Degree Thesis
Computer Engineering, 180 credits

Analyzing public transport delays using Machine Learning

Degree Project in Computer Engineering, 15 credits

Alexander Hirvonen, Marcus Robertsson
Abstract

Delays is a big factor when considering taking the public transportation or taking your own car. If delays were more predictable, more people would take the bus instead. This thesis results can be used to further develop more robust systems for predicting delays, thus, more people using the public transportation systems.

This was done in collaboration with Hogia. Hogia is a company in Sweden that have their own solutions for calculating delays within public transportation. This thesis investigates if predictions using Machine Learning can improve Hogia’s predictions on bus delays. Python and various libraries are used for training and testing the Machine Learning model.

The data available for this study was gathered and provided by Hogia. Raw data were analyzed and preprocessed to create and find features in it, and then used to train a Random Forest Regressor. The model’s predictions are analyzed with various measurements and then compared against their current solution, as well as the actual delays.

The result of this study looks promising since only a small dataset of 30 days was used. Also, it gives an understanding of what features that can be of value when training a model. Even though the model’s predictions were in some cases far off compared to Hogia’s current solution due to outliers in the data, this study can be used for further research of utilizing Machine Learning for predicting delays.
Dictionary

**Bus stop** – A location where the bus stops and passengers embark and disembarks.

**Journey** – A collection of bus stops that are connected through the timetable.

**Delay** – When a bus arrives late to a bus stop. (Higher than 0 seconds)
# Table of Content

Abstract ................................................................................................................... i

Dictionary .................................................................................................................. ii

Table of Figures........................................................................................................ v

Acronym ....................................................................................................................... vi

1 Introduction ........................................................................................................... 1
   1.1 Purpose ........................................................................................................... 1
   1.2 Limitations ...................................................................................................... 1
   1.3 Specifications ................................................................................................ 1

2 Background ............................................................................................................ 3
   2.1 Previous research .......................................................................................... 3
   2.2 Current methods ............................................................................................ 3
   2.3 Machine Learning .......................................................................................... 3
      2.3.1 Features and labels ................................................................................. 3
      2.3.2 Classification, Regression and Clustering ............................................... 4
      2.3.3 Supervised vs Unsupervised Machine Learning ........................................ 5
      2.3.4 Decision trees ......................................................................................... 6
      2.3.5 Random Forest ....................................................................................... 7
      2.3.6 Cross validation ..................................................................................... 8
   2.4 Python, Pandas and Scikit-Learn ....................................................................... 8
   2.5 Conclusions .................................................................................................... 9

3 Method ................................................................................................................ 11
   3.1 Data ............................................................................................................... 11
      3.1.1 Data storage overview ............................................................................. 11
      3.1.2 Preprocessing ......................................................................................... 11
   3.2 Features- and label selection ......................................................................... 11
      3.2.1 Dwell Time ............................................................................................. 12
      3.2.2 Arrival and departure differences ......................................................... 13
      3.2.3 Difference between stops ...................................................................... 13
      3.2.4 Passenger data ....................................................................................... 13
      3.2.5 Analyzing features ............................................................................... 13
   3.3 Analyzing the result ........................................................................................ 14
      3.3.1 Mean Absolute Error ............................................................................ 14
      3.3.2 Mean Squared Error ............................................................................. 14
      3.3.3 Median Absolute Error ........................................................................ 15
      3.3.4 Coefficient of Determination (R²-score) ................................................ 15
      3.3.5 Heatmap ................................................................................................ 15
   3.4 Machine learning ............................................................................................ 15
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4.1 Hyperparameter optimization</td>
<td>16</td>
</tr>
<tr>
<td>3.4.2 Training</td>
<td>16</td>
</tr>
<tr>
<td>3.4.3 Testing and evaluating</td>
<td>16</td>
</tr>
<tr>
<td>4    Result</td>
<td>17</td>
</tr>
<tr>
<td>4.1 Evaluating the data</td>
<td>17</td>
</tr>
<tr>
<td>4.2 Evaluating ML Model</td>
<td>21</td>
</tr>
<tr>
<td>4.2.1 Hyperparameter tuning for Random Forest</td>
<td>22</td>
</tr>
<tr>
<td>4.2.2 Feature importance</td>
<td>22</td>
</tr>
<tr>
<td>4.2.3 Mean Absolute Error (MAE)</td>
<td>23</td>
</tr>
<tr>
<td>4.2.4 Mean Squared Error (MSE)</td>
<td>23</td>
</tr>
<tr>
<td>4.2.5 Median Absolute Error (MAD)</td>
<td>23</td>
</tr>
<tr>
<td>4.2.6 R² Score</td>
<td>24</td>
</tr>
<tr>
<td>4.2.7 Standard Deviation</td>
<td>25</td>
</tr>
<tr>
<td>5    Discussion</td>
<td>27</td>
</tr>
<tr>
<td>5.1 Future work</td>
<td>28</td>
</tr>
<tr>
<td>Schedule</td>
<td>30</td>
</tr>
<tr>
<td>References</td>
<td>31</td>
</tr>
<tr>
<td>Figure Number</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>2-1</td>
<td>Illustration of features and labels.</td>
</tr>
<tr>
<td>2-2</td>
<td>Mapping from string to integer.</td>
</tr>
<tr>
<td>2-3</td>
<td>Classification vs Regression.</td>
</tr>
<tr>
<td>2-4</td>
<td>Clustering.</td>
</tr>
<tr>
<td>2-5</td>
<td>Branches of Machine learning.</td>
</tr>
<tr>
<td>2-6</td>
<td>Decision tree example.</td>
</tr>
<tr>
<td>2-7</td>
<td>Example data.</td>
</tr>
<tr>
<td>2-8</td>
<td>Decision tree structure.</td>
</tr>
<tr>
<td>2-9</td>
<td>Random Forest.</td>
</tr>
<tr>
<td>2-10</td>
<td>Cross validation.</td>
</tr>
<tr>
<td>3-1</td>
<td>Overview of data collecting.</td>
</tr>
<tr>
<td>3-2</td>
<td>Illustration of dwell time calculation.</td>
</tr>
<tr>
<td>3-3</td>
<td>Illustration of arrival difference calculation.</td>
</tr>
<tr>
<td>3-4</td>
<td>Example of scatter plot.</td>
</tr>
<tr>
<td>4-1</td>
<td>Variance in delays.</td>
</tr>
<tr>
<td>4-2</td>
<td>Observed delayed vs Hogia predictions.</td>
</tr>
<tr>
<td>4-3</td>
<td>Number of passengers – On time vs delayed.</td>
</tr>
<tr>
<td>4-4</td>
<td>Dwell time – On time vs delayed.</td>
</tr>
<tr>
<td>4-5</td>
<td>Bus arrivals – Delayed vs in time.</td>
</tr>
<tr>
<td>4-6</td>
<td>Bus arrivals – Delayed vs in time.</td>
</tr>
<tr>
<td>4-7</td>
<td>Mean delay per hour.</td>
</tr>
<tr>
<td>4-8</td>
<td>Heatmap of correlations between features.</td>
</tr>
<tr>
<td>4-9</td>
<td>MAE for unseen data.</td>
</tr>
<tr>
<td>4-10</td>
<td>MSE for unseen data.</td>
</tr>
<tr>
<td>4-11</td>
<td>MAD for unseen data.</td>
</tr>
<tr>
<td>4-12</td>
<td>$R^2$-score for unseen data.</td>
</tr>
<tr>
<td>4-13</td>
<td>Standard Deviation ML.</td>
</tr>
<tr>
<td>4-14</td>
<td>Standard Deviation Comparison.</td>
</tr>
<tr>
<td>5-1</td>
<td>Comparing RF, Linear, and Ridge regression algorithms.</td>
</tr>
</tbody>
</table>
Acronym
ANN - Artificial neural network
API – Application Programming Interface
COD - Coefficient of Determination
CV – Cross-Validation
DBS - Difference between stops
DF – Data Frame (Pandas)
ETA – Estimated time of Arrival
GPS – Global Positioning System
GSCV – Grid Search Cross-Validation
HPTS – Hogia Public Transport Systems AB
MAD – Median Absolute Error
MAE – Mean Absolute Error
ML – Machine Learning
MSE – Mean Squared Error
RF – Random Forest
RFR – Random Forest Regressor
RSCV – Randomized Search Cross-Validation
SML – Supervised Machine Learning
SVM - Support vector machines
UML – Unsupervised Machine Learning
1 Introduction
The government of Sweden have set a goal for 2025 that 25% of all transport should be done by public transport (Eneroth, et al., 2018). In a report (Näringsdepartementet, 2003) they have examined that people are 9-19 times more annoyed with delays rather than longer routes. Therefore, it’s an interesting field to research why the delays in public transport happens and if there are cheap and easy ways for more accurate predictions.

Improving public transportation delay predictions is key to increase the number of people using it. If the bus arrives 10 minutes late, the people waiting for that bus will soon lose faith in public transportation systems and use their own car/their vehicle of choice. That is why this field of study is so important. This paper will look further into understanding in how to develop better systems for predicting delays for public transportation vehicles.

1.1 Purpose
The purpose of this study is to analyze better ways to predict delays in public transportation systems using historical data from a specific route and Machine Learning (ML).

The following questions will be tried to be answered:

- Is it possible to improve the current predictions that Hogia currently have using ML?
- What are considered valuable features when predicting bus delays based on the data Hogia provides (3.2.5)

1.2 Limitations
Due to the complexity of this area, this study will be focusing on a single route. This chosen route must follow important criteria’s such as: covers a larger area, a lot of bus stops along the route, the capability of the bus of recording useful data such as GPS and passengers onboard.

Also, GPS positions recorded from this route will not be used because it is updated irregularly. GPS positions are updated every 1-9 minute and therefore, can’t be used as a trusted feature in the model.

1.3 Specifications
The outcome of this project is a result that examines if the data that is available in Hogia’s database is enough to produce a ML model capable of improved forecasting which will be compared to what Hogia currently uses.

The model should be able to predict the forecast on each bus stop within a given route.
2 Background
There have been plenty of studies done in this field where researches have adopted various techniques to solve the important and complex problem of making accurate public transportation forecasts. Historical and real-time data approaches are the most widely known, there have been a few using machine learning techniques (Artificial neural network (ANN), Support vector machines (SVM) and Random Forest (RF)).

2.1 Previous research
In a study (Garcia & Retamar, 2016) they have investigated a specific machine learning algorithm (Extremely randomized trees) and GPS positions from buses to predict the time it will take between bus stops. Their results are promising with between 90-100% accuracy. They used supervised learning because they had a lot of test data that they knew the result of. Why they choose Extremely Randomized Trees was because of a study (Rich Caruana, 2006) where they concluded that Gradient Boosted Trees, Random Forests, Neural Networks, and Support Vector Machines have high predictive accuracies. So, they choose an ensemble of randomized decision trees also known as Random Forest and had a high success rate.

According to a study conducted (Bertini & El-Geneidy, 2004) on the metropolitan area of Portland, almost one third of the total travel time is lost due to serve bus stops. With 16% of the total time being from actual dwell time.

2.2 Current methods
Current solutions that Hogia uses for bus delays prediction is based on historical data and statistical methods. The ETA for each stop is updated in real-time and keeps the waiting passengers informed when the bus will arrive to a given stop. When a bus arrives to a stop, it sends a message to the system that it has arrived. This also happens when the bus leaves, it will send out a message that it has left this station with a timestamp when it happened. Then an algorithm will be used to calculate how late the bus will arrive to the next stop and update the information board at the upcoming stops.

2.3 Machine Learning
ML is a term used within data science as the ability of a computer to learn from historical data without being explicitly programmed. The more historical data available, the better the computer can learn and predict future values from that data. Correlations between input and output variables is what makes ML possible, the weaker the correlation is the worse the model will perform on new observed data fed to the model. Therefore, a high understanding of the dataset is crucial for obtaining a high performing ML model.

There are multiple learning methods in ML: Supervised, Unsupervised, Semi-Supervised and Reinforcement learning.

2.3.1 Features and labels
A dataset is split into features and labels, also known as input and output. See Figure 2-1.

![Figure 2-1 – Illustration of features and labels](image-url)
Where $X_1, X_2, \ldots, X_{n-1}$ and $X_n$ is the features of the dataset and $y$ is the observed value, also called a label. The features can be seen as the information for the ML model to make its predictions on. Therefore, selecting which features to include in the dataset is key for a successful model. Labels can be seen as the answer or the observed value of the prediction, it’s what the model tries to estimate or guess.

ML is statistical analysis using matrix calculations. Therefore, all the features and labels must be of the type integer or float. E.g., if the feature vector contains any string values, they need to be mapped to either an integer or float before it is fed to the algorithm, see Figure 2-2.

<table>
<thead>
<tr>
<th>Feature</th>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>11</td>
<td>Göteborg</td>
<td>0.15</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>Malmö</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>Göteborg</td>
<td>0.47</td>
</tr>
</tbody>
</table>

In Figure 2-2, the feature vector $X$ contains 3 columns where $X_2$ is of type strings and $X_1, X_3$ is of type Integer. The string values of $X_2$ is mapped to integers and replaced in the dataset.

To choose which one of these to use in this study, the uses of each needs to be discussed. Classification is used to classify input $x$ into one of two or more classes ($Y$). For example:

2.3.2 Classification, Regression and Clustering
Classification and Regression are two branches in Supervised ML. They are used for different purposes.

Classification is used when the output data is categorical. E.g., Figure 2-3 if a christmas tree should be chopped down or not, it classifies the input as either 1 or 0, which is mapped to yes or no.
Regression is used when predicting continuous values such as house prices Figure 2-3 and is either an integer or a floating-point value. Where the goal is to create a function that best predicts the housing price from the observed data in the dataset.

<table>
<thead>
<tr>
<th>Features (Christmas trees)</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diameter (m)</td>
<td>Height (m)</td>
</tr>
<tr>
<td>0.02</td>
<td>0.5</td>
</tr>
<tr>
<td>0.07</td>
<td>1.7</td>
</tr>
<tr>
<td>0.1</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Features (House prices)</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rooms</td>
<td>Sqm</td>
</tr>
<tr>
<td>5</td>
<td>235</td>
</tr>
<tr>
<td>4</td>
<td>145</td>
</tr>
<tr>
<td>10</td>
<td>1027</td>
</tr>
</tbody>
</table>

Figure 2-3 – Classification vs Regression

Clustering is a way to find meaningful groups in a dataset without having or knowing the labels. That is being shown in Figure 2-4, the input features are fed to the algorithm and the output is a set of clusters or groups where the algorithm sees correlations in the data.

2.3.3 Supervised vs Unsupervised Machine Learning

Supervised and unsupervised are two different ways to train a model. Supervised have both features and labels to train on whereas unsupervised only have features. Usage depends on what data is available, if labels are available in the dataset before training the model, supervised learning is preferred, see Figure 2-5.
SML can be described as a supervisor watching over the process when teaching a model by giving it the correct answers $y$ for the current input vector $X$. The model will learn from both input and output. E.g., given the function:

$$y = f(X)$$

*Equation 2-1 - ML function*

The goal is to train the function $f$ that takes a vector $X$ and predicts a value $y$. From this, the model can take unknown values and predict an answer. (Equation 2-1)

SML can be done using classification or regression, where the key differences are described above.

Unsupervised learning is used when there are no known answers. What the algorithm tries to answer is a grouping problem where it is looking for similarities in the dataset. What the user needs to set is how many groups should be used, then the ML algorithm splits the data and organize it into that number of groups. E.g., if there are three different colored balls with the colors red, orange and green and the number of groups is set to two. Then the two groups would possible consist of one group with red and orange and the second group of green, this is a possible outcome because red and orange have more similarities between themselves than to green. Because of this it’s hard to evaluate the result of a UML.

### 2.3.4 Decision trees

A decision tree in its simpler forms can be described as Figure 2-6. The more popular decision tree algorithms are ID3, C4.5 and CART. The key differences between the three is that they are using different algorithms to decide when to split a node.

According to (Gupta, et al., 2017), C4.5 and CART is better than ID3 when there are missing values in the dataset, while ID3 was better at producing faster results.

A tree is built by recursive binary splitting, which is when all the features are considered, and different split points are tried and tested by a cost function. The function with the best cost will be selected for that node.

![Decision tree example](image)

*Figure 2-6 – Decision tree example*

To understand how trees are built and how they work, see Figure 2-7 and Figure 2-8:
Here is a representation of the data in a dataset with features X1 and X2 and the output 1, 2 and 3. Feeding this data to a decision tree classification model will yield the tree in Figure 2-8.

Decision trees are prone to overfitting (high variance) on their own, that is why an ensemble of decision trees are needed to create better predictions, i.e. Random Forest.

2.3.5 Random Forest
Random Forest is one of the most popular supervised ML algorithms. They also have a high-performance value (Fernández-Delgado, et al., 2014). A Random Forest is an ensemble of decision trees and from all these trees, a prediction is generated. Each tree in the forest generates a prediction, the mean of those predictions is the final prediction for the Random Forest, see Figure 2-9.
2.3.6 Cross validation

Cross validation is a technique where the dataset is split between train and test data. The training data is further split into components and then randomized so the model doesn’t become overfit or underfit. Overfitting is when a model that models the training data too well and perhaps only works on that specific data. This can happen if the data that the model trains on is too specific to a certain condition and because of that, only works if that condition is in the dataset. Underfitting is when a model can’t model the training data nor be accurate on the testing data, this can happen because it’s not a good ML algorithm or the data is not suited for that given problem.

As seen in Figure 2-10 the data is first randomized and then split into components, all these components will be used to test and train a model. The first model will be trained on component 1,2,3, and 4 and then it will be tested on component 5. This will be done 5 times, the model that gets the highest score will be used.

2.4 Python, Pandas and Scikit-Learn

Python is a concise high-level programming language that focuses on being easily readable. The language is indent based to minimalize the characters that needs to be written. Python supports many well documented libraries that makes it easy to create complex programs with few lines of code. There are many IDE’s and editors to write Python in. PyCharm is a Python IDE, much like IDE’s for other
coding languages such as Eclipse for Java. PyCharm can handle debugging and more. On the other hand, there is a well-known tool called Jupyter Notebook, which is great for interactive developing and presenting data science projects.

Pandas (pandas, 2018) is a library that can be imported to Python, it focuses on manipulating and analysis of data. The mainframe that the data is stored in is called a DF and in that frame, data can be manipulated, pivoted, reshaped or filtrated etc. It’s a widely used library for data analysts.

Scikit-learn (Scikit-Learn, 2018) is a library used for ML, it’s widely used and documented. It is very easy to implement and test different ML algorithms. Scikit-learn is also called sklearn for short.

2.5 Conclusions
As seen in previous research (2.1), Random Forest generated a result of high accuracy. Because of this, and in agreement with the supervisors of this study, Random Forest will be used.

This project will be a regression problem because what this study is aiming to estimate is the real delay in seconds, not if the bus will be late or not (classification). Also, the answer is known in the testing data and therefore supervised learning is used.
3 Method

3.1 Data
Major part of this study is about collecting data and preprocessing it. This is to enable a model to be trained using ML with historical data. The data used in this study origins from Hogia’s client’s buses and are stored on Hogia’s servers in their PubTrans system, which is a system that processes real time data. Due to the complexity of the area, a single bus route was chosen between two bigger cities in Sweden. Covering a total distance of 20 km with 33 stops and 160 journeys back and forth each day, with a little less during the weekends.

Usually Hogia stores predicted forecasts in their systems for only a few days. But to train a ML model, much more data will be needed. Therefore, Hogia staged a staging environment that stores data for a longer period of times. From this test environment a total of 30 days of data was collected.

3.1.1 Data storage overview
Forecasts and passengers’ onboard origins from different database.

![Figure 3.1 – Overview of data collecting](image)

Figure 3.1 illustrates how the data is gathered using Python.

3.1.2 Preprocessing
Processing the data before feeding it to train a ML model is crucial. The quality of the ML model depends on the quality of the data. Therefore, preprocessing was needed in order to create additional reliable/relevant data.

Raw data were collected from the servers and then processed in Python using Pandas. A function was created that took a single day of raw data and created a dataset of features and labels (3.2). All days were then merged into a complete dataset, holding 30 days’ worth of features and labels.

3.2 Features- and label selection
The features extracted from the dataset is covered in this section. Some features were found in the data whereas other were created, all the features were used when training the model. See Table 3-1.
The difference between observed- and targeted time is that targeted is when the bus should arrive at a stop, whereas observed is the actual observed time for arrival at a stop. Observed and timetable time had to be converted from datetime to seconds. E.g., 19:00 is converted like the following:

$$19 \times 60 \times 60 = 68,400$$

This will show how many seconds since the beginning of the day has passed.

Hogia’s predictions include their predictions of how late a bus will arrive to a given stop. Their predictions are continuously updated between stops and are therefore updated more often than the rest of the data.

- Day of the week: Mapping between Monday-Sunday, 0-6.
- Station number: Number of a bus stop, 1-33.
- Rush hour: Splitting the day into two categories. 13-16 is considered rush hour and is then set to 1.
- Weekend: If day is Saturday or Sunday, weekend is set to 1

State is what kind of a message it is. It contains a value between 0:5 which is mapped from [Arrival, Departure, Passed, AtStop, Expected, Missed]. All messages except expected are updated when a bus arrives and departures from a bus stop. If a bus doesn’t stop at a bus stop, a passed message is sent. Expected messages contains up-to-date information about when a bus will arrive.

Labels for the dataset is the arrival difference for the next station. E.g. a prediction is made at bus stop 20, the label will be how late the bus will arrive to bus stop 21.

### 3.2.1 Dwell Time

Dwell time is the time a bus stands still at a given bus stop. This is a function of seconds (s), where s equals the timestamp of the bus. The following equation $Y_{dwell}$ is used for calculating dwell time:

![Dwell Time Diagram](Image)

$Y_{dwell}$

Figure 3-2 - Illustration of dwell time calculation
In Figure 3-2 the green dot is when the bus arrives, and the red is when the bus departs from the
stop. $A(s)$ and $D(s)$ equals arrival/departure timestamps and Equation 3-1 equals to the dwell time at a
specific stop, given in seconds.

$$Y_{\text{dwell}}(s) = D(s) - A(s)$$

*Equation 3-1 - Dwell time*

### 3.2.2 Arrival and departure differences

Arrival and departure differences are calculated using a function of $s$, where $s$ is the timestamp of the
bus when it arrives or departs from the stop:

$$Y_{\text{arr}\_\text{diff}}(s) = A_0(s) - A_t(s)$$

$$Y_{\text{dep}\_\text{diff}}(s) = D_0(s) - D_t(s)$$

*Equation 3-2 - Arrival and Departure differences*

To get the Arrival difference for arrivals, see Equation 3-2, which is the difference in timestamps
between when a bus arrives to a station and when it should be arrival according to the timetable, the
calculation is shown above. The departure difference is calculated in the same way as illustrated in
Figure 3-3.

### 3.2.3 Difference between stops

Difference between stops (DBS) shows if the bus is gaining or loosing time between the stops, this is
calculated by the equation below.

$$\text{DBS} = Y_{\text{arr}t_{i-1}} - Y_{\text{dep}t_i}$$

*Equation 3-3 - Difference between stops*

To get the DBS (Equation 3-3) the calculation is done by taking the $Y_{\text{arr}\_\text{diff}}$ of the current station and
subtracting it by the $Y_{\text{dep}\_\text{diff}}$ of the previous station, see Equation 3-2. If the value is positive the bus
is losing time and is getting later, and if it is negative its gaining time.

### 3.2.4 Passenger data

Passenger data is being collected on every station if the bus is equipped with the prerequisite
equipment. If not all the rows are filled with 0.

### 3.2.5 Analyzing features

In Scikit Learn there is a function called *feature_importances*. This will generate a list of all the
feature the model was trained with and rate each feature from 0-1 (the higher, the more importance it
had). This will give a better understanding of which features plays an important role when the Random
Forest Regressor (RFR) is trained.
First the model is trained with the features. Then the function feature_importances is called and will return the list of features and their ratings. The information retrieved can be used to separate important features from others. An important note for this is that a feature can have high importance when the tree was trained but that doesn’t translate it because a feature had higher importance for training the RFR.

3.3 Analyzing the result
To analyze the result of this project, the machine learning algorithm will be compared to Hogia’s client’s current timetables. The evaluation will be done by multiple statistical calculations that’s an industry standard. These are the evaluation functions this paper are using. Mean absolute error (MAE), mean squared error (MSE), median absolute error (MAD) and coefficient of determination (COD). All these statistical functions are made to evaluate how close a machine learning algorithm is to the actual values this is to be expected.

3.3.1 Mean Absolute Error
The mean absolute error measures how close a set of points on a scatter plot is to the expected line. Figure 3-4 – Example of scatter plot shows two points on a scatter plot; each point has one x and one y value that will be calculated like this $|y_i - x_i|$. The summation of these two calculations, in this example, will be divided by the number of points. (Equation 3-4)

\[
\sum_{i=1}^{n} |y_i - x_i| \over n
\]

Equation 3-4 - Mean Absolute Error Formula

![Example of scatter plot](image)

Best case scenario, y should be equal to x to match the blue line. This shows how far off the predicted values are compared to reality. So, in this case the MAE is 2, which describes that the equation is not perfect for the given problem.

3.3.2 Mean Squared Error
Equation 3-5 is the mathematical equation of MSE where n is the number of predictions, $y_j$ the observed value and $\hat{y}_j$ the predicted value. What this will show is how far from the regression line the predictions are. The difference between this equation and MAE is that MSE penalize errors that are far away from the observed value. So, in combination with MAE this shows the variance in the data.
\[ \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \]

*Equation 3-5 - Mean Squared Error Formula*

### 3.3.3 Median Absolute Error

Equation 3-6 is the mathematical equation of Median Absolute Deviation (MAD) where \( n \) is the number of predictions, \( y_j \) the observed value and \( \hat{y}_j \) the predicted value. The result of this calculation over the dataset will show how accurate the predictions is on the intermediate of the set. This is a robust method to use when the data have outliers because it takes the median of all absolute differences between the target and the prediction.

\[ \text{median}(|y_i - \hat{y}_i|) \]

*Equation 3-6 - Median Absolute Error Formula*

### 3.3.4 Coefficient of Determination (\( R^2 \)-score)

Equation 3-7 is the mathematical equation of \( R^2 \) (Coefficient of determination) where \( n \) is the number of predictions, \( y_j \) the observed value, \( \hat{y}_j \) the predicted value and \( \bar{y}_i \) is the mean of \( y \).

\[ 1 - \frac{\sum_{i=0}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{n}(y_i - \bar{y}_i)^2} \]

*Equation 3-7 - Coefficient of Determination Formula*

The Equation 3-7 shows how much of the total variation is described by the regression line, this is an indicator of how well the ML model will perform with data from the future. The best score is 1 and worst is 0.

### 3.3.5 Heatmap

A heatmap in this paper is used to view the correlations between the features. Correlation is a value between -1 and 1 that represent the linear relationship between two variables. So, in a heatmap this can be seen as a \( x \times x \) grid where \( x \) is the number of variables. Furthermore, in a heatmap the values are represented with a color scheme, so the correlation can be easily examined.

### 3.4 Machine learning

In this research, supervised learning is used. This is because the labels are known and can therefore be used to train the model. The output for the ML model will be continuous values, i.e. bus delays, which can vary a lot. As described in the background, regression is used with continuous values contra classification. If the goal was to classify if a bus will arrive late or in time, classification would be used instead.

The algorithm used is an ensemble of decision trees, also known as, random forest. As seen in (Garcia & Retamar, 2016), an ensemble of decision trees gave the authors of that research a 90-100% accuracy. In addition, a single decision tree is too weak to use on its own, as discussed in (2.3.4).

For this research, Jupyter Notebook (Jupyter, 2014) is a powerful tool. It can visualize the data immediately and DataFrames are stored in memory while working, this will give a very fast workflow where data only need to load once. Compared to PyCharm where data is loaded each time the script is executed, and when there is a lot of data it’s not time efficient.

When training only one model using the complete dataset, the MAD was 25.65 and \( R^2 \)-score 0.88 versus when training a model for each bus stop yielded an MAD of 22.64 with an \( R^2 \)-score of 0.93. Because of this, each bus stop will have its own ML model for better performance.
3.4.1 Hyperparameter optimization

There are two ways to optimize the parameters of a Random Forest (RF) in sklearn, RandomizedSearchCV (RSCV) and GridSearchCV (GSCV) which are inbuilt functions of sklearn’s library. (Scikit-Learn, 2018)

There is a vector for each parameter in the RF:

\[
RF \text{ Parameters} = \\
\{ 'bootstrap': [boolean], \\
     'max\_depth': [int], \\
     'max\_features': [int], \\
     'min\_samples\_leaf': [int], \\
     'min\_samples\_split': [int], \\
     'n\_estimators': [int] \}
\]

Both methods are using cross-validation but uses different amount of iterations. RSCV have an input parameter \((n\_iter)\) where it takes the number of iterations it should make. E.g. If there are 2,400 different combinations of the RF parameters, RSCV will take a random amount of combinations that are set by \(n\_iter\). Whereas GSCV will try all the 2,400 combinations.

3.4.2 Training

The data is loaded into a Jupyter Notebook, where the unwanted data is filtered out. Since each stop got its own ML model, a filter must be applied to the dataset selecting only the given bus stop’s features and the next bus stop’s label.

A Random Forest Regressor (RFR) is created with parameters generated from hyperparameter optimization procedure.

\[
rf = \text{RandomForestRegressor}(\text{max\_depth}, \text{max\_features}, \text{min\_samples\_leaf}, \text{min\_samples\_split}, n\_estimators, \text{random\_state})
\]

The \text{random\_state} parameter takes an integer as input. This is to enable to re-create the same randomness in the shuffling of the dataset when testing different algorithms.

Cross validation (CV) is done by using sklearn’s in-built function \text{cross\_validate}.

\[
cross\_val = \text{cross\_validate}(rf, X, y, cv, n\_jobs, \text{return\_estimator})
\]

Where \(rf\) is the RFR, \(X\) is the features, \(y\) is the labels, \(cv\) is the number of folds (2.3.6), \(n\_jobs\) is the number of jobs to run in parallel and \text{return\_estimator} makes the function return the models used for each fold. In the book Applied Predictive Modelling (Kuhn & Johnson, 2018), the default number of folds is either 5 or 10, but there is no rule in general. Therefore, 5 folds is what will be used in this study. Each run will be compared and evaluated using \(R^2\)-score, the one with the best \(R^2\)-score will be used for each station.

3.4.3 Testing and evaluating

A new dataset will be used to test the RFR created from CV. The dataset is split into \(X\), \(y\) and \text{hogia\_predictions}. A prediction is made using sklearn’s predict() function:

\[
result = rf.\text{predict}(X)
\]

Result is a vector of arrival differences for each prediction made. If 1000 rows were tested on, the result vector would be 1x1000.

The accuracy of the predictions is evaluated with MAE, MSE, MAD and \(R^2\)-score. This shows how accurate the model is, but not how it compares to Hogia’s predictions. A \(R^2\)-score will be calculated on Hogia’s predictions and then compared to the result of the ML model.
4 Result
A total amount of 30 days of data was collected, containing: A single bus route with 3955 journeys covering 33 bus stops, totaling 2,461,455 number of rows and 21 number of columns. Before ML was applied, evaluating the precision of current methods used by Hogia was done.

4.1 Evaluating the data
In this research, a bus is delayed if it arrives a second later or more than the timetabled time. In this section, a better understanding of why buses arrive late will be explored.

Figure 4-1 shows a boxplot of the arrival difference for each bus stop. What this plot shows the spread of the delays with some outliers. What can be seen is that delays increase the further the bus drives on a given journey. There are a few cases of extreme values that could be explained by i.e. traffic accidents.

![Boxplot grouped by station](image)

*Figure 4-1 – Variance in delays*

The delay was calculated for each arrival (3.2.2). This is to avoid using data when the bus was on time or arrived before it was supposed to. The mean value of the delay for each station was plotted as Actual Delay (Figure 4-2). This was compared to Hogia’s predictions, using the same method explained.
Figure 4-2 explains that Hogia’s predictions are accurate but with room for improvement. It shows that their predictions follow the actual delay very well. Another aspect is that the delay increases over the duration of the journey, and that there are some spikes between certain stations. There is a correlation between number of passengers and delay, as seen in Figure 4-3.

Conclusions drawn from Figure 4-3 is that when the number of passengers increase, the more likely it is that the bus is delayed. Also, a pattern can be seen which stations most passengers embark and disembarks from the bus.
Dwell time (3.2.1) is quite constant for each bus stop, with some variance. This means that dwell time will not have as much of an impact on the ML algorithm. As seen in Figure 4-4, the dwell time is sometimes higher when the bus is on time, this may be because it must wait at the station to leave accordingly to the timetable.

Figure 4-5 shows that the bus is more often delayed, than in time. This is probably caused by buses in general is driving under time pressure and any deviation from intended route will result in a delay.
In Figure 4-6, the threshold for a delay was changed to 5 minutes. There is a clear shift in the data, that most of the buses are delayed with less than 5 minutes. But there is still a significant amount of delayed buses above 5 minutes. On weekends it tends to be more on time versus weekdays.

In Figure 4-7, when only checking the delay, the peak between the hours 13 and 16 is when the delays are at a maximum. That is why the rush hour feature was handcrafted, and to help the ML model to predict better.

In the heatmap (Figure 4-8), correlations between all the features are shown. This shows the linear relationship between the features.
Most of the features have correlation between each other in one way or another. Some outliers, i.e. week_end and week_day, have only correlation between each other. These features were thought to be important before evaluating them. But what the data shows, it has very little importance with the rest of the features. What can also be seen is that arrival difference has correlation with multiple features.

4.2 Evaluating ML Model
Each stop has its own ML model, this is to improve the accuracy as explained in (3.4). The relationship between bus stops is not the same throughout the journey. I.e. the amount of delays may differ from stop to stop. That’s why a unique model needs to be created for each stop.
4.2.1 Hyperparameter tuning for Random Forest

Hyperparameter tuning was made using GridSearchCV (GSCV) because it has the most robust method compared to RandomizedSearchCV, since GSCV tests all the parameters. All the vectors were filled with values to be tested accordingly:

\[
RF \text{ Parameters} = \\
\{ 'bootstrap': ['True'], \\
'\text{max_depth}': [10, 20, 30, 35], \\
'\text{max_features}': [4, 5, 6, 7, 8], \\
'\text{min_samples_leaf}': [5, 7, 9, 11], \\
'\text{min_samples_split}': [2, 3, 4], \\
'n\_estimators': [10, 20, 30, 40, 50, 70, 90, 100, 200, 300]\}
\]

This was done on a single bus stop because it took approximately 9 hours to run this test. The parameters given for the best fit for the Random Forest (RF) was:

\[
RF \text{ Parameters} = \\
\{ 'bootstrap': ['True'], \\
'\text{max_depth}': [20], \\
'\text{max_features}': [6], \\
'\text{min_samples_leaf}': [9], \\
'\text{min_samples_split}': [2], \\
'n\_estimators': [60]\}
\]

This increased the R\(^2\)-score on that single bus stop from 0.9715 → 0.9813. This is an increase of the performance. Because the closer the R\(^2\)-score is to 1, the better the model performs.

4.2.2 Feature importance

In Table 4-1 the importance of each feature can be seen. As seen, the most important features for this model is the arrival difference and station number. Whereas the state and if it is a weekend or not have the lowest score, and are therefore, not that important.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Importance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrival Difference</td>
<td>0.879836</td>
</tr>
<tr>
<td>Station number</td>
<td>0.063997</td>
</tr>
<tr>
<td>Observed time</td>
<td>0.013015</td>
</tr>
<tr>
<td>Difference between groups</td>
<td>0.012953</td>
</tr>
<tr>
<td>Dwell time</td>
<td>0.011102</td>
</tr>
<tr>
<td>Targeted time</td>
<td>0.003770</td>
</tr>
<tr>
<td>Passengers onboard</td>
<td>0.003241</td>
</tr>
<tr>
<td>Day of week</td>
<td>0.002309</td>
</tr>
<tr>
<td>Rush hour</td>
<td>0.000442</td>
</tr>
<tr>
<td>Weekend</td>
<td>0.000289</td>
</tr>
<tr>
<td>State</td>
<td>0.000091</td>
</tr>
</tbody>
</table>

Table 4-1 - Feature importances
4.2.3 Mean Absolute Error (MAE)
In Figure 4-9 the MAE for the ML model and Hogia’s predictions can be seen. This illustrates the observed vs predicted values. Greater values correspond to more inaccurate predictions. This plot shows that Hogia’s are in most cases more accurate than the ML model.

![Mean Absolute Error Unseen Data - Model vs Hogia](image)

Figure 4-9 - MAE for unseen data

4.2.4 Mean Squared Error (MSE)
Illustrated in Figure 4-10, the ML model have higher MSE because its outliers are greater than compared to Hogia’s predictions. Because MSE penalizes outliers, the ML models’ outliers are far greater than Hogia’s. This may depend on the quantity of data the model was trained on.

![Mean Squared Error Unseen Data - Model vs Hogia](image)

Figure 4-10 – MSE for unseen data

4.2.5 Median Absolute Error (MAD)
The MAD is less sensitive to outliers and is therefore not as extreme as MSE. Here (Figure 4-11), the ML model compares very well and, in some cases, even better than Hogia’s predictions.
4.2.6 $R^2$ Score

As in contrast to MSE, this is easier to interpret when evaluating the ML model, because it does not depend on the scale of the data. That is why the peaks are correlated on both. The $R^2$-score for Hogia is always better than the ML predictions made (Figure 4-12). Which shows that Hogia’s predictions are more reliable.
4.2.7 Standard Deviation
In Figure 4-13 the standard deviation shows that the variance in the predictions varies from station to station.

Figure 4-13 - Standard Deviation ML

In Figure 4-14, the ML compared to Hogia’s shows that Hogia have a smaller standard deviation with less variations from station to station. The ML model’s variation is due to some outliers in the data.

Figure 4-14 - Standard Deviation Comparison
5 Discussion

The purpose of this study was to examine if better forecasts could be made with the aid of ML.

What can be concluded from the result is that Hogia’s predictions is more accurate than the ML model. As seen in Figure 4-10 and Figure 4-12, our models’ predictions are far off sometimes, which gives extreme errors. This is due to some outliers in the data and due to the relatively small dataset. But in some cases, our model performs equally to Hogia’s, and sometimes even better. If we look at MAD and MAE, the performance of the model is quite accurate. When checking the standard deviation comparison between the ML model and Hogia (Figure 4-14), Hogia’s system out-performs the ML model. Due to some extreme predictions as mentioned above, the overall performance of the ML model drops and makes it unreliable in real-world situations.

Since predictions was made from stop A to B, but only data from stop A was used to teach the Random Forest and disregarded previous stops. This was done because it was too much noise in the data when all stops before A was included to train the model as explained in 3.4. If there is a way to use previous stops data, the accuracy for the predictions would most likely increase. In general, the more correlated data the more accurate models can be created. E.g. if the model sees a pattern early on the bus route with a certain delay, the predictions can be made with better accuracy for bus stops later in the route.

What was considered before starting this research was to use outside variables such as weather information, GPS positions, accidents on the road and more. This was thought to be a great idea until we realized that i.e. weather information was irregularly updated and would increase the complexity of the research. Using GPS positions of all the buses would also increase the complexity and duration of this research. But the main reason behind us not using GPS was because they were updated with an interval of 1-9 minutes to the database, and therefore an unreliable feature. Furthermore on the features, in 4.2.2 the importance of each feature is evaluated and conclusions from this is that state had a very small impact on the model in contrast to arrival difference and station number. This can also be seen in the heatmap (Figure 4-8) where arrival difference has higher correlation in general compared to other features. If more time was available, changes in the features set could be made and tested to see if it would improve the performance of the predictions.

In previous research (Garcia & Retamar, 2016) the authors predicted time taken between stops using GPS with an accuracy of 90-100%. What our study shows compared to this is that the same kind of results can be achieved by not using GPS. We believe that using a mix of both worlds would predict even better.

In hindsight, it became known to us that other regression algorithms would probably perform better on this kind of data. The algorithms mentioned was Linear and Rigid regression and we ran a quick test using both methods and comparing it towards Random Forest, result can be seen in Figure 5-1.
Figure 5-1 – Comparing RF, Linear, and Ridge regression algorithms

Conclusions from result is that more research on different regression algorithms should have been done. If a deeper understand of which algorithms is used to what data, a better model could be created.

As mentioned in the introduction, the goal for Sweden’s government is that 25% of all transportation should be done using public transportation. An aspect for this to succeed is to make the forecasts more reliable. If forecasts were more reliable, more people would probably use it.

What we believe is that if more people would take the public transportation instead of their own car, the prices for a ticket would most likely decrease. Also, taking that many cars off the road could increase the living standard within cities due to lower noise and traffic jams. Especially if the public transportation companies used their profit to invest in electrical buses, which are both quieter and more environmentally friendly depending on the source of electricity. With more money invested in the future of public transportation such as electrical buses will leave a positive effect on the environment compared to today. According to (Tartakovsky, et al., 2012) the average for occupancy of a car is 1.2 passengers/car which is a waste of space on the roads and unnecessary emissions for the environment.

We are satisfied with the result of this paper even though we were hoping for better results. What could have been improved is the amount of data collection for this research. The project has progressed as intended even if the we didn’t have any prior knowledge on Machine Learning, this is probably because there was a lot of information on the subject.

5.1 Future work
The most important factor of increasing the accuracy is to collect more data to train the model. Since this was done on 30 days of data, a lot of deviations was not captured. If the model was able to learn from more data, it would capture more cases where the delay and more key features was extreme. Taking the methods used in this research and reapply it to a larger dataset would most likely increase the accuracy because the model would be more adapted to variations in the data. Also, by adding outside features such as GPS, traffic flow, weather information, and more, or analyzing already existing features via the importance for each and removing/adding new ones, could be key for improvement.
What also could be done to improve the performance is to use hyperparameter tuning on each bus stop. As seen in 4.2.1, the $R^2$-score increased when applying it to a specific bus stop.

Furthermore, exploring different regression algorithms and compare them to each other. It might be that Random Forest isn’t the best case for every scenario as seen in Figure 5-1.
## Schedule

<table>
<thead>
<tr>
<th>Activity</th>
<th>Week</th>
<th>38</th>
<th>39</th>
<th>40</th>
<th>41</th>
<th>42</th>
<th>43</th>
<th>44</th>
<th>45</th>
<th>46</th>
<th>47</th>
<th>48</th>
<th>49</th>
<th>50</th>
<th>51</th>
<th>52</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get familiar with Microsoft Azure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collect data from Skånetrafiken</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analyse important variables available in database</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Creating datastructures for model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Research ML algorithms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test ML algorithm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test ML algorithm (against real times?)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analyze the result</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Plan Seminar</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Half Time Seminar</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Approved Final report</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Final Seminar</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
References


Gupta, B. et al., 2017. [Online] Available at: https://pdfs.semanticscholar.org/fd39/e1fa85e5b3fd2b0d000230f6f8be9dc694ae.pdf [Accessed 09 12 2018].


Näringsdepartementet, 2003. Kollektivtrafik med människan i centrum. [Online] Available at: https://www.regeringen.se/49b4b4/content/assets/1a02d77d8119430985e0fb81c866863ef-del-1-missiv--kapitel-4 [Accessed 13 9 2018].


