A comparison between artificial neural networks and ARIMA models in traffic forecasting

Delft University of Technology

Luc Lenferink

Abstract

Motivation: Traffic forecasting is becoming a vital component of our travel experience. It plays a key role in intelligent transportation systems that allow us to make smarter use of existing transportation networks. This study focuses on the possible role of artificial neural networks in these systems and what data can be best feed in to them to retrieve the best results.

Aim: The goal of this study is to see whether two layered feed forward neural networks outperform the statistical ARIMA model in motorway traffic forecasting. In specific, whether or not the usage of upstream and multivariate data decreases the forecasting errors of the neural network, how this relates to the amount of samples used as input, and how this relates to the amount of time steps that is forecasted ahead.

Results and conclusions: Two different traffic networks are used to train and test the models. The testing results show that, when doing predictions using time steps covering 10 minutes of traffic data and forecasting one time step ahead, the optimal amount of samples used as input is 4. Increasing the input length after this does not result in better predictions, it even slightly increases the prediction errors. Moreover, it became clear that up to 3 or 4 time steps forecasting in the future, the neural networks using upstream data outperform the ARIMA model. After this an ARIMA model that uses deseasonalized data or a neural network that uses deseasonalized data is a better option. There is always a two layered neural network that outperforms the ARIMA models. Furthermore, the usage of upstream data almost always decreases the prediction errors. This is different with the usage of multivariate data, which hardly contributes to a better prediction in the used form.

Keywords

Traffic, Artificial Neural Networks, Multivariate Time Series, Upstream data.

I. Introduction

Traffic forecasting plays a key role in intelligent transportation systems (ITSs). Work on intelligent transportation systems started in 1994 when the first ITS World Congress was held [1]. A key element in an ITS is the ability to forecast traffic.

There growing need for ITSs. We cannot rely on road adjustments only, figure 1 shows why. From this figure we can conclude that the amount of motor vehicles on the Dutch road is growing rapidly, after a stagnation between 2012 and 2015 it has picked up the pace with a growth rate of 19,7% in 2018. Based on this growth the Dutch ministry of transport estimates a growth in travel time loss of 35% [2].

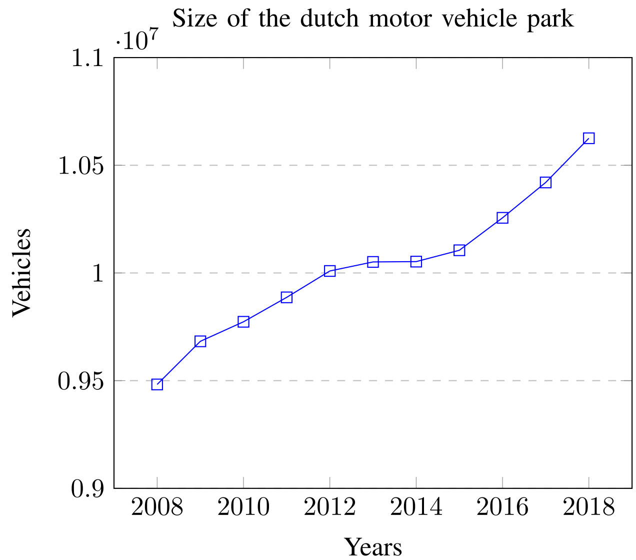


Fig. 1: Size of the Dutch motor vehicle park[1].

Besides the development of ITSs, short term traffic forecasting can for example be used to develop applications that calculate the departure times with the shortest travel time for a given route. Where short term prediction deals with the network in its current state, long term traffic prediction can also be used to predict future bottlenecks and the need for road adjustments.

This paper focuses on the sort term prediction of motorway traffic, in specific 10, 20, or 30 minutes ahead. The statistical ARIMA model is compared to two layered artificial neural networks. The networks differ in the size of the input layer and the form of the input data. Previous work done on this topic

[1]Numbers retrieved from the Dutch government institute CBS (Statistics Netherlands) Retrieved April 25, 2019 from https://statline.cbs.nl/StatWeb/

mainly focuses on new implementations of neural networks. They make use of different input lengths, upstream data and multivariate data. A study on the possible combination or a comparison between these models/techniques is missing.

This study tries to explore the boundaries regarding to what is possible with a two layered neural network in the field of motorway traffic forecasting.

This paper will address the following: 1) the effects of increasing the input length on the prediction capabilities of a model, 2) the effects of using upstream data in downstream predictions, 3) the effects of using multivariate data, and 4) how this all relates to the amount of time steps that is predicted in the future.

II. CONCEPTS

A. Time series

The analysis and predictions in this study are based on time series. A time series is a collection of observations $x_t \in \mathbb{R}^d$, recorded at times $t = 0, 1, ..., n$. A whole sequence of measurements is denoted by $\{x_t\}$ and by $\{X_t\}$ when viewed as a realization of random variables.

There are various definitions for a multivariate time series. In this paper we refer to a multivariate time series as a time series with values of multiple different features per time step. To be specific, a multivariate time series is in this paper a collection of the form $x_t \in \mathbb{R}^2$ i.e. two features per time step, vehicle flow rate (number of vehicles per hour) and average vehicle speed (km\h).

In other papers both the usage of upstream data and the amount of time steps used for a sample are also referred to as multivariate or multivariate time series. In this paper, the usage of upstream data is explicitly mentioned and the amount of time steps used for a sample is referred to as input length.

B. Upstream traffic

Some of the predictions in this paper make use of upstream traffic data. This refers to the usage of measurement sites that lie on the same road and measure traffic that still has to pass the measurement site that is used as target. The "Data - Selection and retrieval" section in the research method gives the exact measurement sites that are included when predictions are made with upstream data.

C. Statistical models

In this section the ARIMA (autoregressive integrated moving average) model is discussed. ARIMA is a statistical model used for time series analysis, either to get a better understanding of the data or to do forecasting.
A time series X_t can be decomposed in the following way $X_t = m_t + s_t + Y_t$ where:

- m_t: a slowly changing trend component
- s_t: a periodic function that reflects a seasonal component
- Y_t: a random noise term that is assumed to be stationary

In order to fit such a statistical model to a time series, it needs to be stationary. A stationary time series has constant expectation and variance over time, so trends and in some cases seasonality indicate nonstationarity. A more formal definition is the following: A time series $\{X_t\}$, with $\mathbb{E}[{X_t}^2] < \infty$ is stationary if the following holds:

- $\mathbb{E}[X_t] = \mu$ is independent of t.
- $Cov(X_{t+h}, X_t)$ is independent of t for each h.

A traffic time series is likely to contain trends[2] and seasonality. If needed, the data can be transformed such that the stationarity assumption is more reasonable. This is done by differencing as explained in the next section.

The last step in the process is to find a suitable probabilistic model to fit the time series. This is done using the ACF[3] (Autocorrelation function) and PACF[4] (Partial autocorrelation function). Details about this process can be found in *Introduction to Time Series and Forecasting* (Brockwell and Davis, 2002).

The forecasting can now be achieved by: 1) fitting the model based on the ACF and PACF, 2) making a forecast based on the input data, and finaly (if the data was differenced) 3) inverse differencing the predictions.

D. Differencing

Differencing ($\nabla(X_t) = X_t - X_{t-i}$, with interval i) can be used to remove trends and/or seasonality and thus to make a time series stationary. This process can be repeated until the time dependence is removed.
After the predictions have been made, the seasonality and/or trends can be brought back into the data by inverse differencing ($\nabla^{-1}(X_t) = X_t + X_{t-i}$, with interval i). By doing this the predicted values can be compared with the non differenced targets.

Figure 2 shows an example of this process without the forecasting, with interval $i = 2$. So if we follow the value at time step $t = 3$ we get: differencing $8 - 2 = 6$ and inverse differencing $6 + 2 = 8$.

non-differenced	value	2	4	8	16	32
	time step	1	2	3	4	5
differenced	value			6	12	24
	time step			3	4	5
inverse differenced	value			8	16	32
	time step			3	4	5

Fig. 2: Differencing process without forecasting

[2] This depends on the term, short term data will most likely not contain any trends.

[3] $\rho x(h) = \frac{Cov(X_{t+h}, X_t)}{Var(X_t)}$

[4] As defined in Introduction to Time Series and Forecasting [3]

E. Artificial neural networks

Artificial neural networks (ANN) are computing systems that contain many non-linear computing nodes interconnected by directed edges. An ANN can be partitioned into layers, in which every node in a layer has directed edges connected to the nodes in the next layer. The first and the last layer are called the input layer and the output layer respectively. The layers in between are called the hidden layers. Every edge has a weight and the network is trained by modifying these weights. The network is basically a function that maps the input layer onto the output layer.
The simplest form of an ANN is a so called "feed forward" neural network (FNN). In a FNN the information always moves in one direction, the graph does not contain cycles. A two-layer FNN is shown in Figure 3.

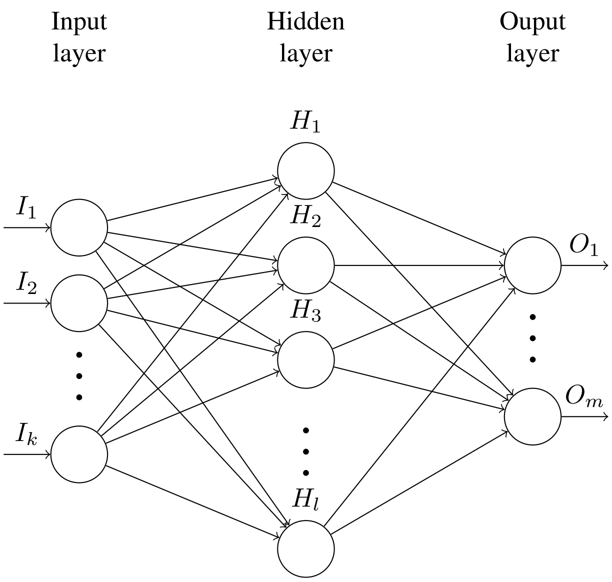


Fig. 3: Structure of a two layer feed forward neural network model

All layers have an activation function. An activation function calculates the output of a layer based on its input and the weights. Activation functions add non-linearity properties to the network. They allow the network to learn complex function mappings. The networks learns by finding the minimum of the loss function, the metric that is minimized during the training. This is done by taking derivatives. The optimizer function then determines at what rate the weights of the network are changed.
Two other types of neural networks that pass by in the literature study are: Convolutional neural networks (CNN), and recurrent neural networks (RNN). CNNs are fully connected and their layers often have higher dimensions. RNNs contain cycles, which allows for more dynamic behavior.

F. Metrics

1) Mean squared error: The mean squared error (MSE) is the average of the squares of the errors, where the error is the difference between what is to be estimated and what is estimated. The squaring is done such that the negative values do not cancel positive values.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

2) Root mean squared error: The root mean squared error (RSME) is similar to the MSE, but it has the same units as the quantity that is that is predicted and is therefore easier to interpret.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

3) Mean absolute error: Similar to the metrics above, the mean absolute error (MAE) also avoids that negative values cancel out positive values but takes the absolute value of the error instead of squaring the error. Just as the RMSE, it expresses the average prediction error in the same units as the quantity that is predicted.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Both the RMSE and the MAE express the average prediction error in the same units as the quantity that is predicted. There are however some differences. The RMSE squares the errors before the average is taken and gives therefore relatively heavy weights to bigger errors in comparison to the MAE.

III. Literature Review

With the rise of intelligent transportation systems, research in traffic forecasting increased in popularity. A lot of previous research mainly focussed on univariate linear statistical models such as Kalman filtering and various combinations and versions of autoregressive integrated moving average (ARIMA) models.

A. ARIMA Techniques

Besides univariate models researchers have searched for increased predicting performance by using upstream traffic and splitting components:

The papers of Williams [4] and Kamarianakis and Prastacos [5] brought up the idea to include upstream traffic in the ANN forecasts.

Williams investigated a multivariate ARIMAX forecast model in which upstream data is used to improve forecasts at downstream locations. The upstream traffic flow time series are treated as transfer function inputs into the ARIMA model for the forecast location. This model is compared to an univariate ARIMA model. He concludes that the trade-off between increased complexity and increased forecast accuracy must be carefully weighed in any decision between univariate and multivariate forecasting.

Kamarianakis and Prastacos compared the univariate ARIMA with the multivariate VARMA (vector ARMA) and

STARIMA (space-time ARIMA). Although the multivariate models are expected to be more accurate they found that the different models have similar forecasting performance.

Note that, for both studies, multivariate refers the amount of measurement sites used for prediction, i.e. upstream/downstream, and not to multivariate time series.

Ghosh et al. [6] developed a multivariate short-term traffic forecasting algorithm that is parsimonious and computationally simple. The structural time series model they use models the different components of the time series, such as trends and seasonal components separately.

This sparked the idea to remove trends with differencing before forecasting, instead of modeling the trends and components seperately as the paper suggests.

B. ANNs

More recently the research on forecasting multivariate time series is shifting to artificial neural networks (ANNs), many researchers have claimed to reduce the prediction errors with ANNs. Examples of which are:

One of the papers that together with the papers of Williams [4], Kamarianakis and Prastacos [5] and Ghosh et al. [6] form the base of this comparision study is the paper of Charkraborty et al. [7].

Charkraborty et al. are using simple FNNs to predict floor prices. They performed three experiments: separate modeling, combined modeling and single modeling. Without going into the details about these experiments, they concluded that the combined modeling, i.e. using multivariate time series with flattening, gives the best performance. Finally they compared this model to an ARMA model to conclude that the combined model is better. Claiming that: the root mean squared errors (in prediction) obtained using this approach are better than those obtained from the statistical model by at least an order of magnitude. The graphs that are included show that especially the multi-lag prediction is more accurate.

The study of Raman et al. [8] confirms this. It uses FNNs to predict water resources. They trained the networks to learn the patterns in the multivariate time series. The results from this model show similar results to [7]. They conclude that neural networks are a good alternative compared to autoregressive models for the multivariate modelling of water resources time series.

More traffic related examples are the studies of Kumar et al. [9] and Gltekin et al. [10].

Kumar et al. use ANNs to predict non urban highway traffic volume using average speeds and quantities of vehicles. The study compares ANNs with different transfer functions, different learning methods and different amounts of neurons in the hidden layer. The results that ANNs are able to predict traffic volumes with a consistent performance.

Gltekin et al. use ANNs to model historical traffic data and to predict traffic volume. The research uses two different FNN structures to forecast 5 minutes and 1 hour ahead. The results of the study show that the features used: day of the week, hour, minute and last cases were very effective in predicting the traffic volume.

C. RNNs and CNNs

Other papers related to traffic forecasting that use more advanced neural networks include:

The paper of Ishak et al. [11] which is an example of a comparison study. They present an approach that use Jordan-Elman networks, partially RNNs, and time-lagged FNNs to predict short term traffic. They compare their findings to several ARIMA models proposed in other studies. The three proposed network topologies outperformed in most cases the statistical models.

Another interesting studies on this topic are:

The research of Li et al. [12]. They use a Diffusion Convolutional Recurrent Neural Network (DCRNN) to capture the spatiotemporal dependencies in the traffic data. This way recurring events such as rush hours or accidents can be captured. They observed a consistent improvement of 12% - 15% over state-of-the-art baselines. Among these were ARIMA and FNN models.

The study of Yun et al. [13], which studied the relation in forecasting traffic volume between data characteristics and the forecasting accuracy of different models. They conclude that a time-delayed RRN gives the best results.

The paper of Dia [14] which uses a TLRN (time-lag recurrent network) to predict traffic speed

And the paper of Ma et al. [15] which proposes a CCN based method that learns traffic as images to forecast traffic speeds.

D. Combinations

Furthermore research has been done in the combination of statistical models and/or artificial neural networks:

Van der Voort et al. [16] introduce the KARIMA method, this method uses a self organizing neural network as an initial classifier in which each class has an individually tuned ARIMA model associated to it. This method is compared a method called ATHENA, a complicated method that is superior to ARIMA but suffers from practical difficulties since it is more of a brute force approach. They conclude that the level of performance of the KARIMA method is equal to the ATHENA method and is substantially better than straightforward ARIMA modeling.

Zheng et al. [17] designed a method that tries to combine several predictors based on the bayes rule. The method combines two neural networks into one.

E. Conclusion

From the literature study it became clear that there is a lot of research in the combination of traffic forecasting and neural networks. What is missing however is a comparison between models that used different ways to prepare their data and seemed successful, i.e. studies on input length, multivariate data, upstream data. For example Charkraborty et al. [7] do not compare their multivariate model to model that use upstream data.

IV. Research Method

A. Research Questions

The main research question is formulated as follows: Which approach for the neural network, upstream/local, multivariate/univariate would give the best motorway traffic predictions? With this question in mind I defined four sub questions that I will try to answer. The questions are answered using motorway traffic data, each time step covers 10 minutes of traffic data. Some testing pointed out that this suited the experiments best. The amount of noise is relatively low, the same holds for the amount of networks that have to be trained to reach a conclusion.

RQ$_1$. What are the effects of increasing the input length on the prediction capabilities of the different models? - The answer to this question gives us a starting point in this research. The input length (the amount of samples used for prediction) is one of the variables that can be used to improve the predictions and is therefore worth looking at. Is there possibly an optimal input length and how does this relate to the sample skip size (the amount of time steps predicted in the future)? To get an idea of the effects, an upstream multivariate model was trained using different input lengths ranging from 1 to 6 and sample skips 1, 3, and 5.

RQ$_2$. What is the effect of increasing the number of time steps on the prediction capabilities of the different models? - The goal of this question is to get an overview of which model performs best in relation to the sample skip size. In particular: Does the order of best performing models change when the sample skip size is increased? This sub question is answered by training all the models on samples skips ranging from 1 to 3. After this the best models and the models with the least prediction decay are selected to be trained on sample skips ranging from 1 to 10.

RQ$_3$. What are the effects of using upstream data in downstream predictions? - Previous results [4], [5], and [16], have shown that the usage of upstream traffic data is useful for the prediction of downstream traffic. The networks used in this research are of a different scale size. It is therefore interesting to see if the same holds for a network which consist of solely a motorway traffic junction and a network from which we include the on-ramps and off-ramps. To answer this question we have a look at the results produced for the previous questions.

RQ$_4$. What are the effects of using multivariate time series instead of univariate time series? - The goal of this question is to see whether the usage of both the vehicle flow rate and the average vehicle speed gives better predictions than the usage of solely vehicle flow rate. The question is answered using the data obtained from answering the questions above.

B. Data - Selection and retrieval

The Dutch motorway system is closely monitored by the traffic control stations of Rijkswaterstaat[5]. Rijkswaterstaat is responsible for the management of the main infrastructure facilities in the Netherlands. The traffic measurements are made using detection loops, see figure 4.

Fig. 4: A motorcycle passing a detection loop[6].

The loops measure how much traffic and how fast traffic is passing by. The data retrieved from these measurement sites is collected by the NDW[7] (the Dutch national data bank for traffic data). The NDW has made their data publicly available, it gathers data from 20.000 measurement sites covering 10700 km of roads[8]. The measurement sites of the NDW measure every minute (for every lane) both the vehicle flow rate in vehicles per hour and the average vehicle speed in kilometers an hour.

The amount of data offered, the availability of the data, and the acurracy of the data make the NDW data base a good data source for this research.

For this paper, two motorway networks, Junction Kethelplein and A4 Delft, see Appendix A, are composed to compare the different forecasting methods. The names indicate the approximate locations of the chosen measurement sites, the exact locations and measurement sites can be found in Appendix B.

A specific measurement site is referred to as Name_Direction1 when the traffic on the lane(s) enters the network and Name_Direction0 when the traffic on the lane(s) leave the network. So for example, Rijswijk_S1 refers to a measurement site located on the on-ramp of the Rijswijk ramp in southerly direction.

[5]Traffic control stations Rijswaterstaat: https://www.rijkswaterstaat.nl/over-ons/onze-organisatie/organisatiestructuur/verkeer-en-watermanagement/index.aspx

[6]Google. (n.d.). [Google Street View, Den Haag, Utrechtsebaan, hectometer pole 3.7, north-west direction]. Retrieved June 17, 2019, from https://www.google.nl/maps

[7]NDW: https://www.ndw.nu/pagina/nl/103/datalevering/120/open_data/

[8]Numbers retrieved from the NDW. Retrieved April 25, 2019 from https://www.ndw.nu/pagina/nl/4/databank/31/actuele_verkeersgegevens/

For the network Junction Kethelplein, local predictions refer to the prediction of East_E0 only using data of East_E0. Upstream predictions refer to the prediction of East_E0 using, East_E0, North_S1, West_E1, and South_N1.
For the network A4 Delft, local predictions refer to the prediction of South_S0 only using data of South_S0. Upstream predictions refer to the prediction of South_S0 using, Delft_S0, Delft_S1, Den_Haag-Zuid_S0, Den_Haag-Zuid_S1, Rijswijk_S0, Rijswijk_S1, and North_S1.

C. Data - Preparation

Having well prepared data reduces the modeling errors and the subsequent prediction errors. The data preparation is done in two parts.
Part one, which defines the problem, will explain how different predicting problems can be created using four variables: the **differencing interval**, the usage of **multivariate data** or not, the **input length** and the **sample skip**.

1) Differencing
2) Vectorizing
   a) Multivariating
   b) Timeseriealizing
   c) Flattening

And part two, which prepares the samples and targets for training:

3) Normalizing
4) Splitting

1) Defining the problem - Differencing: The usage of an ARIMA model requires the data to be stationary, see the ”Concepts - Statistical models” section. This means that if an ARIMA model is used and the data is not stationary, trends and in some cases seasons have to be removed. The raw traffic data that is used to make the predictions does not contain trends. For traffic data this is something that comes in to play when the predictions are made over longer periods of time, say weeks, months or years. The raw data that is used does however contain strong seasonality: there is an early morning rush hour when people tend to go to work and an evening rush hour when people return to their homes. See Figure 5.

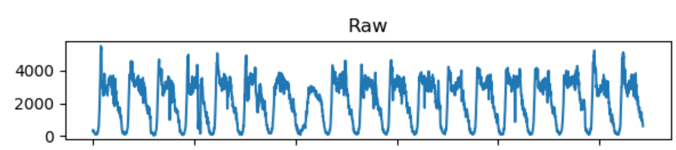


Fig. 5: Raw traffic data, vehicle flow rate per 10 minutes for 19 days at the A4 near Delft

In our case the data is already reasonably stationary: the expectation and variance are almost constant over time. We are therefore able to do ARIMA predictions without differencing the data.

The data is however differenced in some cases, this is explicitly mentioned. In these cases, the differencing is used to remove the seasonality.

To remove the seasonality in our traffic data, the interval parameter i is set to the size of a season. The raw traffic data used in this paper is collected per ten minutes, there are $24 * 60/10 = 144$ ten minutes in a day so parameter i equals 144. After the differencing the data does not contain any seasons and the stationary assumption is stronger. The data now looks like Figure 6.

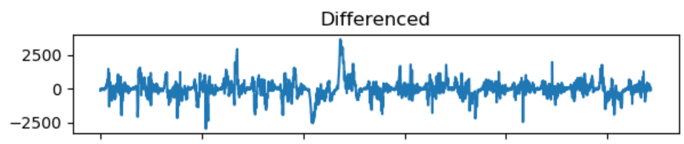


Fig. 6: Differenced traffic data, vehicle flow rate per 10 minutes for 19 days at the A4 near Delft

2) Defining the problem - Vectorizing: The next step is to create samples and targets out of the time series. Some of the predictions will make use of **multivariate data**, however the input layer of our models consist of a 1d tensor (a vector) containing the features. So our samples have to be vectors too.

When transforming the data into vectors two important parameters come into play, the **input length** and the **sample skip**. The input length is the amount of time steps that are included in the sample. The sample skip is the amount of time steps difference between the samples and the corresponding targets, this determines the amount of time the model predicts in the future.

Figure 7 shows this process with an input length of 2 and a sample skip of 2. The values of time step 1 and 2 from the multivariate time series (2 time steps: input length 2) are used to predict the value of time step 4 from the univariate time series (4 - 2 = 2: sample skip 2). The raw data is now transformed in usable samples and targets that reflect the problem.

multivariate ts	vfr	A	B	C	D
	avs	E	F	G	H
univariate ts	vfr	A	B	C	D
	time step	1	2	3	4
samples	vfr				A
	avs				E
	vfr				B
	avs				F
targets	vfr				D
	time step				4

Fig. 7: Vectorization process

There are only univariate predictions made with ARIMA, e.g. the samples and targets were made out of the same time series.

3) Train preparation - Normalizing: The normalization of the samples, subtracting the mean and dividing by the standard deviation, is only done for data used by ANNs. All features have to be normalized independently to have a mean of 0 and a standard deviation of 1.

Normalizing the data ensures that the values that are feed into the neural network do not have wildly varying ranges, most values should be in the 0-1 range. It also ensures that the features to predict are take values in roughly the same range, i.e. made homogeneous. This are both important characteristics that help to prevent large gradient updates such that the network is able to converge. Figure 8 shows the differencing and normalization in one graph.

4) Train preparation - Splitting: The final step in the data preparation is the splitting of the data in a training and test set. The samples and targets from the training set are used to train the model. The samples from the test set are used to make the prediction and the targets from the test set are used as the truth to compare the predictions with.

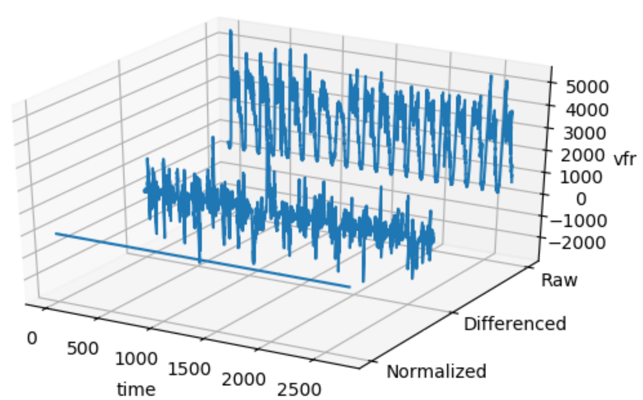


Fig. 8: Raw, Differenced and Normalized differenced traffic data, vehicle flow rate per 10 minutes for 19 days at the A4 near Delft

D. Training

1) Folding: To be able to reliably evaluate the model during the training, k-fold cross-validation is used. K identical models are initialized and trained with different training and validation data.

The training and validation data are retrieved from the train set of data. It is used to validate the model during the training. The test set of data is only used for testing and not for training or validation during the training. The test set of data (test samples and test targets) is the set of data that is used to make predictions and to calculate the errors.

The folding process works as follows, the data (training samples and training targets) is split into k partitions. Every ”fold” k - 1 partitions are used to train the model of the fold and 1 partition is used to validate this model. After the folds have been ”folded”, the average validation score of these models then taken as the validation score of the model and can be used to analyze the training process. Figure 9 shows an example of a 3 fold cross-validation.

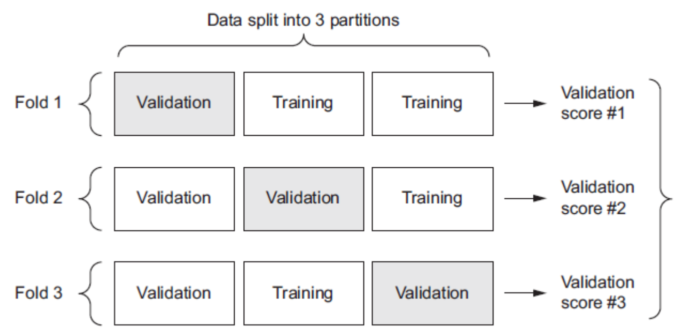


Fig. 9: Folding process[9]

2) Model: As explained in the concepts section a two layered FNN model is used to make the predictions. The network consists of 1d tensors.

The amount of neurons per layer is as follows:

- input layer: input length * dimension of the data[10]
- hidden layer: 64 if not explicitly specified
- output layer: 1

The only variable here is the amount of neurons in the hidden layer. The hidden layer and the output layer are dense layers that work the following way: output = activation(dot(input, weights) + bias). Where bias is a bias vector that is able to change the mapping of the activation function. Which can improve the learning.

The model parameters where chosen on their usage. They are all widely used for regression problems and it provides a nice base for the experiments. Parameters

- activation function: rectified linear unit
- optimization function: rmsprop,
- loss function: mean squared error

The batch size for training the model is chosen to be 1, this resulted from some testing. The batch size that delivered the best results was chosen. The same holds for the number of neurons.

3) Early stopping: Early stopping is the action of stopping the training when the amount of change in the monitored quantity is less then a specified value. The number of epochs that a model will run is therefore determined by this action. The training of the models used for this study are stopped when the amount of change in the valuation loss is zero. In

[9]Figure retrieved from Deep Learning with Python [18].

[10]1 for univariate data and 2 for multivariate data.

practice this happens somewhere before 500 epochs. Seldom all epochs (5000) were used.

E. Prediction and analysis

1) Predicting: The predicting it self relatively straightforward. In the case of an ANN, the test inputs are provided to the trained model and the prediction is calculated. In the case of an ARIMA model, there can only be one value predicted at the time. After this prediction the model has to be trained again to be able to predict the next time step. The use of an ARIMA model takes therefore more effort and needs to be maintained while an ANN model can be used without the need for further training or adjustment.

2) Inverse differencing: The predicted values are inversed differenced when the input of the model was differenced before the prediction. It brings the seasonality back into the data, so that the model predicts the actual targets and not the differenced version of the targets. The full differencing process with an example can be found in the concepts section.

3) Analysis: For the analysis the same metrics as defined in the "Concepts - Metrics" section are used except the mean squared error.

V. Results

The plots in this section refer to the data in the following way: MV multivariate, UV univariate, UP upstream, LO local (does not include upstream data), DIF differenced.

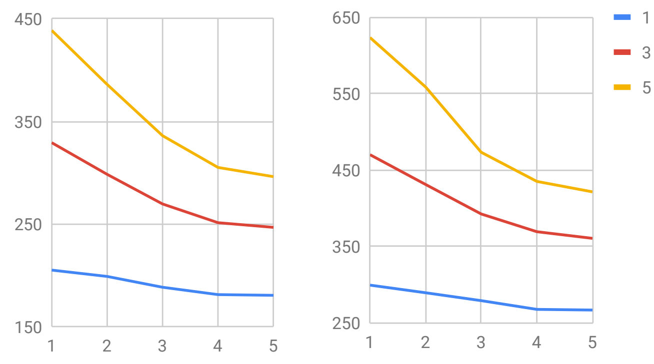
A. Input length

One of the variables that can be used to possibly improve the predictions is the input length, i.e. the amount of samples used for the prediction. One of the questions that rises is the following: **What are the effects of increasing the input length on our prediction capabilities?** or more specifically: To what extend does the increase of input length improve the predictions, or does it not improve the predictions at all?

To get a feeling for this question we first have to look at the time interval between time steps. In this paper, the time in between two time steps is 10 minutes. Since it concerns traffic vehicle flow and average vehicle speed, it is expected that the improvement of the prediction will stop. The speed of a vehicle that passed the measurement site one day ago will probably tell you nothing about a vehicle that currently passes the measurement site. It is however more likely that in some cases, the speed of a car that passed the measurement site 20 minutes ago does give you some information about the vehicle that is currently passing by.

In order to find an answer to this question, all the variables except the input length and the sample skip have been frozen. Firstly ANNs containing 64 neurons have been trained with samples of different input lengths ranging from 1 to 5. This is done with multivariate samples that include upstream data from the Junction Kethelplein network.

Figure 10 shows that both the MAE and RMSE agree that the improvement of the predictions stops after the input length has reached a length of 4 or 5, i.e. when 40/50 minutes of data is used to predict the next 10 minutes. It is interesting to see how the input length relates to the sample skip size. Increasing the input length has bigger effects when the amount of time steps that is predicted in the future is larger.

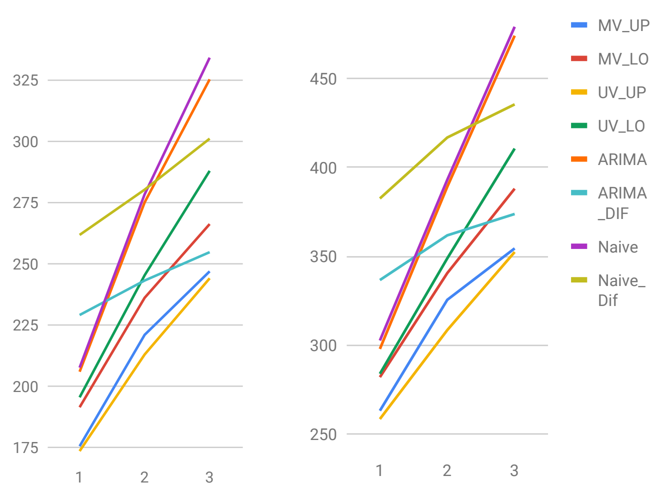


(a) Mean absolute error (b) Root mean squared error

Fig. 10: Increasing input length, 64 neurons

B. Sample skip

With the answering of the above posed questions, new questions raise. We now know that there is an "optimal" input length after which the error stops decreasing. However we do not know yet what the effects of the sample skip size are on the prediction capabilities of the different models. i.e. Does the order of best predicting models change when the sample skip size is increased?



(a) Mean absolute error (b) Root mean squared error

Fig. 11: Increasing sample skip, 64 neurons

Figure 11[11] shows a plot of the errors of all models, pre-

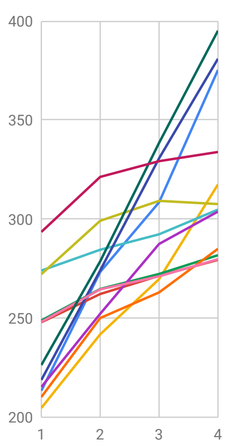
[11] ARIMA is fitted with $p = 1$ and $q = 3$, this resulted from ACF and PACF analysis after which the surrounding values, e.g. in the case of $q = 3$: 1, 2, 4, and 5, where tested to confirm the right fit.

dicting the next time step (1 sample skip), up to and including 3 time steps in the future (3 sample skips). The forecasts are made on the Junction Kethelplein network with input length 3. The plot also includes a naive (NAIVE) implementation where simply the samples where used as a prediction.

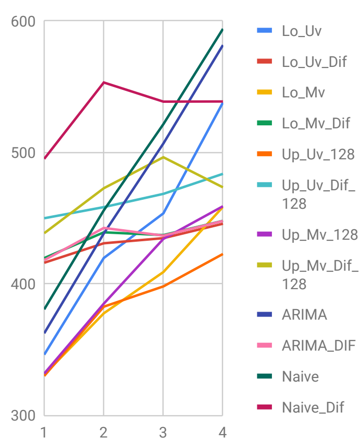
In this figure we can see that the error of the ANN based models increases at a higher pace than the error of the deseasonalized ARIMA model (ARIMA_DIF).

The question now is: does the deseasonalized ARIMA model outperform the ANNs at higher sample skips?
Before the deseasonalized ARIMA model is fitted, the data is differenced. This is not the case for the ANNs used so far. Differencing the data on the one hand, removes seasonality from the data and thereby removes the possibility for a neural network to learn the seasonality, but on the other hand it also removes a changing component and should therefore make the training simpler since there is less to learn. Therefore it seemed reasonable to include a ANN that is trained with differenced samples on differenced targets.

Figure 12, contains the predictions on the Delft A4 network up to and including 4 time steps in the future. The ANNs with deseasonalized data are included. Here we see similar results.



(a) Mean absolute error



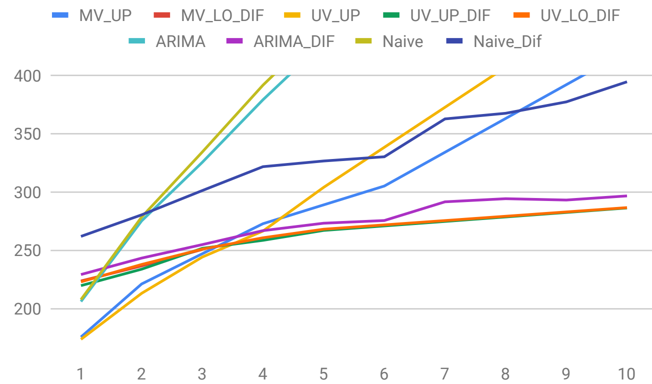
(b) Root mean squared error

Fig. 12: Increasing sample skip, 64 neurons

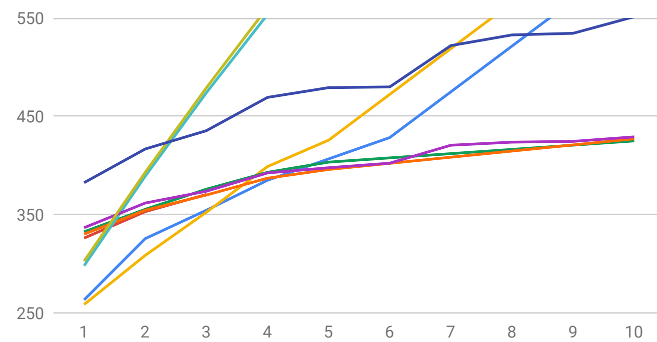
Taken all the previous in account we plot the interesting models from 1 sample skip upto 10 sample skips[12]:

Figure 13 shows that a model using univariate and upstream data gives the best predictions up a sample skip of 4. From there on the models that use deseasonalized data outperform the other models. There seems to be a slight advantage for the neural networks although the difference is marginal.

[12]Both UV_UP and UV_UP_DIF are plotted with values from sample skips 1, 2, 3, 4, 5 and 10.



(a) Mean absolute error



(b) Root mean squared error

Fig. 13: Increasing sample skip, 64 neurons

C. Multivariate data

This paper focuses on different ways to predict traffic, in specific the vehicle flow rate, the rate at which vehicles pass a measurement point in terms of vehicles per hour. One way of improving the predictions could be the usage of features other than the vehicle flow rate to predict the traffic. In this paper the term multivariate data refers to the usage of both the vehicle flow rate and the average vehicle speed in kilometers an hour. The research question formulated to address this possible improvement is as follows: **What are the effects of using multivariate time series instead of univariate time series?**

To answer this question, we can have a look at the figures that were introduced earlier. Figure 11 and 12 show that in both networks multivariate outperforms univariate only when local data is used. This means that the usage of multivariate data in this form, i.e. by flattening it into a 1d tensor, does not increase the performance and only adds complexity to the learning process.

D. Upstream data

One of the sub questions posed to answer the research questions is the following: **What are the effects of using upstream data in downstream predictions?** The test done on both networks give the same answers to this question, namely it improves the predictions. All of the tests and figures show that neural networks using upstream data have the smallest prediction error. As noted before this changes after a sample skip of 3. Quick testing showed that using all the measurement sites in the networks (both in the direction of the target measurement site and the opposite measurement site) does not improve the results, only the measurement sites in the same direction (as used above) improve the results.

It is likely that also in this case more data is not always better. The measurement sites used in this research lie relatively close to another, it would be interesting to see to what distance the measurement sites add value to the predictions. This is however left for future research.

VI. Future Work

This study tries to explore the boundaries regarding to what is possible with a two layered neural network in the field of traffic forecasting. There are however a lot of things left to explore. Regarding the tests done in this paper, it would be nice to see if the same results hold for time series that cover 1 minute or 5 minutes of data per time step. On top of that it would be nice to see to what distance from the target measurement site adding a measurement site, i.e. increasing the amount of measurement sites in the upstream prediction, adds to the prediction accuracy.

Regarding the usage of multivariate data, it would be interesting to see if it does improve the predictions if for example a CNN is used, so that the flattening of the data is unnecessary.

VII. Conclusion

The testing results show that, when doing predictions using time steps covering 10 minutes of traffic data, the optimal amount of samples used as input is 4. Increasing the input length after this does not result in better predictions, it even slightly increases the prediction errors.

Moreover, it became clear that up to 4 or 5 time steps forecasting in the future, the neural networks using upstream data outperform both the ARIMA model with seasons and the differenced ARIMA model. After this the differenced ARIMA model is might be the better choice. The ANNs using differenced data are able to match the results from the differenced ARIMA model, but they take a lot more time to train.

The data also show that the usage of upstream data almost always decreases the prediction errors. The neural networks with the smallest prediction errors are the neural networks that use upstream data.

We see a different result with the usage of multivariate data. We cannot draw any conclusions other than that the usage of multivariate data by flattening a multivariate time series into a 1d tensor does not reduce the prediction errors. This might be different with for time series with smaller time steps. It could however also be that the average vehicle speed is not a good predictor for the vehicle flow rate. Although linear regression analysis shows that with a p value of less than 0.001 that the usage of the average vehicle speed could add significant information to the model. It certainly not excludes the option that it does add information to the model.

References

[1] CORDIS, "Fifth world congress on intelligent transport systems," 1998.

[2] Ministerie van Infrastructuur en Waterstaat, "Kerncijfers mobiliteit 2018," 2018.

[3] P. J. Brockwell, R. A. Davis, and M. V. Calder, *Introduction to time series and forecasting*, vol. 2. Springer, 2002.

[4] B. M. Williams, "Multivariate vehicular traffic flow prediction: evaluation of arimax modeling," *Transportation Research Record*, vol. 1776, no. 1, pp. 194–200, 2001.

[5] Y. Kamarianakis and P. Prastacos, "Forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches," *Transportation Research Record*, vol. 1857, no. 1, pp. 74–84, 2003.

[6] B. Ghosh, B. Basu, and M. O'Mahony, "Multivariate short-term traffic flow forecasting using time-series analysis," *IEEE transactions on intelligent transportation systems*, vol. 10, no. 2, p. 246, 2009.

[7] K. Chakraborty, K. Mehrotra, C. K. Mohan, and S. Ranka, "Forecasting the behavior of multivariate time series using neural networks," *Neural networks*, vol. 5, no. 6, pp. 961–970, 1992.

[8] H. Raman and N. Sunilkumar, "Multivariate modelling of water resources time series using artificial neural networks," *Hydrological Sciences Journal*, vol. 40, no. 2, pp. 145–163, 1995.

[9] K. Kumar, M. Parida, and V. Katiyar, "Short term traffic flow prediction for a non urban highway using artificial neural network," *Procedia-Social and Behavioral Sciences*, vol. 104, pp. 755–764, 2013.

[10] B. Çetiner, M. Sari, and O. Borat, "A neural network based traffic-flow prediction model," *Mathematical and Computational Applications*, vol. 15, no. 2, pp. 269–278, 2010.

[11] S. Ishak, P. Kotha, and C. Alecsandru, "Optimization of dynamic neural network performance for short-term traffic prediction," *Transportation Research Record*, vol. 1836, no. 1, pp. 45–56, 2003.

[12] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[13] S.-Y. Yun, S. Namkoong, J.-H. Rho, S.-W. Shin, and J.-U. Choi, "A performance evaluation of neural network models in traffic volume forecasting," *Mathematical and Computer Modelling*, vol. 27, no. 9-11, pp. 293–310, 1998.

[14] H. Dia, "An object-oriented neural network approach to short-term traffic forecasting," *European Journal of Operational Research*, vol. 131, no. 2, pp. 253–261, 2001.

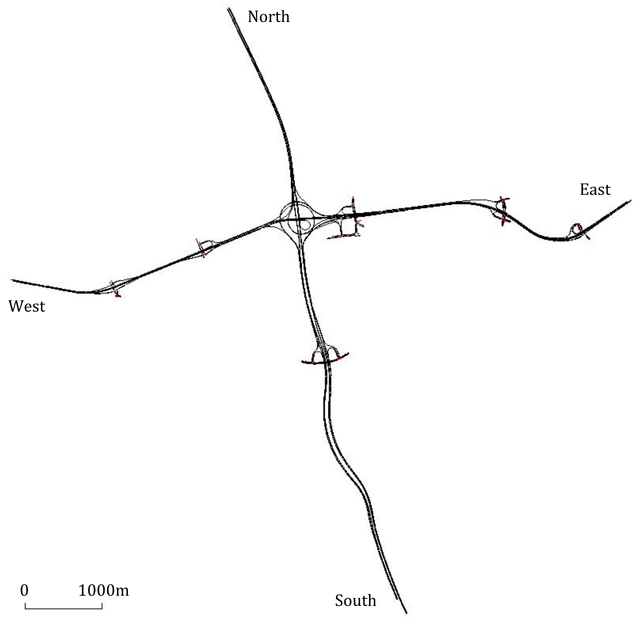
[15] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, 2017.

[16] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining kohonen maps with arima time series models to forecast traffic flow," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307–318, 1996.

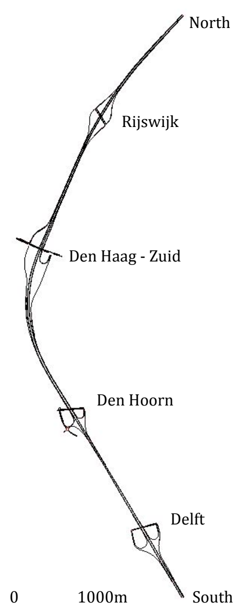
[17] W. Zheng, D.-H. Lee, and Q. Shi, "Short-term freeway traffic flow prediction: Bayesian combined neural network approach," *Journal of transportation engineering*, vol. 132, no. 2, pp. 114–121, 2006.

[18] F. Chollet, *Deep Learning with Python*. Greenwich, CT, USA: Manning Publications Co., 1st ed., 2017.

Appendix A: Networks



Network Junction Kethelplein



Network A4 Delft

APPENDIX B: MEASUREMENT SITES

Junction Kethelplein

Ramp name	Road direction	Ramp type	Number of lanes	NDW measurement site id(s)	Hectometre pole	Id
North	North	Off	3	RWS01_MONIBAS_0041hrl0606ra	60.7	North_S0
	South	On	2	RWS01_MONIBAS_0041hrr0606ra	60.6	North_N1
East	East	Off	3	RWS01_MONIBAS_0201hrr0281ra	28.1	East_E0
	West	On	3	RWS01_MONIBAS_0201hrl0280ra	28.1	East_W1
South	North	On	5	RWS01_MONIBAS_0041hrl0751ra	75.1	South_N1*
				RWS01_MONIBAS_0041hrl0751rb	75.1	
	South	Off	5	RWS01_MONIBAS_0041hrr0752ra	75.2	South_S0*
				RWS01_MONIBAS_0041hrr0752rb	75.2	
West	East	On	2	RWS01_MONIBAS_0201hrr0197ra	19.7	West_E1
	West	Off	2	RWS01_MONIBAS_0201hrl0197ra	19.7	West_W0

A4 Delft

Ramp name	Road direction	Ramp type	Number of lanes	NDW measurement site id(s)	Hectometre pole	Id
North	North	Off	3	RWS01_MONIBAS_0041hrl0502ra	50.3	North_N0
	South	On	3	RWS01_MONIBAS_0041hrr0504ra	50.4	North_S1
Rijswijk	North	On	1	GEO0B_R_RWSTI357035	51.1	Rijswijk_N1
		Off	1	RWS01_MONIBAS_0040vwc0518ra	51.8	Rijswijk_N0
	South	On	1	GEO0B_R_RWSTI357028	51.7	Rijswijk_S1
		Off	1	RWS01_MONIBAS_0040vwa0511ra	51.1	Rijswijk_S0
Den_Haag-Zuid	North	On	2	GEO0B_R_RWSTI357011	52.9	Den_Haag-Zuid_N1
		Off	1	RWS01_MONIBAS_0040vwc0541ra	54.1	Den_Haag-Zuid_N0
	South	On	1	GEO0B_R_RWSTI43	53.7	Den_Haag-Zuid_S1
		Off	4	RWS01_MONIBAS_0040vwa0529ra	52.9	Den_Haag-Zuid_S0
Den_Hoorn	North	On	-	-	-	Den_Hoorn_N1**
		Off	-	-	-	Den_Hoorn_N0**
	South	On	-	-	-	Den_Hoorn_S1**
		Off	-	-	-	Den_Hoorn_S0**
Delft	North	On	1	RWS01_MONIBAS_0040vwd0568ra	56.9	Delft_N1
		Off	1	RWS01_MONIBAS_0040vwc0573ra	57.3	Delft_N0
	South	On	1	RWS01_MONIBAS_0040vwb0573ra	57.3	Delft_S1
		Off	1	RWS01_MONIBAS_0040vwa0568ra	56.9	Delft_S0
South	North	On	3	RWS01_MONIBAS_0041hrl0579ra	58.0	South_N1
	South	Off	2	RWS01_MONIBAS_0041hrr0579ra	58.0	South_S0

* The data from these measurement sites are merged
** Do not have corresponding NDW measurement sites