Comparison of ARIMA, SVR and ANN models including exogenous variables for short-term freight demand forecasting at a large national food distributor

Juliana Schneider

April 26, 2019

1 Introduction

While technological progress in the past decades has led to a significant decrease in relative emissions per tonne-kilometre in truck haulage, this achievement is compensated by an increase in freight transportation (59). In fact, absolute emissions of carbon dioxide rose by 20 % between 1995 and 2017, while freight transportation in general increased by 64% between 1991 and 2016 (60)
Between 2010 and 2030, according to (14), transport volumes for road haulage (in millions of tons) will face an estimated increase of 17% while its transport performance (in billion kilometres) will increase by about 39%. Thus, about 80% and 74% of predicted growth until 2030 for transport volumes and performance respectively will be due to road haulage.

Traditionally, freight transportation demand models are concerned with either vehicle or commodity movements, focusing on origin-destination matrices and geographical scope (e.g. urban, regional, international (46). Accordingly, the question of how many passengers and how much of different kinds of commodities is transported by which type of vehicle on which route is a task of interest for political institutions such as the German Federal Ministry of Transport and Infrastructure (Bundesministerium für Verkehr und Infrastruktur, BMVI) as it is essential for the planning of infrastructure based on transportation demand.(14)
Furthermore, in order to comp with the enormous impact of road haulage on climate, it is a major task for the government to improve and adapt infrastructures as well as strategic planning to transportation demands (13).

However, considering the growing demand for freight transportation, not only politics but also companies share an interest in the modeling of freight transportation demand (58).

What's more, according to (24), freight forecasting can help improve profitability as empty or imbalanced trips are reduced. Furthermore, by planning efficiently, the drivers' quality of life and work-life-balance is enhanced and employee turnover is reduced. This, of course, again reduces a company's expenses, since a high turnover rate imposes costs by the need to constantly hire and train new employees.
Another advantage of sophisticated freight demand forecasting is the enhancement of customer satisfaction through shorter delivery times, higher reliability and adapted pricing.

Unfortunately, little advancement has been made in the forecasting of short-term freight demand for truckage companies so far. This thesis now aims at building a bridge towards the objective of providing a useful, and applicable model to users.
By comparing three different kinds of methods - a rather classic time series approach and two machine learning models - combined with the incorporation of exogenous variables provided by public sources, first advice for future forecasts for this business case shall be given as well as a framework for further research.

After a brief description of the use case at hand, an overview of existing approaches, the theoretical background for variable selection and statistical methods used in this thesis is provided, followed by an introduction to the business case, the truckload company providing the data and a description of these data. Afterwards, the procedure of variable selection, model establishment and model testing is decribed in detail. Finally, the results achieved for this business case are reported and discussed.

2 Use Case: Short-term forecasting of freight weight for road haulage - on the example of NAGEL group

2.1 NAGEL group

2.2 The data

The data supplied by NAGEL and the University of Regensburg (a research partner of Fraunhofer's) for use in this coursework was supplied in .csv-format and included 472,137 observations between 17/09/2015 and 31/12/2016 for the following 30 variables: (TABELLE SPALTENBESCHREI-

BUNG)

2.3 External data

hier nur kurz beschreiben, dass es externe Daten geben wird.

3 State of th Art in the Modeling of Freight Transportation Demand

NOCH USE OF EXTERNAL DATA BESCHREIBEN

First of all, while browsing literature on the topic of freight transport demand, the sheer amount of different aspects to take into account and consequently the various different quantities under observation in different papers was striking. Common measures in freight demand modeling are: tonne-kilometres, mode choice, freight volume, origin-destination-matrices, and so on.

As the goal of this thesis is to establish a model to predict short-term freight demand for a single user, comparability and therefore scalability of the target variable is not necessary. Hence, the target variable "weight" included in the data provided by NAGEL is not altered, but of course, in further works it might be of use to modify the target variable, e.g. to value/weight-ratios.

As initially mentioned, freight transport modeling has so far been concerned with the so-called four-step transport models consisting of trip generation (number of trips from or to origin), trip distribution (destination of trips from origin), mode choice, and trip assignment (choice of path or route between origin and desitnation), adopted from traditional passenger transport modeling(19; 57).

However, as there are differences in freight demand and passenger demand modelling, adaptations have been proposed by (19). Among other things, the first step in the four-step model - trip generation - should not count the number of trips starting from a certain region of origin, but count the amount of freight (in tonnes) delivered from this origin.

In order to construct an entire freight demand model system, the separate measures of all four steps need to be connected, e.g. through value/weight ratios.

However, the objective of this thesis is not to establish an entire freight model system but it is rather concerned with questions that would belong to step one.

Despite the adaptations proposed by (19), not many other advances have been made. In fact, while passenger transport is a largely studied field due to government regulations, this does not hold true for freight demand (46).

(19) further states that within step one, models so far have only used aggregate data, where time series analysis models, with and without exogenous variables, have already been applied to short-term forecasting.
(46), besides distinguishing between aggregate and disaggregate models, classify into international, intercity and urban freight transportation. Here, at first glance, the category of an intercity model seems suitable, but in this category, freight transportation demand is either modeled as a utility maximisation task or an inventory based model for mode choice and production decision, which again is not the scope of this work.

Meanwhile, since, as mentioned above, most literature so far is concerned with aggregate data, within the use case at hand the focus is on the analysis of disaggregate data. Apart from this, further parts of the four step model are not of interest, i.e. neither other parts of step one (e.g. I/O-models) nor other steps, which is why the research on existing literature was expanded to include more specific studies.

(55) used artificial neural networks to predict the demand for weather-sensitive retail products.
(33) predicted container throughput using Dynamic Factor Anaysis and ARIMAX models.
(37) established hybrid ARIMA and ANN models based on DWT decomposition for general time series analysis. (41) used ARIMA to predict full truckload transportation prices.
(42) applied SARIMA models to forecast the demand in a beverage supply chain.
(25) combined grey models, ANN and Support Vector Machines to model full truckload volume. (6) use both univariate and multivariate ARIMA models, examining the impact of new links, new destinations and lower fares on air transport demand at the Port of Reggio Calabria. (35) build a hybrid of ARIMA and SVR for general long term time series prediction. (38) suggest Ensemble Neural Network models to yield more accurate results in time series forecasting.

As is evident, all of the papers mentioned above applied models to either measures similar to but not exactly weight or time series analysis in general. Several authors proposed hybrid models consisting of combined versions of the basic models to be compared further below.
Regardless of the lack of literature exactly matching the target of this thesis, the combined information gained from all of these papers provides reasonable background to be supportive of it. Furthermore, this scarcity in literature proves the necessity of a primary approach to adress the objective of the use case at hand.
Hence, it is important to first retrace the reasoning behind the choice of rather basic versions of AR(I)MAX, SVR and ANN models for the prediction of freight weight demand, which the following section is dedicated to.

4 Model Choice and Specification

There are several ways to model time series data, some of which have been well established for many decades already, while others have recently emerged in the space of popular methods as well . ARIMA models are an example of the former kind of methods, while SVM and ANN are examples of the latter kind. (4)
(35) state ARIMA, SVM and ANN as the three major methods used in time series analysis.

ANNs are particularly useful as a data-driven method, if there are data but difficulties in describing the process behind them (63). In the case at hand, as there ist not much literature exactly fitting the purpose of my research question, ANNs might be a convenient approach to handle the data.

All three of the aforementioned methods may prove suitable to forecast freight weight, as they each have advantages (see table NUMMER). Furthermore, it might prove insightful to contrast a classical statistical, a machine learning and a deep learning method with each other.

4.1 Model specification

As we shall see, for model specification - i.e. the modelling of the parameters - *Information Criteria* are commonly used. For this reason, the most prevalent of them, the *Akaike Information Criterion* (5) and the *Bayesian Information Criterion* (or *Schwarz Information Criterion*, (52)) are shortly introduced in this section before moving on to descriptions of the specific types models.

$$AIC = 2k - 2ln(\hat{L}) \tag{1}$$

$$BIC = ln(n)k - 2ln(\hat{L}) \tag{2}$$

$\hat{L}$ denotes the Log-Likelihood of the model, k the number of parameters and n the sample size. The AIC's aim is to choose the model that approximates the true data generating process as closely as possible (62). Generally, the model yielding the smallest AIC or BIC respectively is chosen (27). Neither one of the two criteria is superior to the other; AIC tends to overfit the larger the sample size, whereas BIC tends to choose overly simple models in finite sample sizes (27).
Both of them have been used in the three models presented in this coursework. In the following sections, each model's theoretical framework and their associated specification techniques shall be illuminated in more detail. First,

time series models are introduced as the constitutive baseline of time series modelling. Then, SVR and NNs are presented.

5 Time series

Forecasting of time series with (S)AR(I)MA models is a well-established concept that has been studied thoroughly for many decades and provides good forecasting accuracy (8; 37).It has found application in many domains such as economy (MEHR QUELLEN),

The assumption ARIMA models are based on states that the values of a target variable are generated by a linear combination of past values of the same variable and white noise (37), thus making it a stochastic process, "i.e. an ordered sequence of random variables" (6), with data entries at equally distant intervals (30).

A mathematical assumption underlying times series processes is stationarity. (Weak) Stationarity is given when mean and covariance are independent oft time *t*, and the relationship between two values at time points t and $t + i$ is the same as the relationship between two values at time points s and $s + i$, i.e. independent of the exact position in the time series, but provided the distance between any two values is the same (61).

5.1 AR models

Autoregressive (AR) processes are processes where a value of a variable at t depends on weighted previous values of the variable itself plus a white noise term $e \sim WN(0, \sigma_\epsilon^2)$. An *AR(p)* process of order p has the form:

$$Y_t = C + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + e_t \tag{3}$$

where C is the intercept.

The order p of an AR process may be determined by a visual check of the plotted Partial Autocorrelation function (PACF). The PACF and the corresponding empirical PACF are denoted in the following form: Considering a time series as a regression where τ denotes the lag:

$$Y_{t+1} = \Phi_{\tau,1} Y_{t+\tau-1} + \Phi_{\tau,2} Y_{t+\tau-2} + ... + \Phi_{\tau,\tau-1} Y_{t+1} + \Phi_{\tau,\tau} Y_t + \epsilon_{t+\tau} \text{for } \tau = 1, 2, ... \tag{4}$$

The PACF is now defined as the regression coefficient $\Phi_{\tau,\tau}$.

$$PACF : \pi(\tau) := \begin{cases} \Phi_{\tau,\tau} & \tau = 1, 2, ... \\ 1 & \tau = 0 \\ \pi(-\tau) & \tau = -1, -2, ... \end{cases} \tag{5}$$

with $-1 < \pi(\tau) < 1$ (61). The empirical PACF can be determined by using Yule-Walker equations or the Durbin-Levinson algorithm (for a detailed description see (61)).

5.2 MA models

Moving Average (MA) processes of order q are denoted like this:

$$Y_t = \mu_{MA} + \epsilon_t - \phi_1 \epsilon_{t-1} - \phi_2 \epsilon_{t-2} - ... - +\phi_q \epsilon_{t-q} \tag{6}$$

This means the value of a target variable at t depends on a white noise process and previous white noise weighted by ϕ. ϵ in this context are called *innovations*. Furthermore, μ may be 0 and MA-processes are stationary and causal (61). The order q of an MA process may, equal to AR processes, be determined by a visual check of the Autocorrelation Function (ACF)'s plot (correlogram). The ACF and the empirical ACF are denoted as:

$$ACF : \rho(\tau) := \frac{\gamma(\tau)}{\gamma(0)} = \frac{Cov(Y_t, Y_{t+\tau})}{Var(Y_t)}, \tau \tag{7}$$

$$empirical\ ACF : \hat{\rho(}\tau) := \frac{\hat{\gamma}(\tau)}{\hat{\gamma}(0)} = \frac{\sum_{t=1}^{n-\tau}(y_t - \bar{y})(y_{t+\tau} - \bar{y})}{\sum_{t=1}^{n}(y_t - \bar{y})^2} \tag{8}$$

A condition introduced by (12) is the **invertibility** of the MA process into an AR(∞) process, which ensures that only one parametrization of an MA process can be identified for any ACF.

However, the process underlying time series data may change over time - it is subject to uncertainty (4). A time series model may be biased or overfitted as well as its parameters misspecified.

5.3 AR(I)MA models

As the name suggests, AR(I)MA(p,d,q) - auto-regressive integrated moving average - models model time series data with an AR and an MA component, and trend in data through differencing (which is the "I-part").
The parameters p, d, and q respectively denote the order of the AR component, the degree of differencing and the MA component (64). If there is also a seasonal component - thus a SARIMA model is to be fitted - there are additional parameters P, D, Q referring to the seasonal orders (or degrees) of AR, differencing and MA.

$$Y_t = \mu_{ARMA} + \sum_{i=1}^{p} \Phi_i Y_{t-i} + e_t + \epsilon_t - \sum_{m=1}^{q} \Theta_m \epsilon_{t-m} \tag{9}$$

where μ_{ARMA} actually equals the intercept C of the aforementioned $AR(p)$ model.

5.4 Seasonal AR(I)MA - SARIMA models

Of course, a time series may also exhibit recurrent seasonal patterns, e.g. an increase in demand in december every year. If there are recurrent fluctuations that ocurr yearly or otherwise periodically, there is a seasonal or cyclical component, respectively (61) that has to be taken into account; the time series has to be "deseasonalized" accordingly (11).

Seasonal components can be detected by visual checks of the time series plots. Seasonality is modeled in AR(I)MA models by an additional tuple of orders (P, D, Q); the complete model is then denoted as $SARIMA(p, d, q)(P, D, Q)$, where the additional tuple of orders essentially has the same meaning as the former one, only applied to seasonal magnitude.
The length of the period of a cyclical component may furthermore be determined through a fourier series (61). (32) also recommends modeling larger seasonal components (such as yearly reocurrences in daily measured data) or multiple seasonalities by a fourier series. A time series y_t can be expressed as a set of orthogonal trigonometric functions - actually as a linear combination of cosine and sine functions - reduced to only the terms up to a certain lag k, so fitting it is less tedious (especially for large sample sizes) (**?**).

5.5 Violation of assumptions and other issues

If $\mu(t) \neq \mu$, i.e. the mean is not independent of time, the assumption of stationarity is violated; there is a so-called trend. In the case of a non-stationary time series, stationarity can be obtained by differencing (6; 30) First-order differencing is denoted as:

$$BY_t = Y_t - Y_{t-1} \tag{10}$$

, where B is the backshift operator (30). If $\sigma^2(t) \neq \sigma$, i.e. the variance is not independent of time, the time series can be logarithmised to obtain stationarity (6). Tests for stationarity include (Augmented) Dickey-Fuller-Test (ADF test) (64) and a visual check of the time series plot (see Box-Jenkins program further below).

ADF tests are part of the family of unit roots tests, i.e. the null hypothesis states that the process is non-stationary (64). The order k that denotes the order of the lag is suggested to be set to $k \approx \sqrt[3]{n-1}$ and reduced successively by t-testing the last parameter yo of equation NUMMER 4 - as long as it does not differ significantly from $zero$, k should be further reduced. See (61) for a detailed description of the test metric.

Another condition of time series processes is **causality**, i.e. future values only depend on current and past values and not on future values themselves (61).

As with any kind of data, missing values may pose a problem for inference and forecast. While in real applications, the mechanism causing the missingness is unknown, at least for time series, the imputation of values for *Missing At Random (MAR)* and *Missing Completely At Random (MCAR)* mechanisms is nearly identical. [1] Imputation for time series does not solely depend on covariates, but instead on the "hidden variable" time. (43) found that linear interpolation yielded

5.6 The Box-Jenkins program

In order to iteratively model time series data, (12) proposed a method to identify suitable parameters - AR, MA and differencing - of an ARIMA model. It consists of the following four steps, as described in (22):

5.6.1 Order selection

The orders p and q of the AR and MA have to be determined. This can be done via a visual identification through ACF for MA- and PACF for AR-orders, as mentioned above.
(61) provides an overview for signs of AR, MA and ARMA processes in ACF and PACF plots:

AR(p) processes:
- ACF fades with increasing lag τ, possibly sinusoidally or alternatingly.
- PACF breaks off with lag $\tau > p$.

MA(q) processes:
- ACF breaks off with lag $\tau > q$.
- PACF fades with increasing lag τ, possibly sinusoidally or alternatingly.

ARMA(p,q) processes:
- Both ACF and PACF fade with increasing lag τ, possibly sinusoidally or alternatingly.

(65) recommend settling on an order of both p and q in a magnitude between 0 and 10, while d's most sensible magnitude is between 0 and 2. For ARMA models, simple plot checks are insufficient and commonly, candidates for p and q are chosen by minimizing an Information Criterion such

[1] In MCAR, missingness in the data is completely random, whereas in MAR, missingness in one variable depends on the value of another variable. Lastly, in MNAR, missingness in one variable depends on the value of the variable itself.

as the AIC or BIC (22):

$$AIC(p,q) := log(\hat{\sigma}^2_{p,q}) + 2\frac{p+q}{n} \tag{11}$$

$$BIC(p,q) := log(\hat{\sigma}^2_{p,q}) + \frac{(p+q)log(n)}{n} \tag{12}$$

$$\tag{13}$$

(61) check ACF and PACF plots after regular and seasonal differencing to determine der P and D orders in a seasonal ARIMA model. With seasonal ARIMA however, checking ACF and PACF for the regular order p and q is not so easy; it is recommended to initialize the model with high orders for both and then reduce them while checking residuals.

5.6.2 Estimation of parameters

Next, the parameters (or coefficients) of the AR and/or MA components of the model are estimated, e.g. by Ordinary Least Squares (OLS) or Maximum Likelihood Estimation (MLE) (6).
To estimate the parameters of an ARMA process, MLE is the most common method (61). However, it assumes a distribution for Y_t - usually a normal distribution.
(61) prefers to switch steps 1 and 2, reasoning that the estimation of the parameters depends largely on the order of the ARMA model. However, admittedly to estimate the parameters, the order of the process have to be known beforehand; although they can be re-adapted later on. For this reason, in this coursework, the procedure according to (12) will be followed.

5.6.3 Model diagnostics

These are mainly residual checks (8). After having fitted the ARIMA-moddels, it is necessary to test the residuals for autocorrelation and normal distribution. Using the Ljung-Box-Test (LB), it is possible to check whether the residuals are autocorrelated, and with the Jarque-Bera-Test (JB), one can check for deviation of the time series from a normal distribution (6). The null hypothesis in the LB-test is that the time series process Y_t consists of independent and identically distributed (*i.i.d.*) random variablest, whereas the null hypothesis in the JB-test states that the random variables Y_t are normally distributed. If the conditions do not hold, the model has to be re-specified as in step 1 (61).

$$LB = N * (N+2) * \sum_{\tau=1}^{m} \frac{(\hat{\rho}(\tau))^2}{N-\tau} \tag{14}$$

$$JB = \frac{N - n_p}{6} * (S^2 + \frac{(K-3)^2}{4}) \tag{15}$$

N is the sample size, τ the order of the lag, m the number of lags considered, $\rho(\tau)$ the autocorrelation function at lag τ, n_p the number of parameters, S the skewness and K the curtosis of the time series. If both tests are statistically significant, then the assumption holds that the error term is a white noise process $e \sim WN(0, \sigma_\epsilon^2)$ (33).

5.6.4 Forecasting

The objective in forecasting is to reduce the expected deviation of the estimated outcome $\hat{Y}_t$ from the actual outcome Y_t of a time series (22):

$$E(Y_t - \hat{Y}_t) \tag{16}$$

To forecast an *AR(p)* model, for $t = 1$, i.e. the first point in time ahead that we have no observed values for, one can simply replace the parameters in equation NUMBER by the parameters obtained in step 2 and insert past values for Y_{t-h}, h = 1, 2, ... For an $AR(2)$ process, the one step ahead forecast looks like this:

$$\hat{Y}_{t+1} = C + \Phi_1 Y_t + \Phi_2 Y_{t-1} \tag{17}$$

In *MA(q)* models, reconsider that the innovations are caused by white noise, and for a forecast of Y_{t+1}, ϵ_{t+1} is directly part of the equation whose expected value is 0. ϵ_{t-1}, so the one step ahead forecast for an $MA(2)$ process looks like this:

$$\hat{Y}_{t+1} = \mu_{MA} - \Theta_1 \epsilon_t - \Theta_2 \epsilon_{t-1} \tag{18}$$

But already with a two step ahead forecast, it is obvious that the $MA(2)$ process (as all $MA(q)$ processes) converges towards the mean μ the farther ahead the forecasting step:

$$\hat{Y}_{t+2} = \mu_{MA} - \Theta_1 \hat{\epsilon}_{t+1} - \Theta_2 \epsilon_t = \mu_{MA} - \Theta_1 * 0 - \Theta_2 \epsilon_t \tag{19}$$

because $E(\epsilon_{t+1}) = 0$. In *ARMA(p.q)* models, the farther ahead the forecast, the more $Y_{t+\tau}$ converges towards the expected value of the time series $E(Y_{t+\tau}) = \mu_{ARMA}\ for\ \tau = 0, 1, ...$

5.7 AR(I)MAX

So far, only univariate time series processes, where Y_t is solely predicted by its own past values, $Y_{t-\tau}, \tau =$ 1,2,..., have been considered. Naturally, the question arises whether it is possible to improve predictions based on further explanatory variables (denoted by X or X_t in the following).

Unfortunately, literature on AR(I)MAX models is scarce compared to literature on classic univariate time series analysis.
(33) use ARIMAX to forecast container throughput with additional information of macro-economic indicators at the Port of Koper, whereas (21) forecast macroeconomic time series themselves with both ARIMA and ARIMAX models.
(39) include Ramadan effect in their prediction of sales data.
(7) observed differences in the sales of muslim kids' clothing with Eid holidays every year and modeled this with an ARIMAX model.
(18) compare ARIMAX to SARIMAX modeling in daily traffic counts.

Common AR(I)MA models can be extended to so-called ARMAX models by adding lagged explanatory variables to the model (6):

$$Y_t = \sum_{i=1}^{p} \Phi_i Y_{t-i} + et + \mu_{ARMA} + \epsilon_t - \sum_{m=1}^{q} \Theta_m \epsilon_{t-m} + \gamma X_{t-1} \qquad (20)$$

Equation NUMMER can be rewritten in terms of a transfer function model (21):

$$y_t = C + v(B)X_t + e_t \qquad (21)$$

where $v(B)X_t$ is the transfer function with backshift operator B:

$$v(B)X_t = \sum_{j=0}^{\infty} v_j B^j X_t \qquad (22)$$

AR(I)MAX models can be interpreted as regression problems, where one variable is explained by another so that it can be considered as a linear regression with autocorrelated error terms (18).

6 Support Vector Machines: Support Vector Regression

Support Vector Machines were first developed by (QUELLE) in the 1990s for classification problems. Back then, they became popular quickly because of their high performance and impressive new strategy to solve higher-dimensional classification problems (34). In their simplest application - a linearly separable binary classification - given a training set X, the SVM puts a line (or for multiple classes, a hyperplane) between the observations of the two classes that has a maximum distance to each of them (**?**). Then, given a test set, it can classify the observations into either the one or the other class, using this separating line.

The separating line is chosen to have a maximal distance to the x-vectors (i.e. the observations) of both classes while neatly separating them; it is called maximal margin. Note that it depends solely on its support vectors (these are the vectors directly on the margin or outside of it); observations at a farther distance than the ones closest to the line are not taken into account (34). A problem arises when the classes are not exactly separable. Then, a so-called soft-margin is chosen that separates the classes as well as possible with as few as possible observations being wrongly classified in the training set (and subsequently the test set). Another problem of maximal margin classifiers is that they are very sensitive to the training data initially provided to create the separating hyperplane, as it uses the support vectors to compute the maximum distance for the hyperplane and discards all other observations. Now, if a new observation is added in the test set that would have shifted the maximal margin had it been provided in the training set, it will be missclassified. Again, a soft margin classifier might solve this problem by generally allowing some (but as few as possible) observations to be missclassified - even in training. It allows for observations to be within the boundaries of the margin, or even on the wrong side of the hyperplane. Formally, the classifier is denoted as this:

$$\underset{\beta_0,\beta_[1],\ldots,\beta_p,\epsilon_o,\epsilon_1,\ldots\epsilon_n}{maximize} \quad M \tag{23}$$

$$\text{subject to} \sum_{j=1}^{p} \beta_j^2 = 1, \tag{24}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i), \tag{25}$$

$$\epsilon_i \geq 0; \sum_{i=1}^{n} \epsilon_i \leq C, \tag{26}$$

with M as the margin's width, the ϵs *slack variables* to allow observations to be on the wrong side of the margin or the hyperplane (34) with $\epsilon_i = 0$ indicating that it is on the correct side; $\epsilon_i > 0$ indicating that it is on the wrong side of the margin and $\epsilon_i > 1$ indicating that it is on the wrong side of the hyperplane; C as a nonnegative tuning parameter indicating the total number of ϵs and therefore wrongly classified observations; the rest being a "classic" linear function of x on y.

6.1 The tuning parameter C

The tuning parameter C indicates the tolerance for violations of the side of the margin on which an observation lies. Here $C = 0$ equals a maximal margin where no observations past within the margin are allowed (34). A

smaller C indicates a narrower margin with a higher risk of overfitting to the training data, whereas a larger C might result in a biased classifier. Moreover, the larger C, the more observations are tolerated on the wrong side of the margin and the more support vectors there are.
Generally, cross validation is used to determine C.

6.2 Kernel functions

In non-linear problems, a separating linear hyperplane is constructed by mapping the input to a higher-dimensional space (4). The non-linear mapping of the hyperplane into a higher-dimensional space can be conducted by using a *kernel function.*
According to (34), the support vector classifier can be represented using only the inner products of the observations, and these suffice also to estimate its parameters α_i:

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i \langle x, x_i \rangle \tag{27}$$

and α_i is zero if a training observation is not a support vector. Now, for non-linear problems, the inner product $\langle x, x_i \rangle$ is replaced by a more generalized term, a kernel:

$$K(x_i, x_{i'}) \tag{28}$$

Common kernels are:

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} (x_{ij}, x_{i'j}) \qquad \text{linear kernel} \tag{29}$$

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} (x_{ij}, x_{i'j}))^d \qquad \text{polynomial kernel} \tag{30}$$

$$K(x_i, x_{i'}) = exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2) \qquad \text{radial kernel} \tag{31}$$

Kernels provide advantages such as savings in terms of computation time, as they only require the computation of the kernel function for all distinct pairs of observations and do not need to make use of the enlarged feature space (which can be enormously large).
Most authors such as (4) and (**?**) choose the Radial Basis Function as kernel.

6.3 Parameter estimation in Support Vector Machines

(**?**) propose choosing first which kernel to use before finding the best parameters C and γ by using cross-validation. They also suggest scaling of the data in order to avoid having larger scaled variables dominate variables of smaller scale. The choice of kernel depends on which ones have already been used in related literature (as mentioned above: radial basis function), but also on recommendations by other authors. Again (**?**) provide a hint by suggesting the radial basis function for a start, as it can map non-linear relationships and needs less hyperparameters than other kernels, which renders model selection simpler. Furthermore, it is a useful kernel when the number of features is not too large, which is assumed to be true in this case. Moving on to finding C and γ by cross validation next, a "simple" grid-search is suggested to be favoured over approximate search algorithms, because of two reasons: one being the psychological phenomenon of feeling safer when having tested all possibilites, and the second being that the grid-searches for C and γ can be parallelized since they are independent from each other.

6.4 Extension to regression problems

So far, Support Vector Machines were introduced for binary classification problems. Of course, for the data at hand, a method for regression is required instead. (QUELLE) presented an extension to Support Vector Machines: Support Vector Regression, where the coefficients $\beta_0, \beta_1, ..., \beta_p$ are sought to be minimized by a loss function that only takes into account residuals larger than some positive constant ε; any error smaller than that are accepted (**?**). In many cases, this constraint might not possibly be fulfilled and there might not be a solution where there is a set of coefficients to ensure that the error is smaller than ε. For this reason, so-called *slack variables* ξ_i, ξ_i^* were introduced. This results in the following optimization problem:

$$\text{minimize}\frac{1}{2}||\beta||^2 + C\sum_{i=1}^{l} \xi_i + \xi_i^* \tag{32}$$

$$\text{subject to} \begin{cases} y_[i] - \langle \beta, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle \beta, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases}$$

where b is the intercept. Recall that C is a constant describing the trade-off between a minimizing β and tolerating a larger amount of errors to be larger than ε (**?**).

The so-called *ε-insensitive loss function* $|\xi|_\varepsilon$ in its dual form (using Langrange multipliers and a dual set of variables) possesses a saddle point at the solution for the primal and dual variables and can be solved relatively easily when denoted in this form. The derivation of the *Support Vector expansion* off of the dual formulation using Lagrange Multipliers is not going to be explained at this point; however its result is the following:

$$\beta = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) x_i \tag{33}$$

with α_i being one of the two dual variables. Thus, the coefficients are made up by a linear combination of the training patterns.
Support Vector Regression uses different cost functions than Support Vector Machines (34), e.g. $c(x, y, f(x)) = |y - f(x)|$ or $c(x, y, f(x)) = (y - f(x))^2$.

7 Artificial Neural Networs

Although seemingly new in the world of statistical tools, Artificial Neural Networks have been invented many decades ago and the enthusiasm for them HAS experienced several revivals throughout the years (25). The idea stems from human brains (63) and the way their neurons learn, store and process information together (9). Although this sounds like a very complex, ambitious venture, NNs are "simply a parameterized non-linear function that can be fitted to data for prediction purposes" (44). In principle, an n-dimensional input is mapped onto an m-dimensional output (47). The process, although very much data-driven (63) and sensitive to the training data (4), is not non-parametric (20).
Nowadays, with the advance of fast and high-performance personal computers, basically anyone can fit a neural network to their data (25).
A great strength of NNs is their ability to detect structures between variables that have not yet been discovered (63).
During these different stages of popularity, ANNs have been developed further. Refined architectures allow the researcher to adapt a neural network to their problem's structure, starting from rather simple one-layer feedforward algorithms to more complex long-short-term-memory multi-layer-perceptron NNs. (QUELLE)
Here, a short overview shall be given before discussing the choice of type of NN for this specific case.

7.1 The general structure of NNs

An NN typically consists of (at least) three layers: One input, one hidden and one output layer. ANNs learn by example (9; 25): Neurons in the input-layer receive the input from the data and pass them on to the hidden layer, where information is processed and then passed to the output layer, which usually consists of one single neuron and is equivalent to the target value.

(QUELLE)

Layers - and neurons - are connected by activation, also called transfer functions, which multiply the output received by the preceding neuron by a weight and pass this new output on to the next layer and so on.
Lastly, a learning algorithm is applied to estimate the parameters in the network. When fed some training data, the network's output is compared to the actual output value of the target variable through a loss function. This error is then passed backwards through the layers to adjust the transfer functions' weights.

Multilayer perceptrons are the most popular type of (feedforward) NN (9; 11).

The most common learning algorithm is backpropagation as proposed by (48).

7.2 Network size and structure

A crucial step in modelling NNs is to define the network size and structure, i.e. define the number of layers and nodes (neurons), in order to accurately depict the complexity of the structure of the data and avoid overfitting or underfitting as well as keep computation times low (36).
There is no exact way to determine the number of nodes in each layer, although a lot of previous work has been dedicated to this problem (36). Of course, it depends on the problem at hand.
The number of input neurons may be equal to the number of input variables in one's problem, but in a time series, the number of input neurons is not evident (63). However, obviously for each exogenous variable added there has to be (at least) one input node added to the NN.
It is already clear at this point that the number of output neurons is one, as we are interested in one continuous target variable and as is recommended by (36).

However, the number of hidden layers and the number of neurons within each hidden layer still has to be determined.
As the objective of this thesis is to present basic models to be contrasted with each other, one hidden layer should suffice. Now, the number of hidden neurons has to be established by trial and error - since there is no clear rule - as is done by most researchers (36). If, with an arbitrary small number of hidden neurons, the network should fail to converge even after multiple restarts, the number of hidden neurons should be increased successively.
To determine a suitable number of hidden nodes, (4) use the BIC:

$$BIC_{p,h} := h(p+2)+1 \tag{34}$$

Due to their ability to approximate any (linear or nonlinear) function

underlying the data arbitrarily closely by a nonlinear function, MLPs and RBFNNs classify as universal function approximators (20). They can reduce to an AR[p]-model with an appropriate function that is estimated with a learning algorithm.
Considering the vast number of different types of NNs as well as their general flexibility, it is not a trivial task to choose a suitable model, specify it, and train it (38).

In time series analysis with NNs, Y_{t+1} is the output that is calculated by feeding the network lagged past obervations of $Y_t,\ t for 1, 2, ...T-1$ and possibly lagged values of exogenous variables (38). For this purpose, the time series is divided into equally sized chunks and fed to the network, where the the next value in the chunk is expected to be determined by the preceding elements of the same chunk (**?**).

(20) counts feedforward ANNs, Jordan ANNs, Elman ANNs (the latter two being types of RNNs), and Multicurrent Networks as most relevant for time series analysis. Feed-forward NNs an Recurrent NNs shall be described in more detailed in the following subsections.

7.3 Activation functions

Activation functions - also known as transfer functions - introduce nonlinearity by mapping the input node to the output node(63). They must be differentiable (47).
Generally, the activation a of a neuron n_i is given by

$$a_i = f(n_i) = f(\sum w_{ij} * x_j) \tag{35}$$

where $f(n_i)$ is the transfer function, w_{ij} is the connection weight between node j and node i, x_j the input signal from the node j, and a_i is the output of the neuron i (11). Among the most popular are sigmoid, hyperbolic tangent, sine or cosine, and linear functions.

$$f(x) = (1 + exp(-x)) - 1 \tag{36}$$

sigmoid or logistic. Sigmoid functions can take values strictly between 0 and 1 and their derivatives are always positive (47).

$$f(x) = (exp(x) - exp(-x))/(exp(x)) + exp(-x)) \tag{37}$$

Hyperbolic tangens (tanh) function can take values between -1 and 1.

$$f(x) = sin(x) \tag{38}$$

or

$$f(x) = cos(x) \tag{39}$$

sine / cosine can take values between -1 and 1.

$$f(x) = x \tag{40}$$

linear can take values between $-\inf$ and $\inf$.

(63) point out that for target variables with continuous values, the use of a linear activation function in the output node is advisable. It must be noted that HIER SEKUNDÄRQUELLE COTTRELL ET AL IN ZHANG 1998 linear activation functions may not be applied on data with trends; hence differencing may be necessary before feeding the data to the network.

Since most papers reviewed for this thesis mention the use of sigmoid or logistic transfer functions in the hidden nodes and linear activation in the output nodes, it seems a reliable modus operandi.

7.4 Learning algorithms

NNs may be trained in various ways. Generally, learning algorithms - also called training algorithms - iteratively minimize an error function to adjust the weights on the nodes (44). Backpropagation as the most widely used algorithm is based on the gradient-descent method (47), which is why differentiability of the activation functions and therefore of the error functions as well is necessary. After a random initialization of the network, the initial weights are adjusted according to the gradient of the error function. The Sum of Squared Errors (SSE) might be considered (47):

$$E = \frac{1}{2}\sum_{i=1}^{p} ||\hat{y}_i - y_i||^2 \tag{41}$$

Another type of error function next to SSE may be Mean Squared Error (MSE) (63), which is defined by SSE/n. Because the training set size is commonly much larger than the test set's, the training MSE will usually underestimate the test MSE (34).
The derivatives of E,

$$\nabla E(w) = \frac{\partial E(w)}{\partial w_1}\frac{\partial E(w)}{\partial w_2}, ..., \frac{\partial E(w)}{\partial w_l} \tag{42}$$

are minimized and the weights iteratively updated:

$$w_i = -\gamma\frac{\partial E(w)}{\partial w_i} for i = 1, ..., l \tag{43}$$

, where γ is the step size, also called learning rate. It determines the extent to which the weights are changed in each iteration and must carefully be specified, since too steep a descent might not converge and a too shallow one

might be too slow (63).
In the end, ∇E is expected to be zero. When given input data, the network is fed forward by calculating the input to each node,

$$x_j = \sum_i y_i * w_{ji} \tag{44}$$

, where j indexes output nodes, i indexes the nodes connected to node j, w_{ji} denotes the weight and y_i denotes the antecedent's node's output; outputs in turn are calculated by their activation functions. Furthermore, nodes can be given bias by adding an input with constant weight of 1(48).

Gradient-based learning algorithms can either be conducted in batches or on-line; in the former, all input-output patterns are processed parallely so that the weights are adjusted simultaneously; in the latter, the weights are updated sequentially (47).

The possible flaws of static learning rates have been tried to overcome by expanding the classic gradient descent method with a so-called momentum parameter that controls the direction of the change in weights to not diverge too far from the direction of the antecedent change in weight. (QUELLE)
Backpropagation uses the first derivative of the error function is minimized; other algorithms use the Hessian matrix and thus the second derivative (47):

$$h = -(\nabla^2 E(w))^{-1} \nabla E(w) \tag{45}$$

where w_i is then updated in the following way:

$$w_i = w_{i-1} + h \tag{46}$$

As it is an iterative algorithm and the Hessian matrix has to be computed in each iteration, Quasi-Newton methods make use of a simplification where only the diagonal elements of the Hessian matrix are used:

$$w_i = w_{i-1} + \frac{\nabla_i E(w)}{\partial^2 E(w)/\partial w_i^2} \tag{47}$$

This method is based on the assumption of a quadratic error term. (45) use the Levenberg-Marquardt algorithm, a quasi-Newtonian learning algorithm, to improve learning speed and because it works especially well when there are only few observations; so do (11), (38) and (65). (63) report "faster convergence, robustness and the ability to find good local minima" as key advantages of second-order methods.

As already mentioned, NNs are very sensitive to the data they are trained on; thus, overfitting might occur: the model fits the training data very accurately but performs poorly in predicting new (test) data (44). In order to avoid this issue, so-called early stopping of the training algorithm can be applied, i.e.training is stopped as soon as the training error term (e.g. MSE) starts increasing on a pre-selected validation data set. The flip side of the coin is the problem of NNs being stuck on local minima - independently of the choice of learning algorithm (63) - and therefore being biased; a method of circumvention is the running of multiple epochs, where all steps of training are passed with random initializations of parameters for each epoch. It is avisable to employ a *burn-in*

A third problem that may arise is the *Vanishing Gradient Problem* (51), which a network is more gravely affected by the deeper its architecture. With activation functions that can only take values between 0 and 1, and the cumulative backpropagated error terms that are computed by the chain rule, gradients become "vanishingly" small, as they are multiplied as many times as the network has layers, causing the weights to basically stop changing. If the activation function can take values larger than 1, the gradients may in contrast explode.
As a solution to *Vanishing Gradient Problem*, the Long Short-Term Memory proposed by (29), which will be introduced further below.

7.5 Network architecture

7.5.1 Feed-forward NNs

In feed-forward MLP NNs, an input is fed to the input layer and traversed through the hidden layers without cycles until it reaches the output layer (47). Usually, alls nodes of one layer are connected to all nodes of the next layer, as shown in Picture NUMMER (Rojas p. 134) Picture NUMMER (Benkachcha) shows an MLP feed-forward NN for time series forecasting where Y_{t+1} is predicted by an input of its past values.

7.5.2 Recurrent NNs

According to (9), RNNs are better suited for time series data than FNNS; in fact, bad results in an RNN may hint that there is not time dependency in the data at all. A feedforward NN in its simplest form reduces to an $AR[p]$-model whereas an RNN may reduce to an $ARMA$-model, which is to be kept in mind when analyzing the data structure at hand.
In RNNs, additional "context units" are added to the model, next to the input layer. These store information from previous training loops. This way, they provide feedback about the past. Context units can either connect (1) the input layer with itself, (2) hidden layers with the input layer or (3)the output layer with the input layer. (2) are called *Elman networks*, whereas (3) are called *Jordan networks.* (9) provides two Pictures illustrating the

schemes of these two *Simple Recurrent Networks*: BILDER AUS BALKIN NUMMER

Long Shert-Term Memories (LSTM, (29)) have evolved to become one of the most popular implementations of RNNs (26; Salehinejad et al.), as they can tackle the problem of *vanishing gradients*. In LSTM, hidden units are modified to form *memory cells* that have gates that control (1) input - how much new information is used - and (2) output - or (3) the extent to which information from previous outputs is stored or "forgot". The self-connected *Constant Error Carousels* within the memory cell block use the identity function as their activation function with weight 1; thus its derivative is always 1 so the gradients cannot vanish or explode (51). FIGURE shows an LSTM cell block.

$$g_t^i = \sigma(W_{Ig^i}x_t + W_{Hg^i}h_{t-1} + W_{g^cg^i}g_{t-1}^c + b_{g^i}) \tag{48}$$

$$g_t^f = \sigma(W_{Ig^f}x_t + W_{Hg^f}h_{t-1} + W_{g^cg^f}g_{t-1}^c + b_{g^f}) \tag{49}$$

$$g_t^o = \sigma(W_{Ig^o}x_t + W_{Hg^o}h_{t-1} + W_{g^cg^o}g_{t-1}^c + b_{g^o}) \tag{50}$$

where the matrices $W_{Ig^\bullet}$ is the weight matrix form the input layer, $W_{Hg^\bullet}$ is the weight matrix from the hidden layer and $+W_{g^cg^\bullet}$ is the weight matrix from the cell activation to the input, forget and output gate respectively, and $b_{g^\bullet}$ is the bias (Salehinejad et al.).

8 Variable selection

A well-defined model should include all relevant explanatory variables that contribute to the variance found in the target value and exclude irrelevant, noisy variables. This way, the data and the process constituting it can be better understood, computational time and the so-called curse of dimensionality can be reduced while maintaining a high prediction accuracy (16). The curse of dimensionality describes the problem arising in a high dimensional (i.e. containing many variables) space, where one needs very large samples to accurately depict said space (QUELLE). Variable selection is furthermore important as irrelevant data may cause poor generalization when given new data (16; 10). Variable selection has become an important issue as research is increasingly focussing on high dimensional data, be it in the field of genetics or Data Mining (10). Fortunately, in this case $n > p$, which is why there is no need to give up on reasonable generalization ability, computational speed or model interpretability in favour of a heavily reduced variable subset.

As with many questions that arise in the process of analysing data, the question of how to best select input variables in a time series context, and in this case, to predict freight weight, cannot be answered easily. Directly comparing all 2^p possible subsets of variables is computationally unfeasible, so it is most often infeasible to compare all possible variable subsets and choose the best one.

(62) argue that there is no variable that has zero contribution to the target, and therefore - philiosophically - model selection is a comparison between different sets of patterns (i.e. sets of variables) in order to choose which pattern to focus on, depending on the context in which a research question is proposed. Variable selection also makes a great part in the variance-bias tradeoff, or whether to seek a near-perfect but perhaps overly complex fit of data of the past or a sparse, economic but perhaps overly simple model. Furthermore, minimal computation time and and a reasonably good prediction accuracy are enhanced by thorough variable selection (49). Obviously, the aim should be to find a golden mean between sparsity and accuracy, too many and too few variables.

A first selection of variables should be given by experts, literature of the same field or common sense (28). This is an important step as is may enhance model performance and facilitate the choice of a suitable subset of variables (10) Next to the researchers' expertise, there are several variable selection methods that quantify the explanatory value of all possible variables and include or exclude them in the final model by a prespecified selection criterion.

Albeit the growing number of available methods for variable selection, neither is there one gold standard that is generally useful, nor is there one method especially advisable for each kind of data (55)

8.1 Feature Selection Methods in Machine Learning

Feature selection methods, as variable selection methods are called in the context of Machine Learning, can be classified into *Filter*, *Wrapper* and *Embedded methods* (16), .

8.1.1 Filter Methods

Filter Methods are mainly used in text classification and include standard statistical measures such as Euclidian distance, the Chi-Squared test, the t-Test, ANOVA, correlation coefficients, Markov Blankets and the Fisher Score. Adavantages of Filter Methods are their fast computation times, independency of the classification methods - they only make use of intrinsic properties of the variables - and that feature selection only has to be conducted once (49). But, as they test variables seperately "against" the target variable (e.g. for correlation) to rank them, there's a risk that redundant variables are taken into the model when chosen by a Filter Method (16). Consequently, it is possible that the model established by a Filter Method is not unique, as a different subsets of variables with similar properties w.r.t the Filter Method might have yielded an equal result.

8.1.2 Wrapper Methods

Wrapper Methods measure the predictors' performance through a search algorithm. These methods include *Forward* and *Backward Selection* as well as *Heuristic search algorithms* such as *Simulated Annealing* and *Genetic Algorithms.* Advantageously, *Wrapper Methods* take into account the interdependence of variables, but unfortunately require high computation times and are prone to overfitting (49).

8.1.3 Embedded Methods

Both *Wrapper* and *Embedded Methods* are closely intertwined with the classification algorithm (49) In *Embedded Methods*, variable subset selection is part of the model's training (16), and they combine the search for a variable subset and hypothesis testing (49). They are better able to reduce computation time than Wrapper Methods. An example for an Embedded Method are CARTs (Classification And Regression Trees).

8.2 Variable Selection in AR(I)MA, SVR and NNs

(16) mentions Support Vector Machines and Neural Networks as part of *Embedded Methods.* Resulting from this, for these two methods, variable selection (except for a literature- and expert-driven pre-selection) and model fit could be combined in one step. Similarly, as the AR(I)MAX model can be specified as a linear regression with autocorrelated errors, it is possible to use some kind of stepwise variable selection comparing AICs of the different specifications. As (**?**) showed, AIC minimization and minimization by cross validation are asymptotically equivalent; furthermore, they can be used even when the models being compared are not hierarchically nested (28).
Afterwards, the contribution of the selected variables to the forecast can be compared to the baseline models using only the target variable's time series.

Other approaches to variable selection in the context of freight demand and / or time series forecasting have been found: In their study on the influence of national, regional and local economic indices on freight volume for different segments of the truckload industry in the USA, (24) first examined correlation matrices before using stepwise multiple regression to draw a selection from the large set of initial variables collected. (55) suggest using ensemble methods such as Random Forest to take advantage of several different machine learning methods. (23) used Spearman rank correlation to select external variables to predict railway freight volume. Variable selection methodsfor time series data have also been explored, using NNs (**?**), SVMs (QUELLE) and other, Filter-like techniques (QUELLE).

Obviously, numerous approaches exist that use different methods and comprise of *Filter*, *Wrapper* or *Embedded methods*. In this coursework, to ensure comparability between methods, every model - AR(I)MA, SVM and NN - should initially be fed the same (complete) set of variable. This way, the performance of the AR(I)MA, SVR and NN can be directly compared without being confounded with the use of different subsets of variables for each method. Having said that, the goal of this coursework is to compare the *performance* of different methods as well as models using additional explanatory variables versus the baseline time series. SVM and NN user their own, internal (and rather opaque) feature selection methods. Eventually, they present the variables actually used for estimation, whereas others are more or less discarded by having weights close to zero. These variables could subsequently be excluded from the input.
This is not the case for AR(I)MAX, which needs either a primary *Filter Method* or a *Wrapper* to choose a variable subset that yields the best performance. As AR(I)MA assumes a linear relationship between variables, using a subset of additional variables specifically selected for it might hamper the performance of SVR and NN when provided to them instead of letting them select their own subset. SVR and NN are able to map nonlinear relationships, but being given a subset of variables that was found using a linear assumption might turn out counterproductive. Of course, the same problem might arise when vice versa feeding a variable subset chosen by SVM or an NN to an AR(I)MA model. Thus, including a variable selection process for each method individually and possibly receiving three different "optimal" subsets does not mean that they cannot be compared; instead, this way, the three methods' ability to use additional variables may be compared and overall the ability of additional variables to enhance forecasting performance for freight weight in general. So the variable selection approach chosen for this coursework is to taylor a variable selection process for each inidivdual method and then compare the subsets retrieved by the AR(I)MA, SVR and NN models respectively. One way to ensure some kind of equal conditions could be to use a *Wrapper Method* like e.g. *Backward Elimination* for each model, so as to taylor the subset of chosen variables as closely to the model as possible.
Recursive Feature Elimination, an approach using SVM to backward-eliminate variables from a subset by excluding the variable(s) with the lowest ranking criterion (e.g. the lowest weight), was first introduced by (**?**). An approach very alike to this one but using NNs was proposed by (**?**). This same approach could be adapted to AR(I)MA modeling, but instead of excluding the variables with the highest p-value, comparing subsets of k-1 variables (with k being the number of overall variables) and the resulting AIC, choosing the model with the lowest AIC, and so on. Use the AIC because choosing variable subsets based on p-values will render the p-value invalid (see (28) for a detailed explanation).

The steps of the suggested approach for SVM and NN are generally designed like this:
1. Train a model using the complete set of variables.
2. Compute the ranking criterion (here: weight).
3. Remove the variable that has the lowest rank.

In order to make the algorithm comparable with the one used for model selection in AR(I)MA, instead of ranking the variables, a measure of accuracy is computed for each subset of variables:
1. Train a model using the complete set of variables.
2. Compute an error criterion (e.g. RMSE).
4. Iteratively remove one variable at a time and recompute the error criterion with the obtained subsets.
5. Compare the criterion with the criterion of step 2. 6. Choose the model with higher accuracy. Its subset of variables is now the new "complete" set of variables.
6. Start again at step 1 until no further improvement in the error criterion is achieved.

Of course, this requires a split of the data into training and test sets, which is explained in the following section.

9 Model Evaluation

Generally, data sets are partitioned into training, validation and test data sets in order to estimate parameters (with the training set), fine-tune them during training (on the validation set) and evaluate the model (on the test set). While a validation set is not needed for ARIMA models, it is necessary for NNs (65). Although usually the partitioning is random (in non-time-series data at least) to ensure a good variance of values in the partial data sets, the question arises whether a good (or bad) model performance on the test set is due to the model being well-specified or whether it is not - at least in part - due to the way the data was split. To help eliminate doubts on the accuracy of the model evaluation, *cross-validation* is used. QUELLE In *k-fold cross-validation*, the data set is split into k folds. The model is trained on $k - 1$ folds and tested on the one retained fold, each fold in turn being the *"test fold"* once. In time series, it does not make sense to randomly split the data set into groups, since then the model would also be trained on future values attempting to predict past values. Using the idea of cross-validation sensibly, one could partition the data set in such a ways that in the training set, only values preciding the values in the test set are found. A possible setting may look like the one suggested by (31): The data is sorted by timestamps and then trained on the first p values. Then, the model is tested on the single value $p + 1$. Next, the model is trained

on the first $p + 1$ values and tested on the $p + 2$'nd value and so on. This way, model evaluation makes use of all available observations and the results remain comparable since the test set size remains the same. Asymptotically, this technique, called the "rolling forecast origin", yields the same results as minimization of the *AIC*.
Common metrics to evaluate models (and compare them) are the *Root Mean Square Error (RMSE)* and the *Mean Average Percentage Error (MAPE)*, the latter one of which is especially useful in time series models with a strong trend (43).

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(\overline{y}_t - y_t)^2}{n}} \tag{51}$$

$$MAPE = \sqrt{\frac{\sum_{t=1}^{n}\frac{|\overline{y}_t - y_t|}{|y_t|}}{n}} \tag{52}$$

As the AIC is based on models fit by a likelihood function, it may be used for AR(I)MA model selection; it is not as applicable to SVR, although e.g. (**?**) suggest an approach - its implementation is out of the scope of this coursework. Besides, as already mentioned, it is asymptotically equivalent to cross validation, which is, in combination with RMSE and / or MAPE, applicape to all methods in this coursework.

10 Software implementation

Data preparation and analyses were conducted in `R` (56). The NN was constructed in Python 3.7.2 using Keras (17) with Tensorflow (40) backend. A comprehensive list of the packages used in R and in Python can be found in Appendix (NUMMER).

For all common functions, `stats` was used whenever possible. Furthermore, , (32)'s `forecast` package was chosen for most functions concerning AR(I)MA modeling. If not otherwise specified, functions mentioned in the following belong to said package.

Missing values for time series were imputed by linear interpolation using the function `na.interp`, which was found to supply robust imputation compared to other functions for imputation of mmissing values in time series available in `R` (43).

To specify the AR(I)MA model, the `arima()` function in `forecast` was used. `forecast` furthermore provides the functions `acf()`, `pacf()` for ACF and PACF plots.
For ADF tests, the function `adf.test()` in the package `tseries` was used. In the function, the parameter k (i.e. the lag) is determined according to the

formula suggested by (61) (see subsection NUMMER (Violation of assumptions)), although without t-testing the coefficients of equation NUMMER 4. Manually reducing k and comparing the results yielded no differences in the outcome.

Subsequently, the models' parameters are computed by `arima()` (which is essentially a wrapper on `stats`'s `arima()`, including some extensions, e.g. exogenous variables; possible seasonal components can be modeled by the `fourier()` function given as an argument to `xreg=` in `Arima()`. The order of the fourier series is determined by minimizing the AICc, which is the "corrected AIC" that corrects for the AIC's tendency to favour larger models (62).

$$AICc = -2ln(\hat{L}) + \frac{2k}{n/n-k-1} \tag{53}$$

with the denominator being a bias correction of first order (62).
For AR(I)MAX, `arima()` actually fits a linear regression with AR(I)MA errors (31); no `R` function could be found that models actual ARIMAX models. Because we are dealing with daily forecasts, again, seasonality in the data is captured by passing the additional variables in the `xreg=` argument as a fourier series. Moreover, the `forecast` package provides a function `auto.arima()` which chooses the orders of the AR(I)MA model automatically by minimising the AIC. It can also be given exogenous variables through the `xreg=` argument.

Functions to model Support Vector Regression are contained in the package `e1071` and `kernlab`.

NN IN KERAS

The implementation of the cumstomized recursive feature elimination algortihm was computed manually, but while making use of several functions and packages: The `forecast` package was very useful for cross validation in time series, providing the function `tsCV()`, which was used in combination with `auto.arima()`. In objects of class `ts`, the AIC can be given as an output.
A package that revealed itself to be very useful is the `caret` package that allows to combine SVR-modelling with time series CV by combining the two functions `trainControl()` and `train()` and specifying within them the arguments `method = "timeslice"` (in `trainControl()`) and `method = "svmRadial"` (in `train()` - note that it uses the function provided by the package `kernlab`).

The accuracy metrics RMSE and MAPE were manually computed.

11 Data analysis

11.1 Literature-based variable selection

This thesis is part of a Fraunhofer institute's department for Supply Chain Services (Fraunhofer SCS)'s research project, in cooperation with the University of Erlangen-Nuremberg, the University of Erlangen and NAGEL group as one of several industrial partners. One goal of this research project is to elaborate whether the addition of further information is helpful to predict freight demand (in terms of weight or volume) as opposed to solely basing the forecast on past values.
As the research project is funded by the Federal Ministry for Traffic and Digital Infrastructure, a precondition to which kind of information to choose was to use freely available data sets, such as data provided by the Ministry's *mcloud*, an open online platform with data on mobility and related topics.
When planning the research project, initial ideas were to explore different time scopes and check for seasonalities and calender-related influences on freight demand, such as weekly, monthly, quarterly, or yearly fluctuations and changes in demand due to (upcoming) holidays.

Additional variables found in literature can be partitioned into the following major sections: Calendaric, meterological, sociodemographical, economic and specific effects.

11.1.1 Calendaric effects

Calendaric effects can be divided into two subgroups: periodic and non-periodic effects (**?**). While weekdays, the number of days in a month and certain holidays such as Christmas form part of the former group, holidays such as Easter, that do not occur at the same time every year, belong to the latter group.
(18) accounted for holiday and daily effects on traffic counts and found weekly patterns and holiday impacts especially on the amount of commuters. Since their target was the quantity of passenger transportation on highways and not goods traffic, their results do not exactly indicate the inclusion of holiday and daily effects into the model in this coursework. However, as experts of the industrial partners of Fraunhofer SCS stated the importance of these two factors in their past experience, they are going to be included in this coursework. (18)'s paper provides a useful guidance to how to handle these two variables in time series forecasting.

11.1.2 Meteorological effects

(**?**) found an effect of temperature on the sales of seasonal garments.(**?**) find similar effects of sunshine duration and and increase in consumer spending, which they explain with a psychological phenomenon. (**?**) similarly found that unexpected fluctuations in temperature have an impact on retail sales.

11.1.3 Sociodemographical effects

Population may help in forecasting freight transportation demand, according to (?).

11.1.4 Economic effects

According to (19), GDP has been included as an explanatory variable in other freight models before; (?) included GDP in their paper as well, and furthermore unemployment. GDP as an index to forecast retail sales and freight demand was furthermore used in (?) and (24). (?) provide causal linkage modeling of economic growth on transport.
More economic effects named in studies are key interest rate (?), exchange rate (?), and financial market return (?).

11.1.5 Specific effects

In (Abate)'s study on the effect of fuel prices on average length of haul, inconsistent results were found: A decrease in length of haul before 2008 and an increase after. Furthermore, an improvement of load capacity was detected for higher fuel prices, but also a decline in physical productivity, measured in tonnekilometres. Due to the inconsistency of the findings in this, it is unclear whether fuel price might have an impact on freight weight. Logical considerations, on the one hand, could be that higher prices would hamper demand in deliveries by truck and likely cause a shift towards e.g. rail transportation. On the other hand, infrastructure might not always allow a shift towards other means of transportation, and certain (basic) types of food are demanded regardless of price, so there might not be an influence of fuel price on freight weight (in general), or only on specific types of food, e.g. luxury goods.
(?), however, were able to show an impact of fuel prices on transportation volume.
Moreover, a predictive effect of problems in railway traffic could be found (QUELLE)

11.2 Critical acclaim of literature-based variables

In total, 16 additional variables were found in literature.

A precondition to time series modelling is causality, which is logical and obvious for holidays and daily effects determining freight demand (freight demand cannot influence the ocurrence of holidays); not exactly as much for

fuel price and GDP.
It might make sense to think that the demand for food haulage impacts fuel prices (as prices are a matter of demand and supply), or that it influences the GDP in some indirect way (and (19) admit to this two-way point of view). In fact, the GDP (in this case: the German BIP) is a measure of the value of all inland produced goods and services in a certain time period (Statistisches Bundesamt) and is influenced by demand (Bundesregierung). Still, in the case at hand, for the prediction of daily freight demand, knowing the current GDP might be a useful tool - as e.g. (24), (**?**), (**?**) and (33) have found (and they name further studies with similar results as theirs).
Data on GDP was found as raw data, and adjusted data of two varieties; the so-called x12-arima adjustment and the bv4.1 adjustment. Unfortunately, literature on GDP as an indicator does not specify which kind of values were used, which is why initially, all three varieties of GDP data were included in the variable set.
Considering fuel prices, the counter-argument to higher fuel prices dampening demand for truck haulage would be that an increasing demand in truck haulage may lead to higher fuel prices. However, it is not very likely that an increase in truck haulage for food deliveries in Germany significantly raises prices that are determined on a world-wide level.

Data on key interest rates includes three different kinds: the rate on deposit facility "which banks may use to make overnight deposits with the Eurosystem", ECB main refinancing operations "which provide the bulk of liquidity to the banking system" and marginal lending facility " which offers overnight credit to banks from the Eurosystem" (**?**), all of which were included in the initial set of additional variables, as again, literature did not specify the type of key interest constituting a valid indicator.
The type of exchange rate mentioned by (**?**) was not exactly specific, as it only mentioned the South Korean Won (the researchers are based in South Korea); therefore, for this coursework, the exchange rate between US-Dollar and Euro was chosen, als Euro is the currency in Germany and the US-Dollar the most accepted currency worldwide. One issue arises with stock market data that has to be kept in mind: stock markets are closed on weekends (classically). The data found for exchange rates was thus missing values for all Saturdays.
Some of the variables considered to be included in the data set are less granular than the target variable - i.e., they are not measured or measurable on a day-to-day basis. These variables include the days in a month, population, key interest rate, GDP, consumer price index.

A very practical consideration to whether or not include a variable in a dataset is the goal of the research project to use data that are openly available. For the following variables, data was obtainable from open sources such

as GENESIS (54) or *mcloud*: Holidays, school holidays, temperature, rainfall, GDP, population, fuel prices, key interest rate, railway traffic (though sparse).
For the following variables, this was not the case (March 2019), which is why they were not taken into further consideration: Exchange rate, and financial market return.

11.3 Sources for external data

All data retrieved were restricted to the Federal State of Bavaria between 2015-09-17 and 2016-12-31,as this was the time period of the data provided by NAGEL.

Encoding of weekdays, holidays, holiday weeks and the sum of days in a month was manually computed, as well as the number of days in a month.
Data on holidays and school holidays can be retrieved from the website ().b.

Data on temperature was provided in the initial data set received from University of Regensburg. Furthermore, data on average temperature per day at a weather station located in Nuremberg (the origin of all transports in the dataset at hand) was found on the website of the *Deutscher Wetterdienst (DWD)* on *mcloud*.
This data set also provides data on sunshine duration, rainfall and type of rainfall. The type of rainfall was categorized as the following:
0 - no rainfall
4 - rainfall reported, but type of rain unknown
6 - only rainfall
7 - only snow
8 - rain and snow
9 - missing value
The type of rainfall was re-coded to "0" for no rainfall and "1" for all other types; there were no missing values in the time period of interest.

Data on GDP (in German: BIP), population and consumer price index are accessible from the German Federal Statistical Office (Destatis)'s database "GENESIS".
Data on key interest rates was found on the homepage of the European Central Bank: on this URL (30th March 2019).

Data on fuel prices are accessible from the website of the "German General Automobile Club" (Allgemeiner Deutscher Automobilclub, ADAC) (ADAC).

Data on railway delays was obtained from an API of *Deutsche Bahn*, (on this URL). Unfortunately, data was only available for the time period between 2016-01-15 and 2016-11-30.

Data on exchange rates were taken from this website.

11.4 Data Preprocessing

A first glance at the raw data revealed that the variable "gewicht" (weight) had 49 missing values, there are 5,525 different senders and there are 15,576 different recipients. Furthermore, there are 28 different origin-destination relations; note that there is only one origin in the data set which is Nuremberg (code number 90). There are only 26 weather stations which is 2 fewer than destinations; probably because there are no weather stations near two of the destinations. Some variables ("ABS_REL", "Ziel_REL", and "WetterStationID") had to be redefined as factor. Furthermore, the variable quarter incorrectly labeled dates in the fourth quarter of the year as being in the third quarter, which had to be corrected.
The missing values in gewicht ocurred mostly on holidays:
2015-12-25 - Boxing Day
2016-01-01 - New Year
2016-03-27 - Easter Sunday
2016-05-15 - Whitsunday (Pfingstsonntag)
2016-10-02 - Thanksgiving
2016-12-10 - a regular Saturday; no known holiday
2016-12-25 - Boxing Day

Lastly, the raw data set was transformed into a data.frame aggregated by date and all additional variables were added. It contains the following variables:
"Date"
"Weekday"
"Weekday_No" (numeric representation of the weekday)
"Month"
"Weekend" (boolean; 1 if date was on a weekend)
"Quarter"
"Holiday" (boolean; 1 if date was on a holiday)
"HolidayWeek" (boolean; 1 if date was in a week in which a holiday ocurred)
"Quantity" (number of shipments on a given date)
"Weight"
(daily weight of freight shipment) "Size"
(Weight / Quantity) "Temp" (average temperature measured on a given date)
"msum" (number of days in a month)
"schoolhol" (boolean; 1 if date was within school holidays)
"sun" (sunshine duration)

"rain" (daily precipitation height in mm)
"raintype" (type of precipitation)
"humid" (mean daily relative humidity in %)
"Super.E10" (fuel prices for Super E10)
"Diesel" (fuel prices for Diesel)
"leitzins1" (ECB deposit facility - change in % p.a.)
"leitzins2" (ECB marginal lending facility - change in % p.a.)
"leitzins3" (ECB main refinancing operations - change in % p.a.)
"GDPoriginal" (raw GDP)
"GDPx12" (x12-arima adjusted GDP)
"GDPbv4" (bv4.1 adjusted GDP)
"CPIby" (consumer price index in Bavaria, for all cateogries)
"CPIfood" (consumer prices index in Germany, for food)
"pop" (population in Bavaria)
"endexrate" (exchange rate Dollar-Euro at the end of the day)
"openexrate" (exchange rate Dollar-Euro at the opening of the day)
"exratehigh" (highest exchange rate Dollar-Euro during a given day)
"exratelow" (lowest exchange rate Dollar-Euro during a given day)

The missing values for all Saturdays of the exchange rate data was dealt with by applying na.locf() ("last observation carried forward), essentially adopting a simplified view of stock markets as more or less just falling asleep on Saturday before coming back to life again as if nothing happened.

References

[.b] Schulferien 2015-2017.

[Abate] Abate, M. Does fuel price affect trucking industry's network characteristics? : evidence from denmark.

[ADAC] ADAC. Kraftstoff-durchschnittspreise.

[4] Adhikari, R. (2015). A neural network based linear ensemble framework for time series forecasting. *Neurocomputing*, 157:231–242.

[5] Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.

[6] Andreoni, A. and Postorino, M. N. (2006). Time meries models to forecast air transport demand: A study about a regional airport. *IFAC Proceedings Volumes*, 39(12):101–106.

[7] Anggraeni, W., Vinarti, R. A., and Kurniawati, Y. D. (2015). Performance comparisons between arima and arimax method in moslem kids

clothes demand forecasting: Case study. *Procedia Computer Science*, 72:630–637.

[8] Arlt, J., Trcka, P., and Arltová, M. (2017). The problem of the sarima model selection for the forecasting purpose. *Statistika*, 97:25–32.

[9] Balkin, S. (1997). *Using Recurrent Neural Networks for Time Series Forecasting*. International Symposium on Forecasting. Barbados.

[10] Ben Ishak, A. (2016). Variable selection using support vector regression and random forests: A comparative study. *Intelligent Data Analysis*, 20(1):83–104.

[11] Benkachcha, S., Benhra, J., and El Hassani, H. (2015). Seasonal time series forecasting models based on artificial neural network. *International Journal of Computer Applications*, 116(20):9–14.

[12] Box, G. E. P. and Jenkins, G. M. (1976). *Time series analysis: Forecasting and control*. Holden-Day series in time series analysis. Holden-Day, San Francisco, Calif., rev. ed. edition.

[13] Bundesministerium für Umwelt, Naturschutz und nukleare Sicherheit (2018). Klimaschutzbericht 2018: zum aktionsprogramm klimaschutz 2020 der bundesregierung: Referentenentwurf.

[14] Bundesministerium für Verkehr und Infrastruktur (2014). Verkehrsverflechtungsprognose 2030.

[Bundesregierung] Bundesregierung. Indikator 10: Wirtschaftliche leistungsfähigkeit.

[16] Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28.

[17] Chollet, F. et al. (2015). Keras.

[18] Cools, M., Moons, E., and Wets, G. (2009). Investigating the variability in daily traffic counts through use of arimax and sarimax models. *Transportation Research Record: Journal of the Transportation Research Board*, 2136(1):57–66.

[19] de Jong, G., Gunn, H., and Walker, W. (2004). National and international freight transport models: An overview and ideas for future development. *Transport Reviews*, 24(1):103–124.

[20] Dorffner, G. (1996). Neural networks for time series processing. *Neural Network World*, 6:447–468.

[21] Ďurka, P. and Pastoreková, S. (2012). Arima vs. arimax–which approach is better to analyze and forecast macroeconomic time series. *Proceedings of 30th International Conference Mathematical Methods in Economics.*

[22] Falk, M., Marohn, F., Michel, R., Hofmann, D., Macke, M., Sprachmann, C., and Englert, S. (2012). *A First Course on Time Series Analysis: Examples with SAS*. Epubli, Berlin.

[23] Feng, F., Li, W., and Jiang, Q. (2018). Railway freight volume forecast using an ensemble model with optimised deep belief network. *IET Intelligent Transport Systems*, 12(8):851–859.

[24] Fite, J. T., Don Taylor, G., Usher, J. S., English, J. R., and Roberts, J. N. (2002). Forecasting freight demand using economic indices. *International Journal of Physical Distribution & Logistics Management*, 32(4):299–308.

[25] Gao, S., Zhang, Z., and Cao, C. (2011). Road traffic freight volume forecast using support vector machine combining forecasting. *Journal of Software*, 6(9).

[26] George, K., Harish, M., Rao, S., and Murali, K. (2017). Comparison of neural-network learning algorithms for time-series prediction. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI): took place 13-16 September 2017 in Manipal, Mangalore, India, India*, pages 7–13, Piscataway, NJ. IEEE.

[27] Hastie, T., Tibshirani, R., and Friedman, J. H. (2017). *The elements of statistical learning: Data mining, inference, and prediction.* Springer series in statistics. Springer, New York, NY, second edition, corrected at 12th printing 2017 edition.

[28] Heinze, G., Wallisch, C., and Dunkler, D. (2018). Variable selection - a review and recommendations for the practicing statistician. *Biometrical journal. Biometrische Zeitschrift*, 60(3):431–449.

[29] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

[30] Hunt, Ü. (2003). Forecasting of railway freight volume: Approach of estonian railway to arise efficiency. *Transport*, 18.

[31] Hyndman, R. J. and Athanasopoulos, G. (2018). *Forecasting: Principles and practice.* 2. auflage edition.

[32] Hyndman, R. J. and Khandakar, Y. (2008). Automatic time series forecasting: The forecast package for r. *Journal of Statistical Software*, 27(3).

[33] Intihar, M., Kramberger, T., and Dragan, D. (2017). Container throughput forecasting using dynamic factor analysis and arimax model. *PROMET - Traffic&Transportation*, 29(5):529–542.

[34] James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning: With Applications in R*, volume 103 of *Springer Texts in Statistics*. Springer, New York, NY.

[35] João F. L. Oliveira and Teresa Bernarda Ludermir (2014). Iterative arima-multiple support vector regression models for long term time series prediction. In *ESANN*.

[36] Karsoliya, S. (2012). Approximating number of hidden layer neurons in multiple hidden layer bpnn architecture. *International Journal of Engineering Trends and Technology*, 3:714–717.

[37] Khandelwal, I., Adhikari, R., and Verma, G. (2015). Time series forecasting using hybrid arima and ann models based on dwt decomposition. *Procedia Computer Science*, 48:173–179.

[38] Kourentzes, N., Barrow, D. K., and Crone, S. F. (2014). Neural network ensemble operators for time series forecasting. *Expert Systems with Applications*, 41(9):4235–4244.

[39] Lee, M. H. and Hamzah, N. (2010). Calendar variation model based on arimax for forecasting sales data with ramadhan effect. *Proceedings of the Regional Conference on Statistical Sciences*.

[40] Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Corrado, G. S., Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Jia, Y., Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng (2015). Tensorflow: Large-scale machine learning on heterogeneous systems.

[41] Miller, J. (2018). Arima time series models for full truckload transportation prices. *Forecasting*, 1(1):121–134.

[42] Mircetic, D., Nikolicic, S., Maslaric, M., Ralevic, N., and Debelic, B. (2016). Development of s-arima model for forecasting demand in a beverage supply chain. *Open Engineering*, 6(1):14.

[43] Moritz, S., Sardá, A., Bartz-Beielstein, T., Zaefferer, M., and Stork, J. (2015). Comparison of different methods for univariate time series imputation in r. *CoRR*, abs/1510.03924.

[44] Nakamura, E. (2005). Inflation forecasting using a neural network. *Economics Letters*, 86(3):373–378.

[45] Pinto, R. and Cavalieri, S. (2005). Seasonal time series prediction with artificial neural networks and local measures. *IFAC Proceedings Volumes*, 38(1):337–342.

[46] Regan, A. and Garrido, R. (2001). Modelling freight demand and shipper behaviour: state of the art, future directions. *Travel Behaviour Research*.

[47] Rojas, R. (1996). *Neural Networks: A Systematic Introduction*. Springer, Berlin and Heidelberg.

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

[49] Saeys, Y., Inza, I., and Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics (Oxford, England)*, 23(19):2507–2517.

[Salehinejad et al.] Salehinejad, H., Sankar, S., Barfett, J., Colak, E., and Valaee, S. Recent advances in recurrent neural networks.

[51] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117.

[52] Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.

[Statistisches Bundesamt] Statistisches Bundesamt. Bruttoinlandsprodukt (bip).

[54] Statistisches Bundesamt (2019). Genesis-online.

[55] Taghizadeh, E. (2017). *Utilizing Artificial Neural Networks to Predict Demand for Weather-Sensitive Products at Retail Stores*. Detroit, Michigan, USA.

[56] Team, R. C. (2013). R: A language and environment for statistical computing.

[57] Transport and Infrastructure Council (2016). Australian transport assessment and planning guidelines: T1 travel demand modelling.

[58] Tsekeris, T. and Tsekeris, C. (2011). Demand forecasting in transport: Overview and modeling advances. *Economic Research-Ekonomska Istraživanja*, 24(1):82–94.

[59] Umweltbundesamt (2018a). Emissionen des verkehrs.

[60] Umweltbundesamt (2018b). Fahrleistungen, verkehrsaufwand und modal split.

[61] Vogel, J. (2015). *Prognose von Zeitreihen: Eine Einführung für Wirtschaftswissenschaftler*. Springer Gabler, Wiesbaden.

[62] Wit, E., van den Heuvel, E., and Romeijn, J.-W. (2012). 'all models are wrong...': an introduction to model uncertainty. *Statistica Neerlandica*, 66(3):217–236.

[63] Zhang, G., Patuwo, E. B., and Hu, M. Y. (1998). Forecasting with artificial neural networks. *International Journal of Forecasting*, 14(1):35–62.

[64] Zhao, J., Cai, J., and Zheng, W. (07022018). Research on railway freight volume prediction based on arima model. In Wang, X., Zhang, Y., Yang, D., and You, Z., editors, *CICTP 2018*, pages 428–437, Reston, VA. American Society of Civil Engineers.

[65] Zhou, L., Heimann, B., and Clausen, U. (2006). Short term demand forecasting for a typical logistics service provider.