ETA Prediction for Vessels using Machine Learning

Edwin Flapper
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
e.t.flapper@student.utwente.nl

ABSTRACT

Vessels entering and leaving a port follow a route consisting of waypoints. When a vessel is at one of these waypoints the vessel traffic service system needs to know the estimated time of arrival to the next waypoints. This research compares a variation of machine learning models in order to find the model best suited for short term vessel ETA prediction.

Keywords

Estimated Time of Arrival, Machine Learning, Maritime, SVM, KNN, Gradient Boosting

1. INTRODUCTION

Vessels arrive and leave ports many times a day. This number is growing with the increasing amount of transportation of goods, however the capacity of a port is limited. This capacity is largely limited by the amount of vessels that can be in a waterway at the same time and the number of available docks for loading and unloading vessels[6].

When a vessel arrives at, or leaves, a port they are assigned a route from/to one of the docks. This route consists of multiple waypoints within the port. Upon assignment of the route the Estimated Time of Arrival (ETA) is given for each of the waypoints, including the final destination. The ETA between two waypoints is currently calculated once a day based on the Actual Time of Arrival (ATA) of the previous days. The problem with this method of calculating the ETA is that this ETA will be the same regardless of the vessel type, time of day, draught or any other variables. The result is an often inaccurate ETA prediction.

An inaccurate ETA prediction results in delays in the time it will take to unload an arriving container ship. Based on the predicted ETA of the vessel to the dock manpower will be sent to that dock to help with the mooring of the ship and unloading and/or loading the ship. However if the ship arrives before the ETA the ship may have to wait until the dock is ready for mooring, potentially blocking part of the waterway. On the other side if the ship arrives after the ETA the dock remains unavailable for other ships while not being unused. By having an accurate ETA prediction there is as little delay as possible and the process of unloading and loading will take less time resulting in less time the dock is unavailable for other vessels. Because of this more ships will be able to deliver or collect their loads [6]. Another example where the ETA is important is for vessels that require towboats. Towboats are sent to vessels to help them manoeuvre through the waterways. Based on the predicted ETA of the vessel a towboat is sent for help. Again if the vessel arrives too early it needs to wait for the towboat, resulting in a vessel potentially blocking part of the waterway and delays in the whole process. If the vessel arrives too late the towboat needs to wait while being unavailable for other vessels that may need help.

The difficulty of ETA prediction lies in the many variables that can influence the time it takes for the vessel to arrive. Because there are so many variables the solution of looking at the ATA's of the previous days and using that to calculate one ETA that will be the same for every vessel that day is too inaccurate.

To address the problem above, this research compares multiple machine learning models, using different machine learning methods, which are trained with historical data in order to predict the travel time of vessels.

1.1 Research questions

The goal of this research is to answer the following research questions:

- How can the use of machine learning improve the Estimated Time of Arrival prediction for vessels?
- Among the wide range of machine learning models, is there any particular method which can typically improve the ETA prediction performance?

2. RELATED WORK

There have been many studies in the field of ETA prediction with the use of machine learning [10][3][9][7][1]. The most similar research done is by I. Parolas[7]. One of the key differences with this research is the time window of the prediction. This research focuses on ships that are in or about to enter the port. While in the research done by I. Parolas the vessel was at sea possibly multiple days away from the destination. The data used was GPS data combined with weather predictions. Because of the smaller time frame the data used for this research needs to be more accurate. This is done by using data from a vessel traffic service system.

In the studies done on ETA prediction the models that often performed well were: Gradient boosting, Support Vector Machines, Long Short Term Memory and Kalman Filtering [1][10][4][3][9]. Kalman Filtering was mostly used in

33rd Twente Student Conference on IT 2020, Enschede, The Netherlands.

combination with other machine learning methods. These ensemble models consisting of multiple machine learning methods often out performed single method models [3][9][10]. In these ensemble models often consisted of a method for learning from historical data and a method for learning about the current situation. This way the model can find trends through the historical data and adjust these findings to the situation at that moment, which is useful when for example an accident has occurred and a route is blocked off.

In addition models consisting of Neural Networks often performed the best but requires much more data compared to other models [10][4]. In the study by Z. Wang, M. Liang and D. Delahaye[9] data clustering had a positive result on the predictions of the models, DBSCAN outperformed K-means++ as the method of clustering.

3. PRELIMINARY

In this section the preliminaries for the various machine learning methods used are given.

3.1 Support Vector Machine

Support Vector Machines (SVM)[5] considers each data point into a multi-dimensional space. The dimension of this space is equal to the number of features of the data set. The SVM then tries to create a hyperplane in the space that separates the classes of the data. If this is not possible the SVM adds dimensions to the data space in such a way that it becomes possible. The data is not actually transformed to a higher dimension but it is processed as if it was in this higher dimension, this is called the kernel trick. SVM's are often used for classification of data but can be used for regression as well. When used for regression they are often called Support Vector Regression (SVR)[2].

3.2 Gradient Boosting

Gradient Boosting[11] uses a combination of multiple decision trees for classification or regression. The decision trees are created such that each decision tree improves the prediction of the previous decision tree based on the biggest mistakes that previous decision tree makes. This way each subsequent decision tree further improves the prediction until the maximum number of trees is reached or no improvements can be made based on the training data.

3.3 K-Nearest Neighbours

K-Nearest Neighbours (KNN)[8] plots each data point into a multi-dimensional space, similar to the Support Vector Machine. KNN however uses these data points when a new set of data arrives. A new data point is placed into the same space and then the nearest points are used to determine the result of the new data. K determines at how many nearest points the algorithm looks. If K is 1 the result will be the result of the nearest point. If K is 10 then the nearest 10 points are used to determine the result.

4. METHODOLOGY

In this section the different steps taken during the research are discussed.

4.1 Data sets

For this research a total of 3 data sets are used. Each of the three data sets is confidential, meaning neither details on the structure or actual data of the original data sets will be mentioned in this paper. The first data set used is a test data set. This test data set consists of simulated data based on real data. This test data is good enough for understanding what potential features are available and creating test models. However, since the data within the data set is simulated and contains repeated simulations of roughly the same tracks, this test data set can not be used to test the effectiveness of the different machine learning methods. The repeated simulations allows for learning this repetition which would not be the case for real data.

The second and third data sets consist of real data gathered from one port each. These two data sets became available during the research, data set 2 about one third and data set 3 about two thirds into the research. All three data sets are SQL databases with very similar structure. Whenever a vessel arrives or leaves the port it is assigned a route containing waypoints. When this route is created the ETA to each of the waypoints is calculated by the system. This route with the waypoints is then stored in the port management part of the database. A separate system tracks the vessels movement in the port and stores this track, again with waypoints, in its own part of the database. In this case the route of the vessel is the assigned path the vessel should follow and the track consists of the actual positions where the vessel was. The port management database then updates the stored route with data from the track.

The result is two different sub-databases, the port management part which we will call PM and the vessel tracking part we will call VT. Both the PM and the VT part contain similar data after the vessel finishes its route. There are however a few key differences. First of all the PM contains the current systems predicted ETA while the VT does not. On the other hand the VT contains a vessels speed and direction at each of the waypoints while the PM does not.

In Table 1 an overview of the sizes and time frames of the different data sets can be found. Here 2 PM rec and 3 PM rec contains only recent data, past 3 years, instead of the entire data set. As can be seen in the Table 1 the 1 VT data set is much larger in size because most of it is simulated data.

At the start of this research the VT data set was mostly used. This was because the VT data sets include the vessels speed and direction at each of the waypoints. However, as only the PM data sets include the current systems ETA prediction, by using the VT data set there is no possibility of comparing the result of the new models with the result of the current system.

Data set	Start	End	Tracks	Waypoints
1 VT	2019-07	2020-04	1.277.956	4.603.659
1 PM	2019-07	2020-04	23.868	184.772
2 VT	2017-04	2020-04	211.656	470.100
2 PM	2010-01	2020-04	69.477	690.187
2 PM rec	2017-04	2020-04	18.420	177.350
3 VT	2019-05	2020-04	39.732	145.805
3 PM	2012-10	2020-04	61.239	622.235
3 PM rec	2017-04	2020-04	16.595	195.501

Table 1. Table containing the different data sets used with the start and end date of the time frame and the number of tracks and waypoints.

When data set 3 became available the change was made from using the VT data set to the PM data set. This is partially due to the before mentioned reason of comparing to the current systems performance. The main reason however is that in this new data set the VT part for about two out of three tracks does not contain a reference to a vessel. Without this reference it is not possible to obtain vessel details, such as depth, width or length.

4.2 Preprocessing

Before the data can be used as input into the models it first needs to be reprocessed. This section contains the different steps taken for obtaining a use-able data set. While taking a look at the different steps we will take data set 3 PM as an example of what the effects are. The total number of tracks in data set 3 PM is 61.239.

The first step is to filter out old data. Only data from the past 3 years will be used. As the data becomes older it becomes less relevant to current situations. For example if a new dock is added the route to that dock may become busier increasing the time a vessel takes to travel along that route. Small changes like this over a long period of time can drastically change the travel times of vessels in certain situations. Because of this learning from older data may worsen the models. When only taking tracks of the previous 3 years the number of tracks becomes 16.595.

The data from the different data sets do contain missing data as well. An example of this is a track that consists of certain waypoints, however some of the waypoints might be missing. To fix this problem we will remove all tracks that contain at least one waypoint without a waypoint ID. After applying this filter to all the tracks there are 13.911 tracks remaining.

For the next step we will take a look at the distribution of the number of waypoints each track contains. This distribution can be found in Table 2. There are 355 tracks that only contain one waypoint. The goal of the model is to predict the travel time between waypoints, however if there is only one waypoint this cannot be done. This means all tracks that contain only one waypoint are removed resulting in 13.556 remaining tracks. For improving the accuracy of the models we will be using two starting waypoints instead of only one. This will be explained in the next section, Feature Selection. This does require each track to have at least two starting waypoints and one target waypoint, in total at least 3 waypoints in a track. After removing the 2153 tracks containing only two waypoints there are 11.391 tracks remaining.

Waypoints per Track	Count
1	355
2	2153
3	930
4	1590
5	2013
6	1871
7	1990
8	2037
9	971
10	0
11	1

Table 2. Table containing the distribution of waypoints per track.

In addition to the track data we will also use vessel details as features for the models. In order to use track data in combination with vessel data each track requires a link to a vessel. Again this link is sometimes missing meaning those tracks with a missing link to a vessel need to be removed. For the example data set this means a total of 4471 tracks need to be removed. The result is 6920 remaining tracks.

Sometimes when a track has a link to a vessel that vessel has some missing details that are needed as features for the models. In this case a total of 115 tracks have a link to a vessel with missing details. After removing these there are 6805 tracks remaining.

Using the remaining tracks, for each track the first and second waypoint in the route are used as the starting waypoints. Then each of the next waypoints in the route are used as the target waypoint to which the ETA is to be predicted. For example a track with a route containing 7 waypoints will result in 5 data points each using the first two waypoints as the starting waypoints and each having a different target waypoint.

Each of the features used for the model is then scaled such that the mean value for that feature is 0 and the standard deviation is 1. This is done to normalise the data and help speed up some of the models.

Finally the data set is split into a training and test set. This is done based on the timestamp of each data point, the training set contains the oldest 80 percent of the data and the test set contains the 20 most recent percent. Since the data set is from the previous 3 years, if we assume the number of data points is uniform over time, this means that the last 0.6 years of data is used for testing and everything before is used for training the models.

4.3 Feature selection

The current ETA prediction system often has inaccurate predictions because it makes the same prediction regardless of time of the day and vessel type. By using machine learning we can use the time and vessel details to improve the prediction of the models. In addition to using vessel details and the time the models will receive three waypoints. The first waypoint is the current waypoint where the vessel is when the prediction is made. The second waypoint is the previous waypoint, the waypoint the vessel visited before the current waypoint. The third is the target waypoint to which the travel time will be predicted.

By using three waypoints instead of only two, current and target waypoint, it becomes possible to add the distance and travel time between the current and previous waypoint as features for the models. With these features the model is able to take into account the speed over the last segment for its prediction. When comparing the models a comparison between using two or three waypoints as input is present as well.

For the input of the models the following features are used:

- Time:
  - Month
  - Day
  - Hour
- Vessel details:

  - Width
  - Length
  - Depth
  - Type
- Previous waypoint:
  - ID
  - Longitude
  - Latitude
  - Distance to current waypoint
  - Travel time to current waypoint
- Current waypoint:
  - ID
  - Longitude
  - Latitude
  - Distance to target waypoint
- Target waypoint
  - ID
  - Longitude
  - Latitude

This results in a total of 19 features. For Gradient Boosting this higher amount of features is not a problem as for each decision tree it selects the best feature to use. For the SVM on the other hand, already being the slowest model among the 3, may become slower as for each new feature an additional dimension is needed.

Feature	Rank
Month	19
Day	15
Hour	17
Width	10
Length	6
Type	18
Depth	14
Previous id	8
Latitude previous	13
Longitude previous	11
Distance from previous	9
Time from previous	5
Current id	16
Latitude current	12
Longitude current	7
Target id	4
Latitude target	2
Longitude target	3
Distance target	1

Table 3. Table containing the features used ranked by their importance according to the Gradient Boosting model.

In order to reduce the number of features, the Gradient Boosting model can be used to order the features by their importance. This ranking is shown in Table 3. In this case the lower ranked features are more important to the model. When comparing the results of the different models using all 19 features is compared to only using the 10 most important features.

4.4 Parameter tuning

In order to find the best parameters for each of the models a grid-search is performed on the different combinations of parameters. First for each parameter of a model a list of possible values was created. Then for each possible combination of parameters the model is tested. This is done using 5-fold cross validation.

For the SVR model the best parameter combination uses the Radial Basis Function (RBF) kernel function in combination with a regularization parameter of 1000, epsilon of 50 and a gamma of 0.01. The Gradient Boosting model performs best by using a learning rate of 0.1 in combination with a max depth of 3 for the individual trees. The KNN model performs best when using a K, number of neighbours to look at, of 11. Furthermore the KNN model has the option of using the distance to the nearest neighbours when determining the result, meaning the data points that are more similar become more important. The other option is to not use this distance and instead uniformly use the nearest neighbours in predicting the result. Both options performed similarly in the grid-search. Because of these similar results both models are used when comparing the different models.

4.5 Comparison

In order to compare the different models each model is first trained on the training data set. After the training is complete the models receive the test data set and make their predictions. These predictions are compared with the actual travel times and the differences are the errors. The Root Mean Squared Error (RMSE) is then calculated for each of these models. The RMSE is obtained by taking the root of the Mean Squared Error (MSE). The MSE is obtained by taking the mean after squaring each error. In addition to the RMSE the errors are also shown in a graph in order to see the distribution of the errors.

5. RESULTS

This section contains the results of comparing the different models. This includes the comparison between the RMSE for each model and graphs showing the error distributions of the different models used.

Table 4 shows a comparison between the different models with the achieved RMSE and the running time necessary to train and make the predictions. These results show that

Model	RMSE	Time in Minutes
SVR	9.48	1.0957
GB	8.84	0.0603
KNN dist	10.12	0.0164
KNN uni	10.29	0.0135
Current	72.55	-

Table 4. Table containing the RMSE and execution time for each model. Includes the current systems RMSE.

the Gradient Boosting model achieved the lowest RMSE followed by the SVR model. The SVR model however required much more time to train especially compared to the KNN models. The current systems RMSE is included as well and is at least 7 times as high as the new models. This can largely be explained by a few predictions of the current system that are almost a year off. Because of this large error of a few predictions the RMSE becomes much higher as well.

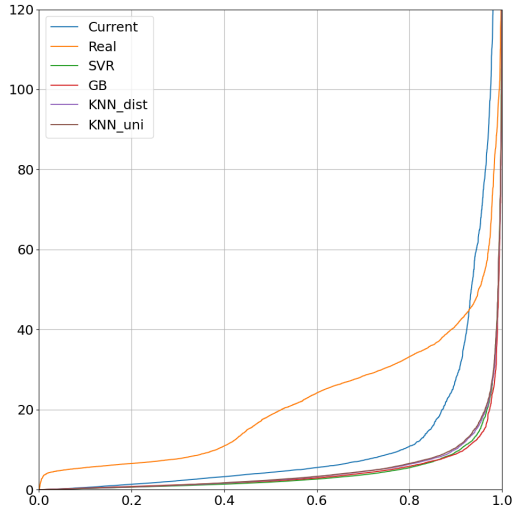

Figure 1. Model error distribution comparison using all 19 features and 2 starting waypoints. The y-axis is the error in minutes and the x-axis is the percentage of the data set after sorting by error.

Figure 1 shows the error distribution for each of the models when using all 19 features and 2 starting waypoints. Each model has the errors resulting from the predictions made during testing sorted and this is shown in the figure. For each model the lowest errors are shown on the left and increase along the x-axis. The y-axis is the error in minutes and the x-axis is the percentage of the data set. For example the area between x = 0 and x = 0.2 contains the 20% lowest errors for each model. The yellow line labeled "Real" contains the actual travel time distribution. The blue line labeled "Current" is the current systems error distribution. The other lines show the error distribution for the different models with KNN_dist using the distance to the neighbours and KNN_uni being uniform over all neighbours.

Figure 2 shows the same graph as in Figure 1, however the y-axis is now a logarithmic scaling. By changing the y-axis to be logarithmic the difference between the models becomes more visible. One clear conclusion that can be drawn from this figure is that the different models always seem to have lower errors compared to the current system, the blue line.

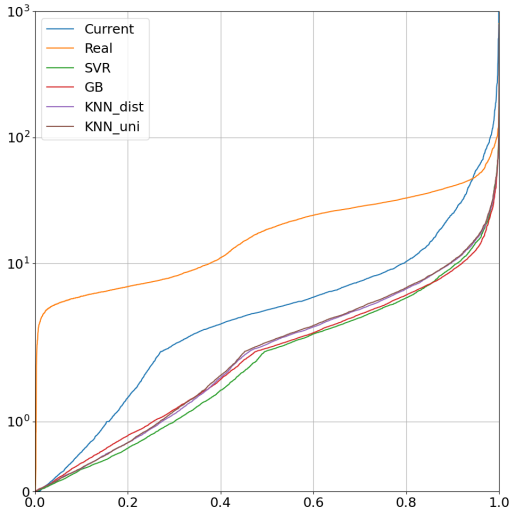

Figure 2. Model error distribution comparison using all 19 features and 2 starting waypoints with logarithmic scaling on the y-axis. The y-axis is the error in minutes and the x-axis is the percentage of the data set after sorting by error.

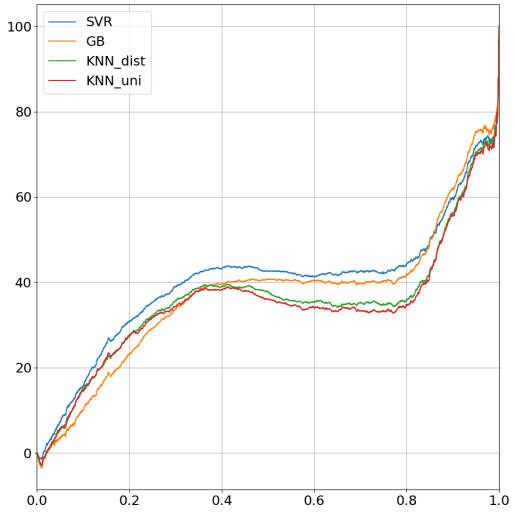

Figure 3. Comparison between the new models and the current system. The y-axis is how many percent the new models error is lower compared to the current system. The x-axis is the percentage of the data set after sorting by error.

In Figure 3 this conclusion is confirmed. This graph shows how many percent each models errors are lower compared to the current systems errors. Only at the first 1 percent the current system has slightly lower errors. At this point the error of each model and the current system is close to 0 minutes meaning that this slight difference is at most a few seconds. At around x = 0.5, the middle of the error distribution, each model has between 35% and 42% lower errors and after x = 0.9 this goes up to about 75%. This is a large improvement where instead of an error of 40 minutes the error becomes only 10 minutes when using one of the machine learning models. When comparing the different machine learning models the figure shows that the SVR model performs the best from x = 0 to about x = 0.85 after which the Gradient Boosting model performs

the best.

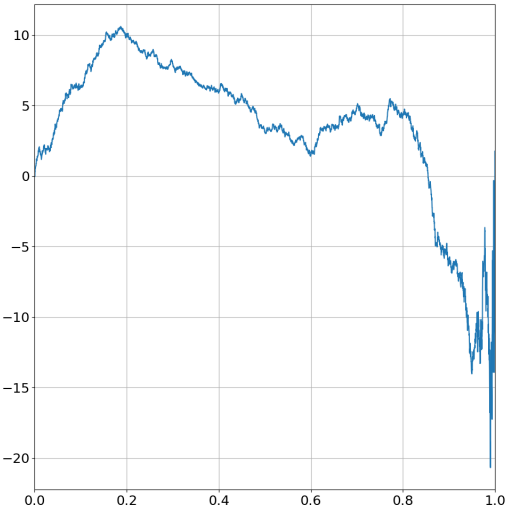

Figure 4. Comparison between the SVR and Gradient Boosting models. The y-axis is how many percent the SVR model's error is lower compared to the Gradient Boosting model. The x-axis is the percentage of the data set after sorting by error.

Figure 4 shows that this is indeed true with the SVR model achieving around 5% lower errors compared to the Gradient Boosting model from x = 0 to about x = 0.85. After x = 0.85 the SVR has higher errors.

Model	RMSE	Time in Minutes
SVR	10.35	0.7015
GB	9.33	0.0370
KNN dist	10.12	0.0051
KNN uni	9.921	0.0049

Table 5. Table containing the RMSE and execution time for each model when only using the 10 most important features.

The Table 5 shows the RMSE and execution time when only using the 10 most important features according to the Table 3. Compared to using all features the RMSE is slightly higher for the SVR and Gradient Boosting models, while for the KNN uni there was a slight improvement. The time required for training was also much lower compared to the time required when using all 19 features.

Figure 5 shows the same graph as Figure 2 except with the error distribution obtained from only using 10 features. The KNN models now perform better compared to the SVR and Gradient Boosting models for the first 80% after which the models perform similar again.

Table 6 contains the RMSE and execution time when using all features but only one starting waypoint. Compared to using two starting waypoints the RMSE is now much higher for all models. The Gradient Boosting model does still perform the best out of all models.

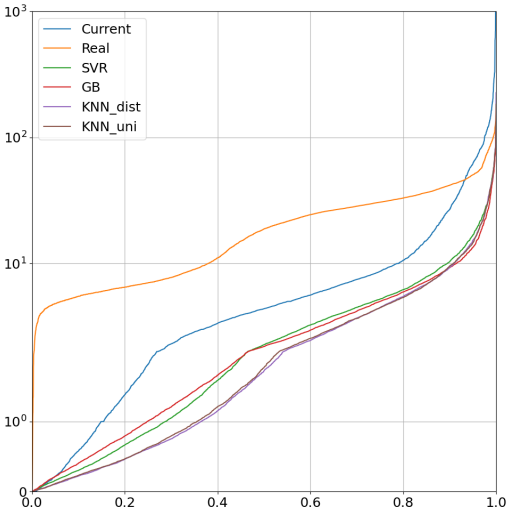

Figure 5. Model error distribution comparison using 10 features and 2 starting waypoints. The y-axis is the error in minutes with logarithmic scaling and the x-axis is the percentage of the data set after sorting by error.

Model	RMSE	Time in Minutes
SVR	17.51	1.0291
GB	16.37	0.0538
KNN dist	17.05	0.0126
KNN uni	17.11	0.0124

Table 6. Table containing the RMSE and execution time for each model when only using 1 starting waypoint.

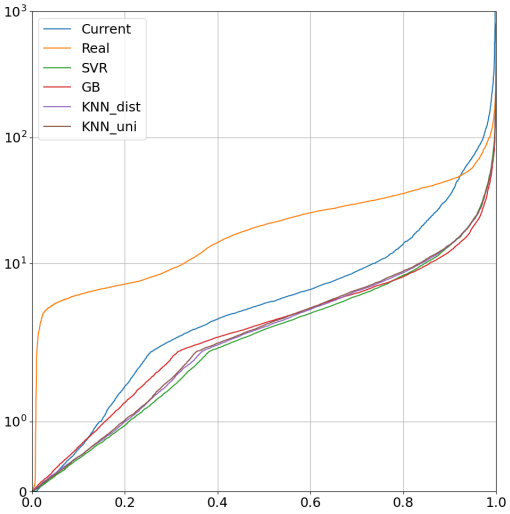

Figure 6. Model error distribution comparison using all features and 1 starting waypoints. The y-axis is the error in minutes with logarithmic scaling and the x-axis is the percentage of the data set after sorting by error.

Figure 6 shows the error distribution when only using one starting waypoints. In this figure the machine learning models do consistently have lower errors compared to the

current system, however the difference is much lower now compared to before.

6. CONCLUSION

The results show that machine learning can definitely be used to improve ETA prediction. Out of the three methods tested in this research Gradient Boosting performed the best with the lowest RMSE while having reasonable time required for both training and making a prediction. The SVR model performed good as well, even out performing the Gradient Boosting model for about 85% of all predictions, however the Gradient Boosting model performed better when the errors became larger as 10 minutes. Furthermore in order to achieve the best results all features should be used, including two starting waypoints. This does increase the training time of the models slightly which can become important when using even more features or larger data sets.

There is a lot of future work that can be done for this research. This includes trying out different Machine Leaning methods such as deep learning as well as ensemble machines that combine two or more Machine Learning methods. One example for such an ensemble model that was used in literature was the combination of a model learned from historical data such as a SVR together with a Kalman Filter that adjusts the prediction of the SVR according to the current situation in the port.

7. ACKNOWLEDGEMENT

First of all I would like to thank my supervisor Decebal Mocanu for supervising my research. I would also like to thank Nicola Strisciuglio as the track chair together with Decebal Mocanu.

Finally a big thanks to Saab Technologies Apeldoorn for providing the data required to do this research as well as knowledge in the field of port management.

8. REFERENCES

[1] S. Ayhan, P. Costas, and H. Samet. Predicting estimated time of arrival for commercial flights. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 33–42, 2018.

[2] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik. Support vector regression machines. In *Advances in neural information processing systems*, pages 155–161, 1997.

[3] A. Gal, A. Mandelbaum, F. Schnitzler, A. Senderovich, and M. Weidlich. Traveling time prediction in scheduled transportation with journey segments. *Information Systems*, 64:266–280, 2017.

[4] V. Kumar, B. A. Kumar, L. Vanajakshi, and S. C. Subramanian. Comparison of model based and machine learning approaches for bus arrival time prediction. In *Proceedings of the 93rd Annual Meeting*, pages 14–2518. Transportation Research Board, 2014.

[5] W. S. Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[6] C. Pani. Managing vessel arrival uncertainty in container terminals: a machine learning approach. 2014.

[7] I. Parolas. Eta prediction for containerships at the port of rotterdam using machine learning techniques. 2016.

[8] L. E. Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

[9] Z. Wang, M. Liang, and D. Delahaye. A hybrid machine learning model for short-term estimated time of arrival prediction in terminal manoeuvring area. *Transportation Research Part C: Emerging Technologies*, 95:280–294, 2018.

[10] B. Yu, Z.-Z. Yang, K. Chen, and B. Yu. Hybrid model for prediction of bus arrival times at next station. *Journal of Advanced Transportation*, 44(3):193–204, 2010.

[11] Y. Zhang and A. Haghani. A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies*, 58:308–324, 2015.