101:20180114, 20180201:20180225, 20180305:20180331, 20180401:20180407, 20180428:20180430, 20180507:20180531, 20180601:20180603, 20180623:20180630, 20180701:20180731, 20180801:20180831, 20180901:20180910, 20180929:20180930, 20181001:20181028, 20181101:20181102, 20181219:20181231
	01/01/2019 - 31/12/2019	20190101:20190113, 20190131, 20190201:20190228, 20190301:20190315, 20190404:20190430, 20190611:20190630, 20190701:20190731, 20190801:20190831, 20190901:20190910, 20190928:20190930, 20191001:20191031, 20191101, 20191220:20191231
	01/01/2020 - 03/01/2021	20200101:20200106, 20200201:20200229, 20200301:20200315, 20200328:20200331, 20200401:20200426, 20200508, 20200516:20200531, 20200601:20200614, 20200620:20200620, 20200701:20200731, 20200801:20200831, 20200901:20200913, 20201003:20201031, 20201101:20201108, 20201118, 20201219:20201231, 20210101:20210103

Table 14: The public holidays and special days

Dummy Variable	Public Holiday / Special Day Explanation
New_Year	New Year's Eve: 31/12
New_Years_Eve	New Year: 01/01
New_Year_After	The day after New Year's Eve: 02/01
Christmas_Vacation_After	The first Monday after the Dutch and Belgian Christmas Vacation
Easter_Weekend	The weekend of Easter (including Good Friday and Easter Monday)
Easter_Before	The Thursday before Easter (Maundy Thursday)
Ascension_Day_Weekend	The weekend of Ascension Day (Thursday till Sunday)
Ascension_Day_Before	The Wednesday before Ascension Day
Whit_Sunday_Weekend	The weekend of Whit Sunday (including Whit Monday)
Whit_Sunday_Before	The Thursday and Friday before Whit Sunday
Christmas	Christmas' Eve and Christmas: 24-26/12
Christmas_After	The days following Christmas: 27-29/12
Christmas_Before	The day before Christmas: 23/12
Sugar_Feast	The Sugar Feast
Sugar_Feast_After	The day after the Sugar Feast
Kings_Day	The Dutch King's Day: 27/04
Kings_Day_After	The day after the Dutch King's Day: 28/04
Remembrance_ofthe_Death	The Dutch day Remembrance of the Death: 04/05
Liberation_Day	The Dutch Liberation Day: 05/05
Workers_Day	International Workers' Day: 01/05
Armistice_Day	The Belgian day Armistice Day: 11/11
Belgian_National_Day	The Belgian national holiday: 21/07
German_Unity_Day	German Unity Day: 03/10
Maintenance	The yearly maintenance night
Incident	Hours that there was inconvenience due to an incident

A.4 Covid-19 Variables

Following, we see an overview of the added predictive variables for the covid-19 pandemic. These variables are combined with the predictors in Table 10 and the altered vacation variables as mentioned in Appendix A.3.

Table 15: The additional covid-19 related variables for the variables analysis with the inclusion of the year 2020

Variable	Description
Infections	The difference in the number of daily infections with covid-19
Hospitalizations	The difference in the number of daily hospitalizations due to covid-19
Deceased	The difference in the number of deceased covid-19 patients
Lockdown_Intelligent	The difference between the booleans indicating if there is an intelligent lockdown
Lockdown_Hard	The difference between the booleans indicating if there is a hard lockdown

B Sparsity Pattern

Below are the coefficients for the variables selected by the quantile regression with square root Lasso. As mentioned before in Section 4.1, coefficients are set to zero when $\beta < 0.0001$.

Table 16: The predicting variables that are selected for the sparsity pattern in the first quantile regression for the period 2017-2019

Variable	Coefficient
Wind_Force	-1.014
Temperature	1.195
Air_Pressure	0.426
Holiday_Dutch	-1.865
Cos_158	1.036
TOT_168	0.995

Table 17: The predicting variables that are selected for the sparsity pattern in the altered quantile regression for the period 2017-2019

Variable	Coefficient
Wind_Force	-1.000
Temperature	1.165
Air_Pressure	0.417
Cos_158	1.057
TOT_168	0.995

Table 18: The predicting variables that are selected for the sparsity pattern in the tuned quantile regression with $\lambda = 10$ for the period 2017-2019

Variable	Coefficient	Variable	Coefficient
Wind_Force	-1.054	*Christmas*	-37.016
Temperature	1.106	*Christmas_Before*	7.557
Sun	9.984	*Christmas_After*	-20.383
Rain	-2.372	*Sugar_Feast*	-20.242
Air_Pressure	0.367	*Sugar_Feast_After*	-13.983
Fog	-1.004	*Kings_Day*	-175.326
Snow	-37.916	*Kings_Day_After*	-24.436
Thunder	1.316	*Liberation_Day*	4.773
Vacation_Dutch	-7.817	*Armistice_Day*	-1.250
Vacation_Belgian	-1.034	*Belgian_National_Day*	24.227
Vacation_German	-2.143	*Toll_Free*	142.754
New_Year	-17.947	*Maintenance*	-39.257
New_Years_Eve	-90.023	*Incident*	-11.216
New_Year_After	-4.964	*Sin_1*	0.505
Christmas_Vacation_After	-17.541	*Cos_1*	-0.403
Easter_Weekend	-8.996	*Sin_2*	2.288
Easter_Before	13.299	*Cos_8*	1.609
Ascension_Day_Weekend	-11.847	*Cos_9*	1.119
Ascension_Day_Before	21.692	*TOT_168*	0.993
Whit_Sunday_Weekend	-12.675		

C Outliers

In this section, we have the 91 days that contain the most outliers. This is determined by the z-scores of the outliers.

Table 19: The 90 days which contain the most by outlying hours

Date	Number of Outliers	Explanation	Date	Number of Outliers	Explanation
27-4-2017	13	King's Day	24-2-2018	7	Toll Free Saturday
4-5-2018	13	Remembrance of the Dead	2-4-2018	7	Easter Weekend
25-12-2018	13	Christmas	9-4-2018	7	Week after Easter Weekend
25-12-2019	13	Christmas	21-4-2018	7	Week after Toll Free Saturday
4-5-2017	11	Remembrance of the Dead	18-5-2018	7	Days before Whit Sunday Weekend
27-4-2018	11	King's Day	8-9-2018	7	Toll Free Saturday
8-4-2017	10	Toll Free Saturday	27-12-2018	7	Days after Christmas
15-4-2017	10	Easter Weekend	23-2-2019	7	Toll Free Saturday
2-3-2019	10	Week after Toll Free Saturday	5-6-2019	7	Day after Sugar Feast
28-10-2017	9	Toll Free Saturday	7-6-2019	7	Days before Whit Sunday Weekend
25-11-2017	9	Toll Free Saturday	10-6-2019	7	Whit Sunday Weekend
2-12-2017	9	Week after Toll Free Saturday	17-6-2019	7	Week after Whit Sunday Weekend
25-12-2017	9	Christmas	24-12-2019	7	Christmas
3-3-2018	9	Week after Toll Free Saturday	28-12-2016	6	Days after Christmas
21-5-2018	9	Whit Sunday Weekend	17-4-2017	6	Easter Weekend
28-5-2018	9	Week after Whit Sunday Weekend	24-4-2017	6	Week after Easter Weekend
1-12-2018	9	Week after Toll Free Saturday	28-4-2017	6	Day after King's Day
22-4-2019	9	Easter Weekend	25-5-2017	6	Ascension Day Weekend
26-10-2019	9	Toll Free Saturday	26-5-2017	6	Ascension Day Weekend
28-1-2017	8	Toll Free Saturday	1-6-2017	6	Week after Ascension Day Weekend
4-11-2017	8	Week after Toll Free Saturday	27-12-2017	6	Days after Christmas
26-12-2017	8	Christmas	28-12-2017	6	Days after Christmas
27-1-2018	8	Toll Free Saturday	3-2-2018	6	Week after Toll Free Saturday
10-5-2018	8	Ascension Day Weekend	27-10-2018	6	Toll Free Saturday
11-5-2018	8	Ascension Day Weekend	28-12-2018	6	Days after Christmas
17-5-2018	8	Days before Whit Sunday Weekend	31-5-2019	6	Ascension Day Weekend
15-9-2018	8	Week after Toll Free Saturday	27-12-2019	6	Days after Christmas
24-12-2018	8	Christmas	2-1-2020	6	Day after New Year
26-12-2018	8	Christmas	27-12-2016	5	Days after Christmas
7-1-2019	8	First day after Christmas Vacation	29-12-2016	5	Days after Christmas
8-1-2019	8	Week after New Year	2-1-2017	5	Day after New Year
29-4-2019	8	Week after Easter Weekend	9-1-2017	5	First day after Christmas Vacation
30-5-2019	8	Ascension Day Weekend	4-3-2017	5	Week after Toll Free Saturday
6-6-2019	8	Days before Whit Sunday Weekend	29-12-2017	5	Days after Christmas
7-9-2019	8	Week after Toll Free Saturday	2-1-2018	5	Day after New Year
2-11-2019	8	Week after Toll Free Saturday	9-1-2018	5	Week after the day after New Year
23-11-2019	8	Toll Free Saturday	10-1-2018	5	Week after Christmas Vacation
30-11-2019	8	Week after Toll Free Saturday	2-1-2019	5	Day after New Year
26-12-2019	8	Christmas	9-1-2019	5	Week after the day after New Year
26-12-2016	7	Christmas	10-1-2019	5	Week after Christmas Vacation
4-2-2017	7	Week after Toll Free Saturday	26-1-2019	5	Christmas
2-6-2017	7	Days before Whit Sunday Weekend	2-2-2019	5	Week after Toll Free Saturday
5-6-2017	7	Whit Sunday Weekend	13-4-2019	5	Toll Free Saturday
12-6-2017	7	Week after Whit Sunday Weekend	24-10-2019	5	Week after an incident
8-1-2018	7	Week after New Year	23-12-2019	5	Day before Christmas
		First day after Christmas Vacation			

D LSTM Network

In this section, we see the more detailed results for the LSTM networks. As mentioned in Section 4.3.2, the statistical properties are calculated based on the 50 LSTM networks that are constructed per parameter-tuning step.

Table 20: A statistical overview of the MASE and number of epochs per dropout rate for the period 2017-2019

	Mean		**St. Dev.**		**Max**		**Min**	
Dropout Rate	MASE	Epochs	MASE	Epochs	MASE	Epochs	MASE	Epochs
0	1.020	47.1	0.033	11.8	1.125	70	0.969	18
0.05	1.063	45.4	0.045	10.2	1.181	74	0.989	29
0.1	1.141	38.0	0.079	10.1	1.313	56	1.014	19
0.15	1.255	29.6	0.095	10.0	1.531	66	1.098	14
0.2	1.368	30.9	0.105	13.2	1.621	69	1.148	11
0.25	1.552	26.8	0.134	10.0	1.937	50	1.319	10
0.3	1.801	23.9	0.155	9.025	2.136	64	1.436	10

Table 21: A statistical overview of the MASE and number of epochs per batch size for the sparsity pattern from the altered quantile regression for the period 2017-2019

	Mean		**St. Dev.**		**Max**		**Min**	
Batch Size	MASE	Epochs	MASE	Epochs	MASE	Epochs	MASE	Epochs
32	1.031	36.2	0.027	14.9	1.010	81	0.991	13
64	1.020	47.1	0.033	11.8	1.125	70	0.969	18
128	1.010	64.8	0.033	20.5	1.092	123	0.960	25

E Bayesian Linear Regression

Following, we have a statistical overview of the coefficients for all predictors used in the Bayesian regression.

Table 22: The statistics of the Bayesian linear regression coefficients for the altered sparsity pattern for the period 2017-2019

Variable	10^{th} **Percentile**	**Median**	90^{th} **Percentile**	**St. Deviation**
Wind_Force	-0.9082813	-0.1015080	0.6952674	0.626076016
Temperature	2.548960	3.0277775	3.508571	0.374651503
Air_Pressure	1.201691	1.3824101	1.560702	0.140420001
Vacation_Dutch	-50.71006	-45.5009798	-40.30684	4.034383897
COS_158	35.96704	39.4684274	42.93454	2.715541655
TOT_168	0.9525116	0.9556068	0.9586999	0.002419613
σ^2	82675.02	83994.48	85339.19	1039.348

Table 23: The statistics of the Bayesian linear regression coefficients for the tuned sparsity pattern for the period 2017-2019

Variable	Median	10^{th} Percentile	90^{th} Percentile	St. Deviation
Wind_Force	-1.324	-2.088	-0.559	0.594
Temperature	2.449	1.974	2.925	0.371
Sun	15.962	10.148	21.793	4.538
Rain	-2.113	-9.940	5.730	6.103
Air_Pressure	0.887	0.708	1.066	0.140
Fog	13.091	3.772	22.417	7.277
Snow	-43.228	-63.509	-22.930	15.892
Thunder	8.942	-8.248	26.051	13.403
Vacation_Dutch	-41.333	-46.515	-36.144	4.052
Vacation_Belgian	-19.058	-24.323	-13.818	4.097
Vacation_German	-6.502	-11.105	-1.908	3.588
New_Year	-144.026	-173.839	-114.066	23.358
New_Years_Eve	-181.350	-210.791	-151.342	23.258
New_Year_After	-18.037	-47.958	12.378	23.475
Christmas_Vacation_After	139.239	108.245	170.006	24.043
Easter_Weekend	-89.910	-104.753	-75.272	11.534
Easter_Before	113.277	84.439	141.962	22.524
Ascension_Day_Weekend	-173.189	-188.867	-157.641	12.143
Ascension_Day_Before	220.254	190.856	250.261	23.093
Whit_Sunday_Weekend	-122.586	-140.284	-104.651	13.885
Christmas	-226.222	-245.926	-206.663	15.299
Christmas_Before	79.651	52.845	106.610	21.005
Christmas_After	-82.849	-97.632	-67.943	11.585
Sugar_Feast	34.932	5.796	63.917	22.683
Sugar_Feast_After	53.834	24.161	83.338	23.024
Kings_Day	-323.891	-352.954	-294.819	22.675
Kings_Day_After	84.104	50.223	117.933	26.521
Liberation_Day	110.470	76.834	144.255	26.248
Armistice_Day	79.658	51.150	108.882	22.483
Belgian_National_Day	105.801	76.797	134.758	22.665
Toll_Free	240.110	228.459	251.701	9.063
Maintenance	97.088	52.029	142.012	35.158
Incident	-19.329	-55.467	16.421	28.023
Sin_1	1.393	-3.500	6.312	3.846
Cos_1	28.647	14.931	42.263	10.670
Sin_2	77.473	45.446	109.040	24.815
Cos_8	281.376	92.103	470.510	147.895
Cos_9	-294.787	-487.233	-102.122	150.169
TOT_168	0.963	0.960	0.966	0.002
σ^2	73310.0	72133.7	74505.0	925.7

F Covid-19 Analysis

In this section, the results are displayed for the models for the data till 2020. For which we include covid-19 related variables.

F.1 Sparsity Pattern (including data of 2020)

In the table below, the coefficients are listed for the variables selected for the two sparsity patterns, including the covid-19 related data and variables.

Table 24: The predicting variables for 2017-2020 that are selected for the sparsity pattern in the altered quantile regression

Variable	Coefficient
Wind_Force	-1.042
Temperature	1.006
Air_Pressure	0.314
Infections	-0.037
Hospitalizations	-0.060
Cos_210	1.649
TOT_168	0.994

Table 25: The predicting variables for 2017-2020 that are selected for the sparsity pattern in the tuned quantile regression

Variable	Coefficient	Variable	Coefficient
Wind_Force	-1.109	*Christmas_After*	-13.105
Temperature	1.055	*Sugar_Feast*	-14.735
Sun	10.006	*Sugar_Feast_After*	-2.636
Rain	-1.891	*Kings_Day*	-193.354
Rainfall	-0.203	*Kings_Day_After*	-8.566
Air_Pressure	0.291	*Remembrance_ofthe_Death*	-3.715
Fog	-0.732	*Armistice_Day*	-5.234
Snow	-32.572	*Belgian_National_Day*	20.931
Thunder	0.942	*German_Unity_Day*	0.977
Vacation_Dutch	-5.471	*Toll_Free*	135.259
Vacation_Belgian	-2.419	*Maintenance*	-54.598
Vacation_German	-2.916	*Incident*	-24.907
New_Year	-34.248	*Infections*	-0.015
New_Years_Eve	-113.074	*Hospitalizations*	-0.709
New_Year_After	-9.965	*Deceased*	1.423
Christmas_Vacation_After	-12.962	*Lockdown_Intelligent*	-29.558
Easter_Weekend	-7.607	*Lockdown_Hard*	-53.551
Easter_Before	14.407	*Sin_1*	0.746
Ascension_Day_Weekend	-12.743	*Sin_2*	3.230
Ascension_Day_Before	27.353	*Cos_5*	0.769
Whit_Sunday_Weekend	-11.337	*Cos_6*	1.545
Christmas	-36.664	*TOT_168*	0.992
Christmas_Before	7.114		

F.2 LSTM Network (including data of 2020)

Here, we have the more detailed results for the LSTM networks of the data till 2020. As mentioned before, we use 50 LSTM networks per parameter-tuning step to compute the statistical properties.

Table 26: A statistical overview of the MASE and number of epochs per dropout rate for the period 2017-2020

	Mean		**St. Dev.**		**Max**		**Min**	
Dropout Rate	MASE	Epochs	MASE	Epochs	MASE	Epochs	MASE	Epochs
0	1.034	41.8	0.024	15.2	1.073	101	0.976	17
0.05	1.067	35.1	0.058	9.8	1.265	58	0.985	19
0.1	1.152	28.3	0.072	7.7	1.318	47	1.037	16
0.15	1.275	25.4	0.101	9.7	1.480	62	1.068	13
0.2	1.380	23.0	0.117	8.9	1.733	56	1.158	8
0.25	1.534	21.4	0.150	8.5	1.869	50	1.228	9
0.3	1.765	20.7	0.138	8.1	2.093	46	1.523	7

Table 27: A statistical overview of the MASE and number of epochs per batch size for the sparsity pattern from the altered quantile regression for the period 2017-2020

	Mean		**St. Dev.**		**Max**		**Min**	
Batch Size	MASE	Epochs	MASE	Epochs	MASE	Epochs	MASE	Epochs
32	1.054	33.7	0.031	13.4	1.138	63	1.007	8
64	1.034	41.8	0.024	15.2	1.073	101	0.976	17
128	1.013	55.6	0.034	15.4	1.101	97	0.947	33

F.3 Bayesian Linear Regression (including data of 2020)

Next, an overview of the parameters is listed for all predictors used in the Bayesian regression.

Table 28: The statistics of the Bayesian linear regression coefficients for the tuned sparsity pattern for the period 2017-2020

Variable	**Median**	10^{th} **Percentile**	90^{th} **Percentile**	**St. Deviation**
Wind_Force	-0.680	-1.307	-0.051	0.489
Temperature	2.159	1.756	2.561	0.315
Sun	13.471	8.498	18.486	3.896
Rain	-3.301	-11.474	4.819	6.361
Rainfall	-0.080	-3.970	3.856	3.057
Air_Pressure	0.679	0.528	0.831	0.118
Fog	11.677	3.731	19.612	6.163
Snow	-40.709	-59.246	-22.028	14.590
Thunder	4.616	-9.917	19.104	11.327
Vacation_Dutch	-38.250	-42.706	-33.846	3.465
Vacation_Belgian	-23.262	-27.779	-18.783	3.512
Vacation_German	-5.905	-9.877	-1.946	3.090
New_Year	-180.618	-206.119	-155.616	19.806
New_Years_Eve	-212.229	-237.523	-186.469	19.867
New_Year_After	-58.666	-84.045	-33.308	19.832
Christmas_Vacation_After	109.996	84.958	135.325	19.742
Easter_Weekend	-82.755	-95.311	-70.346	9.731
Easter_Before	101.167	76.514	125.516	19.089
Ascension_Day_Weekend	-165.114	-178.317	-152.001	10.240
Ascension_Day_Before	219.704	194.874	244.685	19.375
Whit_Sunday_Weekend	-117.333	-132.174	-102.259	11.676
Christmas	-217.177	-233.922	-200.633	12.924
Christmas_Before	72.867	49.524	96.005	18.191
Christmas_After	-68.344	-81.770	-54.927	10.453
Sugar_Feast	88.971	64.390	113.441	19.171
Sugar_Feast_After	64.293	39.624	89.103	19.370
Kings_Day	-239.467	-268.164	-211.167	22.293
Kings_Day_After	68.245	43.481	92.592	19.182
Remembrance_ofthe_Death	133.866	105.417	162.562	22.223
Armistice_Day	61.715	37.249	86.091	19.093
Belgian_National_Day	115.742	91.037	139.895	19.065
German_Unity_Day	92.934	68.487	117.692	19.156
Toll_Free	236.430	226.108	246.633	7.970
Maintenance	99.699	61.801	137.704	29.613
Incident	2.602	-26.843	31.928	22.894
Infections	-0.027	-0.067	0.013	0.031
Hospitalizations	-4.685	-5.872	-3.504	0.923
Deceased	0.827	-2.197	3.830	2.358
Lockdown_Intelligent	-50.985	-66.968	-34.885	12.513
Lockdown_Hard	-38.793	-71.668	-6.550	25.373
Sin_1	1.358	-2.859	5.538	3.265
Sin_2	12.537	6.172	18.866	4.953
Cos_5	-3.523	-43.899	36.396	31.268
Cos_6	35.314	-1.533	72.194	28.776
TOT_168	0.961	0.959	0.964	0.002
σ^2	69499.6	68530.0	70471.9	760.1

G Lagged Differences

Next, we see the coefficients for the sparsity pattern that was selected by the 50^{th} percentile quantile regression, where we included the lagged differences as described in Section 4.4

Table 29: The predicting variables for 2017-2019 that are selected for the sparsity pattern, where the lagged differences were included

Variable	Coefficient	Variable	Coefficient
Wind_Force	-0.366	*Christmas*	-2.275
Temperature	0.529	*Christmas_After*	-6.765
Sun	8.359	*Sugar_Feast*	-5.602
Rain	-0.941	*Sugar_Feast_After*	-4.280
Rainfall	0.206	*Kings_Day*	-27.908
Air_Pressure	0.176	*Liberation_Day*	2.673
Fog	0.552	*Kings_Day_After*	-5.032
Snow	-10.775	*Belgian_National_Day*	1.861
Thunder	-0.217	*Toll_Free*	47.377
Ice	0.284	*Cos_1*	-0.115
Vacation_Dutch	-2.442	*Sin_2*	2.120
Vacation_Belgian	0.608	*Cos_4*	0.714
Vacation_German	-1.224	*Cos_5*	0.793
New_Years_Eve	-22.559	*TOT_1*	0.635
Christmas_Vacation_After	-1.186	*TOT_2*	-0.042
Easter_Weekend	-4.535	*TOT_3*	-0.049
Ascension_Day_Weekend	-3.139	*TOT_4*	0.012
Ascension_Day_Before	2.280	*TOT_5*	0.026
Whit_Sunday_Weekend	-7.044	*TOT_168*	0.995

Table 30: The predicting variables for 2017-2020 that are selected for the sparsity pattern, where the lagged differences were included

Variable	Coefficient	Variable	Coefficient
Wind_Force	-0.369	*Sugar_Feast*	-2.601
Temperature	0.510	*Sugar_Feast_After*	-6.352
Sun	5.681	*Kings_Day*	-20.756
Rain	-0.909	*Remembrance_ofthe_Death*	-8.381
Rainfall	-1.097	*Armistice_Day*	–1.799
Air_Pressure	0.099	*Belgian_National_Day*	8.934
Fog	0.206	*German_Unity_Day*	-1.150
Snow	-10.805	*Toll_Free*	41.175
Vacation_Dutch	-1.967	*Infections*	-0.006
Vacation_Belgian	-0.595	*Hospitalizations*	-0.342
Vacation_German	-1.652	*Deceased*	0.821
New_Years_Eve	-21.040	*Lockdown_Intelligent*	-11.475
New_Year_After	-1.808	*Lockdown_Hard*	-23.008
Christmas_Vacation_After	-1.484	*Sin_2*	1.961
Easter_Weekend	-3.803	*Cos_4*	0.089
Ascension_Day_Weekend	-4.029	*Cos_5*	1.906
Ascension_Day_Before	15.223	*TOT_1*	0.651
Whit_Sunday_Weekend	-4.466	*TOT_2*	-0.041
Whit_Sunday_Before	3.464	*TOT_3*	-0.040
Christmas	-0.710	*TOT_4*	0.010
Christmas_Before	0.414	*TOT_5*	0.025
Christmas_After	-2.724	*TOT_168*	0.994

H Code

For this research we used four programs. First, we shaped and merged the data with `WSTDataAnalysis.py`. Next, we used `QRwithSquareRootLasso.R` to perform all quantile regressions and construct the sparsity patterns. After this, we used `LSTMNetwork.py` and `BayesianAnalysis.R` to respectively perform our LSTM networks and Bayesian linear regressions. Further details and explanations are provided in the codes themselves. It is important to mention that even though a seed is specified, results will still vary. This is due to the fact that not all functions of the code use this seed.