



National Technical University of Athens
School of Electrical and Computer Engineering
Department of Computer Science

Predictive Refueling

Combining machine learning techniques with optimization calculations

DIPLOMA THESIS

GEORGIOS DASOULAS

Supervisor : Nectarios Koziris
Professor N.T.U.A.

Athens, March 2018



National Technical University of Athens
School of Electrical and Computer Engineering
Department of Computer Science

Predictive Refueling

Combining machine learning techniques with optimization calculations

DIPLOMA THESIS

GEORGIOS DASOULAS

Supervisor : Nectarios Koziris
Professor N.T.U.A.

Approved by the examining committee on the March 22, 2018.

.....
Nectarios Koziris
Professor N.T.U.A.

.....
Georgios Stamou
Associate Professor N.T.U.A.

.....
Georgios Goumas
Assistant Professor N.T.U.A.

Athens, March 2018

.....
Georgios Dasoulas

Electrical and Computer Engineer

Copyright © Georgios Dasoulas, 2018.
All rights reserved.

This work is copyright and may not be reproduced, stored nor distributed in whole or in part for commercial purposes. Permission is hereby granted to reproduce, store and distribute this work for non-profit, educational and research purposes, provided that the source is acknowledged and the present copyright message is retained. Enquiries regarding use for profit should be directed to the author.

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the National Technical University of Athens.

Contents

Contents	5
List of Tables	7
List of Figures	9
Abstract	11
Acknowledgements	13
1. Introduction	15
1.1 Machine Learning for Prediction	15
1.2 Fuel Prices in Germany	15
1.3 Thesis Objective	16
1.4 Text Structure	17
2. Fuel Price Prediction	19
2.1 Introduction	19
2.2 Related Work	19
2.3 Theoretical Background	19
2.3.1 Machine Learning for Regression Problems	20
2.3.2 Time Series and Machine Learning	25
2.3.3 Machine learning and Prediction	25
2.3.4 Single and Multi-Step Prediction	26
2.3.5 Clustering and general prediction model	27
2.4 Methodology	28
2.4.1 Data Extraction	28
2.4.2 Data Preprocessing	28
2.4.3 Local Prediction Model	29
2.4.4 Feature Vector Approaches	31
2.4.5 General Prediction Model	32
2.5 Implementation Details	34
3. Route Optimization	35
3.1 Introduction	35
3.2 Related Work	35
3.3 Theoretical Background	35
3.3.1 The Simple Fixed-Route Refueling Problem	37
3.3.2 Algorithm	37
3.3.3 The case of "zero knowledge"	40
3.4 Methodology	41
3.4.1 Route Definition	41
3.4.2 Route Benchmarking	41

3.5	Evaluation of the prediction-based optimization	44
3.5.1	Policies	45
3.6	Implementation Details	45
4.	Results	47
4.1	Introduction	47
4.2	Prediction - Local Model	47
4.2.1	Embedding Vector	47
4.2.2	Prediction Horizon	48
4.2.3	Parameter Tuning	49
4.2.4	Performance Comparison of Machine Learning Estimators	52
4.2.5	Visualisation of SVR performance for different forecasting steps	53
4.3	Prediction - General Model	54
4.3.1	Parameter Tuning	54
4.3.2	Performance Comparison of Machine Learning Estimators	57
4.4	Prediction and Optimization	58
4.4.1	Optimal Policy	58
4.4.2	Local Model	59
4.4.3	Global Model	61
5.	Conclusions and Further Work	65
5.1	Conclusions and Discussion	65
5.1.1	Prediction Accuracy	65
5.1.2	Financial Time Series - Fuel Prices	65
5.1.3	Contribution of Prediction into Optimization	66
5.2	Further Work	66
	Bibliography	69
	Appendix	73
A.	Parameter Tuning	73
A.1	Multilayer Perceptron	73
A.1.1	Local Model	73
A.1.2	General Model	75

List of Tables

2.1	Features for DaF approach	31
2.2	Features for general prediction model	32
3.1	Route Format	41
3.2	Route Information	43
4.1	k-NN Parameter Tuning - TSA - Local Model	49
4.2	RF Parameter Tuning - TSA	49
4.3	SVR Parameter Tuning - TSA	50
4.4	k-NN Parameter Tuning - DaF - Local Model	50
4.5	RF Parameter Tuning - DaF	51
4.6	SVR Parameter Tuning - DaF	51
4.7	Optimal parameters for each estimator - Local Model	52
4.8	k-NN Parameter Tuning - TSA - General Model	54
4.9	RF Parameter Tuning - TSA - General Model	54
4.10	SVR Parameter Tuning - TSA - General Model	55
4.11	k-NN Parameter Tuning - DaF - General Model	55
4.12	RF Parameter Tuning - DaF - General Model	56
4.13	SVR Parameter Tuning - DaF - General Model	56
4.14	Optimal parameters for each estimator - General Model	57
A.1	MLP for DaF Approach - Local Model	73
A.2	1 Hidden Layer	73
A.3	2 Hidden Layers	73
A.4	MLP for TSA Approach - Local Model	74
A.5	1 Hidden Layer	74
A.6	2 Hidden Layers	74
A.7	3 Hidden Layers for MLP - TSA - Local Model	74
A.8	MLP for DaF Approach - General Model	75
A.9	1 Hidden Layer	75
A.10	2 Hidden Layers	75
A.11	MLP for TSA Approach - General Model	75
A.12	1 Hidden Layer	75
A.13	2 Hidden Layers	75
A.14	3 Hidden Layers for MLP - TSA - General Model	76

List of Figures

1.1	Fuel Price Fluctuation in a 7-day Period [1]	16
1.2	General Workflow	16
2.1	Optimal Hyperplanes for classification (left) and regression(right)	21
2.2	Architecture of Multilayer Perceptron	23
2.3	Schema of Time Series Prediction	25
2.4	Schema of Time Series Prediction [21]	26
2.5	Sample Set	28
2.6	Feature Vector	28
2.7	Tables of Postgre database	28
2.8	Workflow of ML-based rediction	29
2.9	Validation scheme for temporal data	30
2.10	Diagram of general prediction model workflow	32
2.11	Map of gas stations in Germany	33
3.1	Route Setting	36
3.2	Route Definition with Local Model Prediction	42
3.3	Route Definition with General Model Prediction	42
3.4	Examined Route with 50 gas stations	43
3.5	Examined Route with 150 gas stations	44
3.6	Refueling Policies Comparison	44
4.1	Influence of embedding vector	48
4.2	RMSE performance for different prediction horizons	48
4.3	RMSE performance for best configurations of NNR, RF, SVR, MLP	52
4.4	Prediction of SVR regression for 1, 3, 7, 14 days	53
4.5	SVR Prediction for 3.5 months	53
4.6	RMSE performance for best configurations of NNR, RF, SVR, MLP	57
4.7	Percentage Decrease for all 20 routes	58
4.8	Percentage Decrease for DaF Approach per route type	59
4.9	Average Percentage Decrease for DaF Approach	59
4.10	Percentage Decrease for TSA Approach	60
4.11	Average Percentage Decrease for TSA Approach	60
4.12	Percentage Decrease for DaF Approach	61
4.13	Average Percentage Decrease for DaF Approach	61
4.14	Percentage Decrease for TSA Approach	62
4.15	Average Percentage Decrease for TSA Approach	62

Abstract

The purpose of this diploma thesis is the design, the implementation and the experimentation of a recommender system, that aims to minimize the total refueling cost in vehicle routes. Through this work, a prediction-based optimization scheme is suggested, which efficiently combines machine learning algorithms with optimization calculations, in order to provide to the user an optimal refueling policy. This scheme can be divided into three parts. The development of prediction models for fuel prices, the implementation of the optimization algorithm and the unification of the two processes.

In the recent past, a lot of research has been made in the field of financial time series modeling through machine learning. This is a result of the recent emergence of artificial intelligence in real world applications. Moreover, the last decade many efficient fuel optimizers and route optimization algorithms have been developed. However, in order to apply an optimization algorithm, that minimizes the refueling cost, the fuel prices should be known. Through this work we try to combine these two fields and propose a fuel optimizer, which does not have a priori the knowledge of fuel prices, but instead exploits prediction models, in order to get an accurate estimation.

In the first part, we focus independently on the prediction problem for fuel price data in Germany. We examine 4 types of machine learning estimators, Support Vector Machines, Neural Networks, Random Forests and K-Nearest Neighbors. We conclude that specific configurations of Support Vector Machines and Random Forest outperform the other estimators.

Next, we deal with the optimization problem and the contribution of the prediction models into the optimizer. We note that using prediction models, we can achieve at maximum a 5.20% decrease of the refueling cost, that we would have, if we did not use this recommender system. This percentage decrease, indeed, is only 0.20% away from the optimal refueling cost, which we could have if we knew a priori the fuel prices.

Key words

Machine Learning, Time Series Modeling, Artificial Intelligence, Vehicle Routing, Prediction-based Optimization

Acknowledgements

The present thesis was conducted under the supervision by the professor of N.T.U.A, Prof. Nectarios Koziris, whom I especially thank for his total support. This work was a result of the collaboration with the company NIKI Ltd. Digital Engineering, based on Ioannina and the company TWT GmbH Science & Innovation, based on Stuttgart, Germany, by which the specific topic has been assigned to me. I want to separately thank Dr. Florian Mittag (TWT GmbH) and Dr. Alexander Paar (TWT GmbH) for their technical support, their guidance and their trust that they showed to me. Moreover, I want to thank Mrs. Vivi Zapsa (NIKI Ltd.) for her total support. Furthermore, I want to thank Dr. Ioannis Konstantinou (N.T.U.A) for his help.

Continuing, I want to thank my family, who supported me in every choice and step and gave me the appropriate education and supplies.

Last but not least, I want to thank my friends, who were always by me and were an important part of my life.

Georgios Dasoulas,
Athens, March 22, 2018

Chapter 1

Introduction

In the today's digital era, the science of information technology and data becomes a crucial part of every sector in industry and production. One very characteristic example is the automotive industry, where the vehicles are getting smarter throughout the years of technological evolution. Many manufacturing operations in automotive industries are still largely dependent on experience-based human decisions. The emergence of Big Data, in conjunction with machine learning in automotive companies leads to an increased level of accuracy in decision-making and better performance.

The relevant problem that we are going to discuss and analyze in the present work is the combination of machine learning algorithms and optimization calculations in the vehicle refueling cost minimization. Specifically, we are going to investigate into machine learning algorithms for time series prediction, in order to acquire knowledge about the patterns of the fuel price fluctuations in the region of Germany.

1.1 Machine Learning for Prediction

If we look on the research of previous decades there is an abundant amount of work through statistical methods for the modelling of financial time series, like the fuel prices. Such statistical methods are Box-Jenkins ARIMA and GARCH. However, later *Machine Learning* has become an emerging study field of computer science and statistics and has achieved major contributions to tasks concerning prediction. *Machine Learning* exploits artificial algorithms in order to learn patterns, underlying given data and produce a model which is able to predict on new data with the same structure.

1.2 Fuel Prices in Germany

Since 2013, all 14,200 gas stations in the Federal Republic of Germany have been reporting fuel prices (petrol, diesel, LPG, CNG) to a central reporting authority (Benzinpreisstelle). For this period, all historical fuel prices are publicly available. This data basis allows a correlation analysis of, for example, the gas station locations, delivery times, day times, weekdays or the beginning of school holidays with the development of fuel prices.

As we can see from the figure below, gasoline and diesel prices substantially fluctuate in Germany. There are four to nine movements per day on average. As it is shown, the price levels depend strongly on the hour of the day.

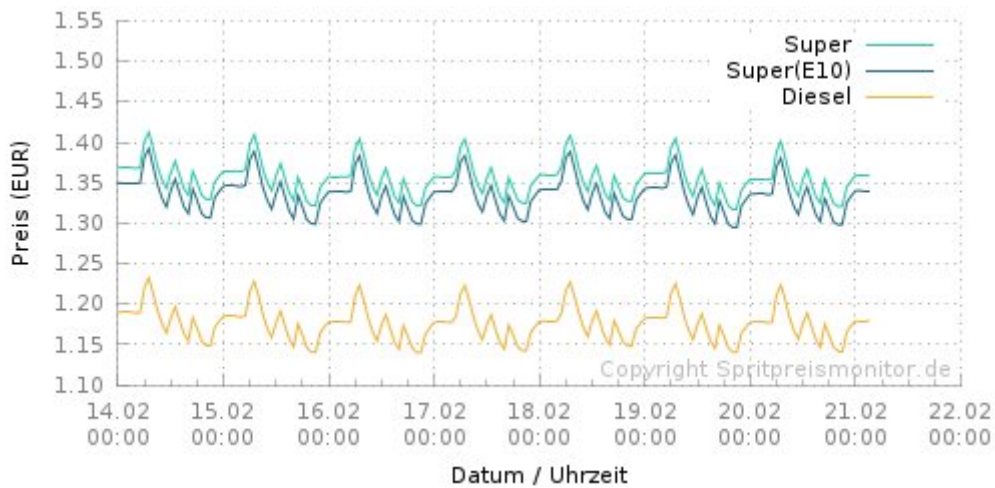


Figure 1.1: Fuel Price Fluctuation in a 7-day Period [1]

Except from the volatility that the fuel prices show throughout the day, we can observe that in the short term an iterative price cycle is followed on. This gives us the opportunity to search for patterns into these financial time series.

1.3 Thesis Objective

The goal of this thesis is the design and the evaluation of a recommender system, which aims to the minimization of the refueling cost for given vehicle routes. Specifically, we want for a given vehicle route to specify the required stops in gas stations, in order to minimize the refueling cost. Thus, it is obvious that we need to know the fuel price of each gas station, which a vehicle meets along its route. For this knowledge, we are going to apply machine learning algorithms, in order to predict the future fuel prices. In this context, we are going to deal with two problems, the problem of prediction of the fuel prices in Germany and the problem of the route optimization. The general workflow of this work is visualized in the next figure:

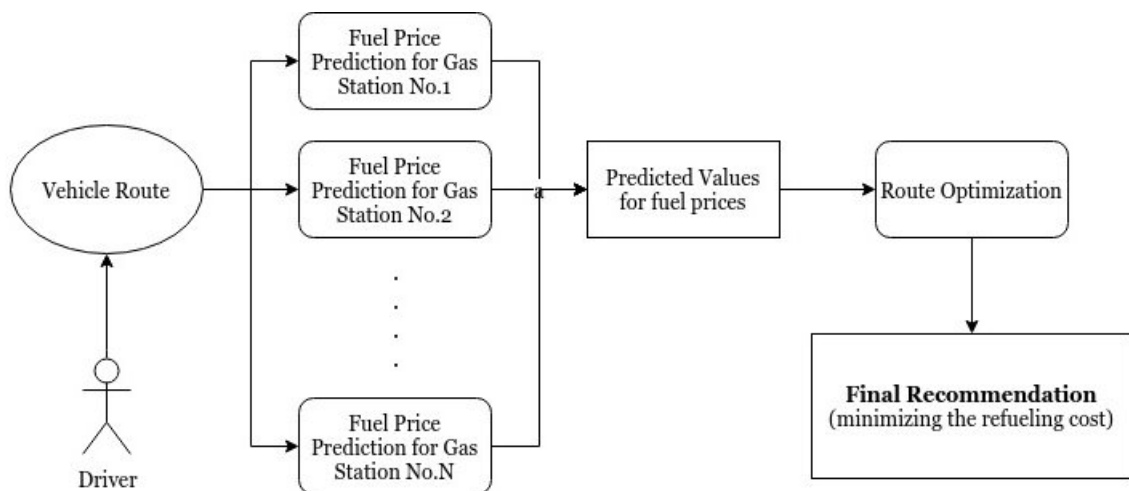


Figure 1.2: General Workflow

1.4 Text Structure

This work is separated into two basic branches, the **Prediction** and the **Optimization** phase.

- In chapter 2, we focus on the fuel price prediction through machine learning techniques. Firstly, we present the related work, the theoretical background and the prerequisite knowledge.
Consequently, the methods and approaches, which were used in the current work are analyzed. Finally, the implementation details are presented.
- In chapter 3, we proceed in the optimization phase, where the analysis and the implementation of the basic algorithm is made. Specifically, as proceeded in chapter 2, we begin by introducing the theoretical knowledge and we continue by analyzing the methodology and implementation.
- In chapter 4, the results of both phases are presented and commented. In the end of this chapter, we present the basic conclusions of the work and topics for further research and discussion.
- In chapter 5, we write down our conclusions and present a few topics for further work that arose during this work.
- In the Appendix, a detailed parameter tuning is presented.

Chapter 2

Fuel Price Prediction

2.1 Introduction

Our goal is to create appropriate prediction models for fuel prices of gas stations in Germany, which they can provide as output the prices for the specific date and hour, which our optimizer requires. The present work focuses on static models. This means that the estimators are pretrained and are used for the prediction of future values. Concerning the used features, 2 approaches are followed. According to the first approach, we used only static time variables, like hour, week, public holidays and other. According to the second approach and following the existing bibliography in machine learning for time series, we used as features the lagged values, i.e the previous observed values of fuel price. For the time series approach, the performance of the estimators in the short and long term was, also, measured and compared. Moreover, due to our need for the development of a general model, which will be able to describe fuel price patterns of all the gas stations, we applied a clustering method, in order to extract spatial features.

2.2 Related Work

Financial time series forecasting has long been a topic of interest. Although traditional time series models like ARIMA and GARCH have been researched and proved to be effective in predicting, their performances are still far from satisfying. Machine learning, as an emerging research field in recent years, has brought about many improvements in tasks such as regression and classification, and it's also promising to exploit the methodology in financial time series prediction. Support Vector Machines and different structures of Neural Networks are widely used in time series forecasting and regression problems.[7] [8] [9] [13] In the last decade, recurrent structures of Neural Networks, as Long-Short Term Memory Networks and Deep Learning are used often for developing patterns among temporal data.[18] [19]

2.3 Theoretical Background

First of all, we are going to present the theory behind the machine learning algorithms that were used for the fuel price prediction. Consequently, we describe the machine learning approach for Time Series, introducing **autoregressive processes** and **multi-step forecasting** strategies. Finally, we describe the clustering algorithm **DBSCAN** for the general model.

2.3.1 Machine Learning for Regression Problems

In this work, we have examined 4 different types of machine learning algorithms:

- Support Vector Machines
- Random Forests
- Neural Networks
- K-Nearest Neighbors

We are going to make a brief introduction, firstly, about the theory behind these algorithms.

Support Vector Machines

Originally SVM was used for classification, i.e. searching for the optimal separating surface, the hyperplane, equidistant from the two classes (Vapnik 1995). More formally, a support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space. A good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. This optimal separating hyperplane has many nice statistical properties. SVC is outlined first for the linearly separable case. Kernel functions are then introduced in order to construct non-linear decision surfaces. Finally, for noisy data, when complete separation of the two classes may not be desirable, slack variables are introduced to allow for training errors.

The Support Vector Methods can also be applied to the case of regression by introducing an ϵ - insensitive loss function (Vapnik 1995, Smola 1996). As with the Support Vector Classification algorithm, optimal separating hyperplane is searched for regression. Support Vector Regression (SVR) relied on defining a loss function that ignored errors that were within a certain distance of the true value. Moreover, loss function allows the concepts of margin to be carried over to the regression case keeping all of the nice statistical properties. Mathematically, training the original SVR is described by the following:

- $\vec{w} \cdot \vec{x} - b = 0$, the hyperplane for the set of points $(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)$
- minimize $\frac{1}{2} ||w||^2$
- subject to $\begin{cases} y_i - \langle w, x_i \rangle - b \leq \epsilon \\ \langle w, x_i \rangle + b - y_i \leq \epsilon \end{cases}$

where w is the normal vector of the hyperplane, x_i is a training sample with target value y_i .

We can see from the figure below the difference between the case of the classification and regression for a two dimensions space:

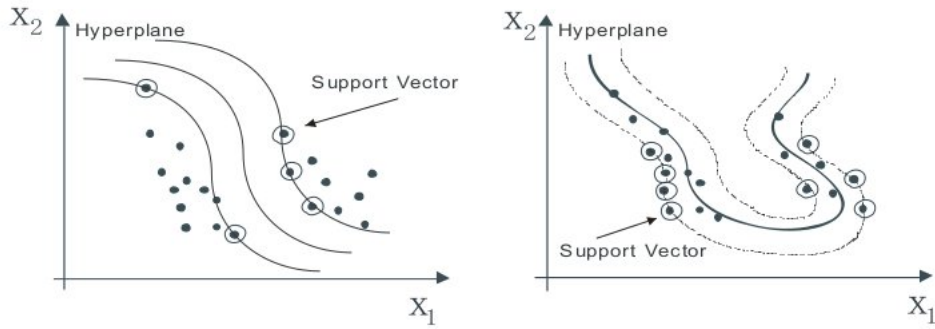


Figure 2.1: Optimal Hyperplanes for classification (left) and regression(right)

Sometimes, however, this may not be the case, or we also may want to allow for some errors. Analogously to the soft margin loss function, one can introduce slack variables ξ_i , ξ_i^* to cope with otherwise infeasible constraints of the optimization problem (2). Hence we arrive at the formulation:

- $\vec{w} \cdot \vec{x} - b = 0$, the hyperplane for the set of points $(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)$
- minimize $\frac{1}{2} \|\vec{w}\|^2 + C * \sum_{i=1}^l (\xi_i + \xi_i^*)$
- subject to $\begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases}$

From this formulation, it is obvious that we introduce now 2 free parameters, the parameter C and the parameter ε .

Free Parameters

Parameter C: Parameter C determines the trade off between the model complexity (flatness) and the degree to which deviations larger than are tolerated in optimization formulation for example, if C is too large (infinity), then the objective is to minimize the empirical risk only, without regard to model complexity part in the optimization formulation.

Parameter ε : Parameter ε controls the width of the ε -insensitive zone, used to fit the training data. The value of ε can affect the number of support vectors used to construct the regression function. The bigger, the fewer support vectors are selected. On the other hand, bigger ε -values results in more 'flat' estimates. Hence, both C and ε -values affect model complexity (but in a different way).

Regression Trees and Random Forest

A regression tree is built through a process known as binary recursive partitioning, which is an iterative process that splits the data into partitions or branches, and then continues splitting each partition into smaller groups as the method moves up each branch. All records in the training set (pre-classified records that are used to determine the structure of the tree) are grouped into the same partition. The algorithm then begins allocating the data into the first two partitions or branches, using every possible binary split on every field. The algorithm selects the split that minimizes the sum of the squared deviations from the mean in the two separate partitions. This splitting rule is then applied to each of the new branches. This process continues until each node reaches a user-specified minimum node size and becomes a terminal node.

Pruning the tree:

Since the tree is grown from the training set, a fully developed tree typically suffers from over-fitting (i.e., it is explaining random elements of the training set that are not likely to be features of the larger population). This over-fitting results in poor performance on real life data. Therefore, the tree must be pruned using the validation set.

The tree is pruned to minimize the sum of:

- 1) the output variable variance in the validation data, taken one terminal node at a time; and
- 2) the product of the cost complexity factor and the number of terminal nodes.

Random Forest:

Random Forest is a type of ensemble learning for decision trees. Specifically, this is a variation of the bagging method. This method works by training multiple weak regression trees using a fixed number of randomly selected features ($\frac{\text{features}}{3}$ for prediction), then takes the average value for the weak learners and assigns that value to the strong predictor. Typically, the number of weak trees generated could range from several hundred to several thousand depending on the size and difficulty of the training set. Random trees are parallelizable, since they are a variant of bagging. However, since random trees select a limited amount of features in each iteration, the performance of random tree is faster than bagging.

Multilayer Perceptron and Neural Networks

The *Multilayer Perceptron* is a kind of feedforward artificial neural network, a series of model partly inspired by working principles of human brain and it consists of at least 3 layers of nodes.

MLP fits a non-linear function(mapping) from input variable vector to output value vector. It's essentially a directed graph consists of layers of neurons linked with each other. Every link between neurons has a weight.

As is shown in Figure 2.2, every circle stands for a neuron, while a line of neurons forms a layer. Layers in MLP can be mainly classified into **input** layer, **hidden** layer and **output** layer according to their different role.

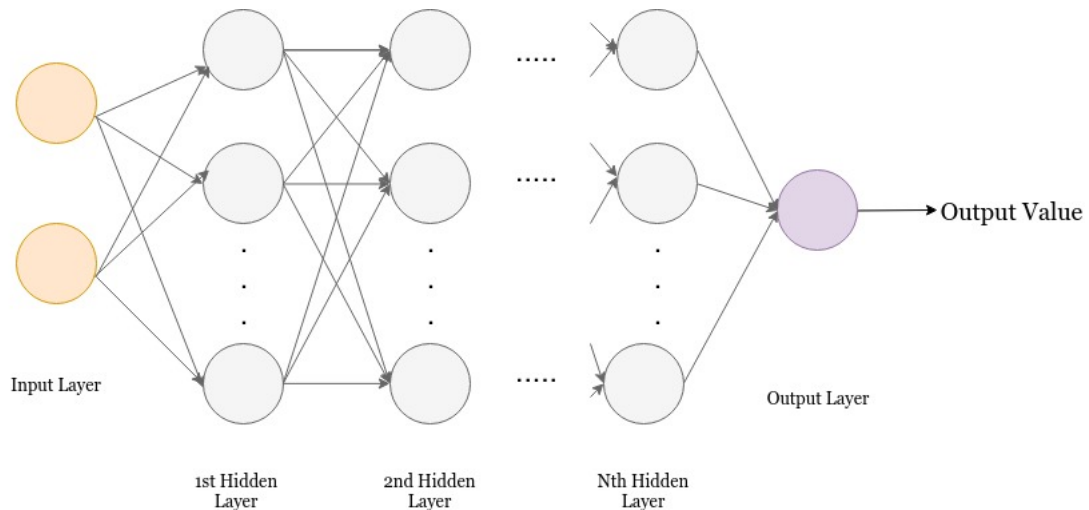


Figure 2.2: Architecture of Multilayer Perceptron

Layers:

- The input layer accepts input variable vector
- The output layer outputs the result of processing input vector
- The hidden layer(s) between input layer and output layer, they accept vector from output by the input layer or the previous hidden layer, and output vector to next hidden layer or the output layer.

The process of vector transferring between layers can be seen as finding correct mapping function. Each layer is fully connected to next layer, while within each layer there is no any connection. Since MLPs are fully connected, each node in one layer connects with a certain weight w_{ij} to every node in the following layer.

Nodes:

The number of nodes for each layer can be different. Specifically, the size of the input layer is the same with the size of the feature vector. The size of the output layer is going to be 1, as we want to predict a single value.

Activation Function:

Except for neurons in input layer, every neuron in other layers has a non-linear activation function, which would perform certain function transformation to the input of each neuron. The most common activation functions are the:

$$y(x) = \tanh(x) \text{ and}$$

$$y(x) = (1 + e^{-x})^{-1}$$

For our experimentations and the parameter tuning, we are going to use the hyperbolic tangent as the activation function: $y(x) = \tanh(x)$.

K-Nearest Neighbors

K-Nearest Neighbors is a nonparametric classification method, based on the measurement of a point's similarity to a training set containing patterns for which class labels are supplied. k-Nearest Neighbors is a memory-based method and does not build a model through learning. The classification is made by aggregating the values provided by the training patterns in the vicinity of the current point.

k-NN can be also used for regression, when $y_i \in R$. In this case, one computes the x associated value as an average of the ones associated to the k nearest neighbors of x :

$$y = \sum_{i=1}^k \frac{y_{o(i)}}{k}.$$

A better accuracy of the prediction is reported to be obtained when one weights the neighbors, according to their closeness to x :

$$y = \frac{\sum_{i=1}^k w_{o(i)} * y_{o(i)}}{\sum_{i=1}^k w_{o(i)}}.$$

where $w_{o(i)}$ is a non-decreasing function of the distance between x and $x_{o(i)}$. This approach gives more importance to the closest neighbors. For example, one can use weights that are (approximately) inverse proportional to the distance:

$$w_{o(i)} = \frac{1}{\epsilon + d(x, x_{o(i)})}.$$

In order to use the k-NN method in a time series-prediction problem, one has to:

1. choose the concrete similarity functions used to find the closest neighbors
2. decide how to effectively produce the prediction.

In our experimentation, we are going to examine the inverse of the distance function, as this described above and the uniform function, where all weights are the same.

2.3.2 Time Series and Machine Learning

A time series is a sequence of observations $s_t \in \mathfrak{X}$, usually ordered in time. A fairly general model for the time series can be written : $s_t = g(t) + \phi_t, t = 1, \dots, T$, where $g(t)$ is called systematic part, or signal or trend, and it is a deterministic function of time, while ϕ_t is called stochastic sequence, or noise and follows a probability law.

Autoregressive Processes:

In order to exploit machine learning techniques for the modelling of time series, we first need to define the term of **autoregressive process**.

Suppose that w_t is a purely random process with zero mean and variance σ_w^2 . A process ϕ_t is said to be an autoregressive process of order n if $\phi_t = \alpha_1 * \phi_{t-1} + \dots + \alpha_{t-n} * \phi_{t-n} + w_t$. This means that the next value is a linear weighted sum of the past n values plus a random shock.

Of course, these models assume that the relation between past and future is linear. Once we assume that the linear assumption does not hold, we may extend the current formulation to a **Nonlinear autoregressive** formulation:

$\phi_t = f(\phi_{t-1}, \phi_{t-2}, \dots, \phi_{t-n}) + w_t$, where the missing information is lumped into a noise term w .

In what follows we consider this relationship as a particular instance of a dependence $y = f(x) + w$, where y denotes the next value of ϕ_t to be predicted and $x = [\phi_{t-1}, \phi_{t-2}, \dots, \phi_{t-n}]$ denotes the n -dimensional vector also known as **embedding vector**.

2.3.3 Machine learning and Prediction

For this approach, we consider the prediction problem as a problem of **supervised learning** problem, where we have to infer from historical data the possibly **nonlinear** dependance between the input (past embedding vector) and the output (future value). The procedure that we apply is the following:

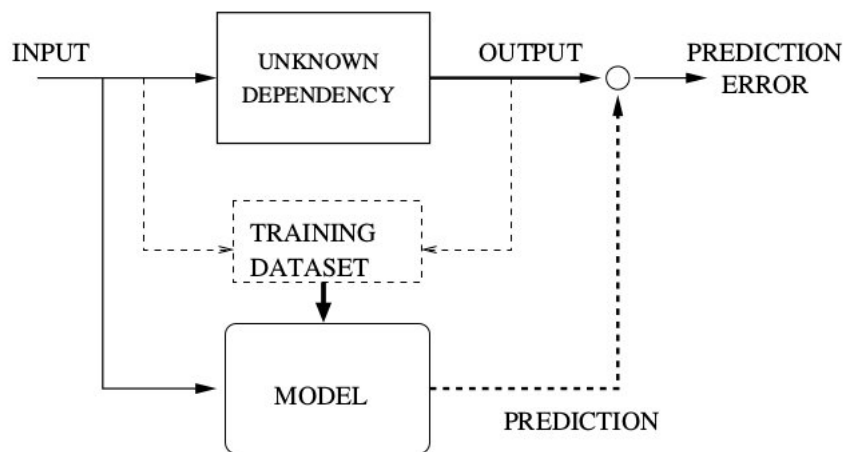


Figure 2.3: Schema of Time Series Prediction

2.3.4 Single and Multi-Step Prediction

Once a model of the embedding mapping is available, it can be used for two objectives: one-step-ahead prediction and multi-step ahead prediction. In one-step-ahead prediction, the n previous values of the series are available and the forecasting problem can be cast in the form of a generic regression problem.

The approximator \hat{f} returns the prediction of the value of the time series at time $t + 1$ as a function of the n previous values (the rectangular box in the figure below containing z^{-1} represents a unit delay operator, i.e., $\phi_{t-1} = z^{-1}\phi_t$)

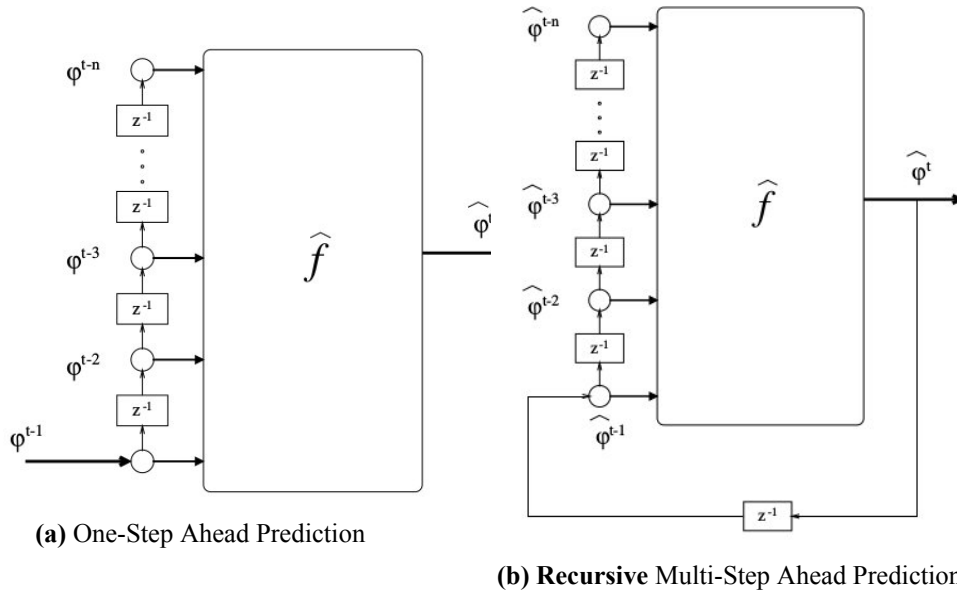


Figure 2.4: Schema of Time Series Prediction [21]

In the case of multi-step ahead prediction, there are different strategies according to two features: the *horizon of the training criterion* and the single-output or multi-output nature of the predictor. Specifically, the most common strategies are :

1. **Recursive:** the model predicts H steps ahead by iterating a one-step-ahead predictor whose parameters are optimized to minimize the training error on one-step-ahead forecast (one-step-ahead training criterion).
2. Iterated strategy where parameters are optimized to minimize the training error on the iterated h_{tr} -step-ahead forecast (h_{tr} -step-ahead training criterion) where $1 < h_{tr} \leq H$.
3. **Direct:** the model makes a direct forecast at time $t + h - 1, h = 1, \dots, H$ by modeling the time series in a multi-input single-output form
4. **Direc:** direct forecast but the input vector is extended at each step with predicted values.
5. **MIMO:** the model returns a vectorial forecast by modeling the time series in a multi-input multi-output form

In this work, we are going to use the **Recursive** strategy, as it is shown in the figure 2.4b.

2.3.5 Clustering and general prediction model

During our effort to develop an efficient prediction model, we faced the fact that the approach of local models, which are personalized for a single gas station, requires the creation of an equal number of models and gas stations. Specifically, the number of gas stations in Germany amounts to 14000, so we would need 14000 personalized models. This fact creates the need for a general prediction model, which can describe patterns from many gas stations.

DBSCAN Algorithm for geospatial features

In order to develop a general spatiotemporal prediction model, we need an efficient algorithm for the clustering of gas stations according to their latitude and longitude. In 1996, Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu proposed the algorithm *Density-based spatial clustering of applications with noise* (DBSCAN). The meaning of density-based clustering is that given a set of points in some space, the algorithm groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away).

For the sake of completeness, we are going to show a brief description of the algorithm:

For the purpose of DBSCAN clustering, the points are classified as **core** points, **(density-)reachable** points and **outliers**, as follows:

- A point p is a **core** point if at least $minPts$ points are within distance ϵ (ϵ is the maximum radius of the neighborhood from p) of it (including p). Those points are said to be directly reachable from p .
- A point q is **directly reachable** from p if point q is within distance ϵ from point p and p must be a **core** point.
- A point q is reachable from p if there is a path p_1, \dots, p_n with $p_1 = p$ and $p_n = q$, where each p_{i+1} is **directly reachable** from p_i (all the points on the path must be **core** points, with the possible exception of q).
- All points not reachable from any other point are **outliers**.

Algorithm:

Consider a set of points in some space to be clustered. In our case, this set of points consists of pairs of latitude and longitude. We define the **core** points as follows:

A point p is a **core** point if at least $minPts$ points are within distance ϵ (ϵ is the maximum radius of the neighborhood from p) of it (including p). Those points are said to be directly reachable from p .

1. Find the ϵ neighbors of every point, and identify the **core** points with more than $minPts$ neighbors.
2. Find the *connected components* of **core** points on the neighbor graph, ignoring all non-core points.
3. Assign each non-core point to a nearby cluster if the cluster is an ϵ neighbor, otherwise assign it to noise.

2.4 Methodology

Now, we are going to discuss about the methodology and the structure of the experimentation. Specifically, we describe the process of data and feature extraction, the model training and the validation schema. The next figures show the different approaches that were followed :

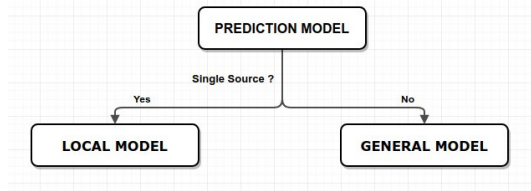


Figure 2.5: Sample Set

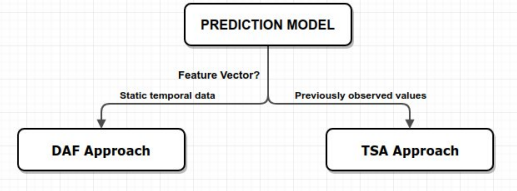


Figure 2.6: Feature Vector

Following the categorization above, in the experimentations of this work we are going to compare the 4 different combinations of these approaches.

2.4.1 Data Extraction

2013, German government enacted the compulsory record of the daily fuel price changes in every gas station in Germany. We can access these data for the training of our models from 2 Postgre databases in <https://www.tankerkoenig.de/>. The first database provides information for the date interval 8.6.2014 - 2.5.2016, while the second one for the date interval 2.5.2016 - today. Each one of the databases consists of 2 tables, one for the general information of each gas station and one for the fuel price history. The attributes of these tables are shown in the next figures:

Tabelle gas_station

Column	Type	Modifiers
id	uuid	not null
version	character varying(10)	not null
version_time	timestamp without time zone	not null
name	text	not null
brand	text	
street	text	
house_number	text	
post_code	text	
place	text	
public_holiday_identifier	text	
lat	double precision	not null
lng	double precision	not null
ot_json	text	not null

(a) Table of gas station information

Tabelle gas_station_information_history

Column	Type	Modifiers
id	integer	not null
stid	uuid	not null
e5	smallint	
e10	smallint	
diesel	smallint	
date	timestamp(0) with time zone	not null
changed	smallint	

(b) Table of history of gas price

Figure 2.7: Tables of Postgre database

2.4.2 Data Preprocessing

The tables above include only the fuel price **changes** in a daily basis. Due to this fact, we modified the data, in order to describe a stable 24-hour timeable. This modification can provide us equidistributed time intervals for the fuel price vector. For this modification, we think the following:

- In the vector of time, between 2 observed fuel price changes x_i, x_{i+1} , we observe always the fuel price x_i .
- We have a 1-hour accuracy in our observations.

2.4.3 Local Prediction Model

The first case, which was examined, is that of a model, which describes the price progress of a single entity-gas station. Here, we use only temporal data, which refer to the corresponding gas station. This case requires the widely known machine learning workflow, as it is appeared in the next figure:

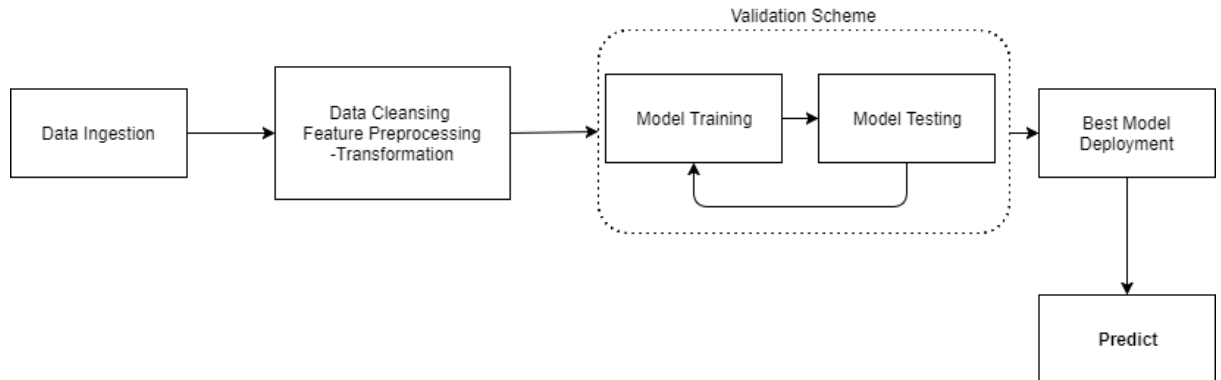


Figure 2.8: Workflow of ML-based prediction

Data Cleansing/Feature Preprocessing-Transformation

Feature Preprocessing: As we have observed from our experimentations, transforming our features is very useful for the performance of specific predictors. For this reason, we use 2 different transformations according to the machine learning estimator

- **Min-Max** transformation, which is defined as: $x' = \frac{x - \min(x)}{\max(x) - \min(x)}$. This transformation transfers each feature into the interval $[0, 1]$.
- **Standardization**, which is defined as: $x' = \frac{x - \bar{x}}{\sigma}$, where x is the original feature vector, \bar{x} is the mean of that feature vector, and σ is its standard deviation. This transformation removes the mean and scales to unit variance.

Validation Scheme

As soon as, we are dealing with time series, we are not allowed to use the classical cross-validation schemes, due to the fact that we need the temporal dependence between different data-points. For this reason we follow a different validation scheme:

Specifically, we follow the sequence of steps below:

Parameters:

n: number of split iterations

Process:

1. We compute **n times** the indices of the train and the test set. The indices are computed in such a way, that the train sets of every iteration are not overlapping and have the same size. The corresponding indices of the test set are following the train indices.
2. For each pair of indices, we fit a **new** model, based on our current estimator (ML algorithm).
3. We predict the values of the test set and then compute the metric of error
4. After all **n** iterations, we compute the **mean** of the metric of the error.

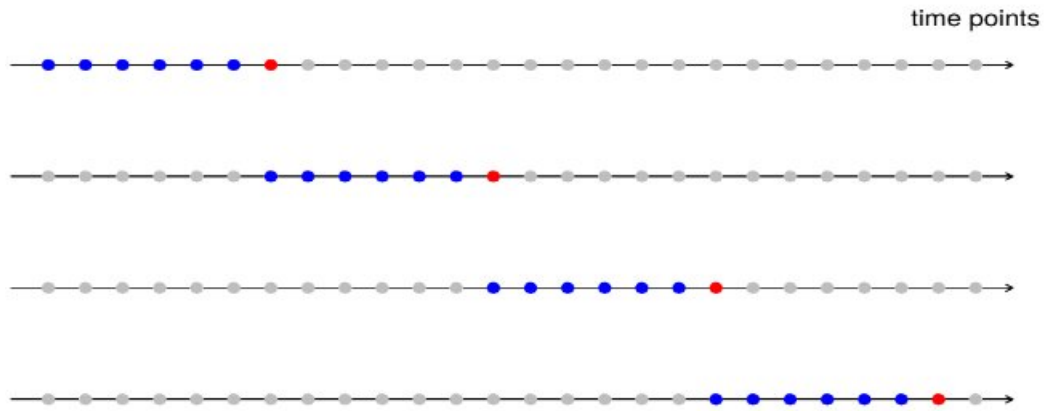


Figure 2.9: Validation scheme for temporal data

Model Training and Testing:

In this phase we are going to find the optimal configuration of our **Predictor**, which will be used as the component to the general system. For the selection of the configuration we make a search across a list of different machine learning estimators, modified for time series. This list includes the below :

- Support Vector Regression
- K-Nearest Neighbor Regression
- Random Forest
- Neural Networks

For each estimator we make a parameter selection, so as to find the optimal set of parameters.

Performance Measurement and Prediction

During the experimentation of the predictors, we used two performance metrics, which are widely used in performance evaluation for regression problems and specifically time series.

- The first one is the **Root Mean Squared Error** (aka. RMSE). The RMSE of predicted values \hat{y}_i for observations i of a regression's dependent variable y_i is computed for n different predictions as the square root of the mean of the squares of the deviations:

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

- The second performance metric is the **Mean Absolute Percentage error** (MAPE). This metric expresses accuracy as a percentage and is defined by the formula:

$$MAPE = \frac{100}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right|$$

2.4.4 Feature Vector Approaches

As we have already noted, we are going to follow two different approaches according to the feature vectors that are used for the prediction model. In the first approach, which we are going to call **DaF** ("Date as Feature"), we exploit the date information as features. In the second approach, **TSA**, we treat the temporal data as time series, implementing Time Series Analysis through machine learning.

DaF Approach

According to this approach, we exploit the hourly and weekly trend of the data. The features that are used are shown in the following table:

Table 2.1: Features for DaF approach

	Features	Value Range
1	Day of Month	1 - 31
2	Day of Week	1 - 7
3	Day of Year	1 - 365
4	Week of Month	1 - 5
5	Week of Year	1 - 53
6	Month of Year	1 - 12
7	Year	2014 - 2017
8	Hour	0 - 23
9	Holiday	{0,1}

As it is shown, we use as a feature a holiday flag, which denotes if the examined date is a holiday in Germany. This is important due to the fact that during holidays, the fuel prices are going up. In order to produce this flag, we had access to the yearly calendar in Germany.

TSA Approach

Here, we treat the temporal data as an autoregressive process, as this was described in the section 2.3.2. Specifically, we use previously observed values of fuel price as features, so as to predict next values. These values are called **lagged variables** and the feature vector is known as **embedding vector** or **look-back** window. During our experimentation, we have to define the length of this embedding vector, which consequently defines the size of the feature vector. The results of this experimentation is shown in the chapter 4.

Differences

The two approaches have the following different characteristics:

- Using the DaF approach, we do not have a view on the dependencies and the correlations between the fuel prices themselves, as we use only time information as features. The nature of this approach makes the prediction model independent from previously observed values, in order to predict new price values.
- The output target using the TSA approach is the immediately next value for the fuel price. That means that we have a predicting horizon of 1 hour. That means that we have to apply a multi-step forecasting strategy, as described in the section 2.3.4. On the other hand, using the DaF approach, we are independent of multi-step forecasting strategy, as we can make predictions for every future timestamp.

2.4.5 General Prediction Model

In the general prediction model, we started enriching our model with **spatial features**, as our spatio-temporal database could provide us. Our goal is to create a model, which can describe the patterns of fuel price changes from not only one gas station, but the whole set of gas stations in Germany. For this reason, experimenting with the same estimators we expanded the size of the feature vector, in order to include spatial information and the size of the sample set, so as to train the model with time series from different sources-gas stations. That means that now we treat the time series from different sources (gas stations) as dependent measurements.

Moreover, taking into account that different fuel brands can show a different behavior in the price fluctuation, we integrated, also, brand information into the feature vector.

In the figure below, we describe graphically the spatial information integration into our model.

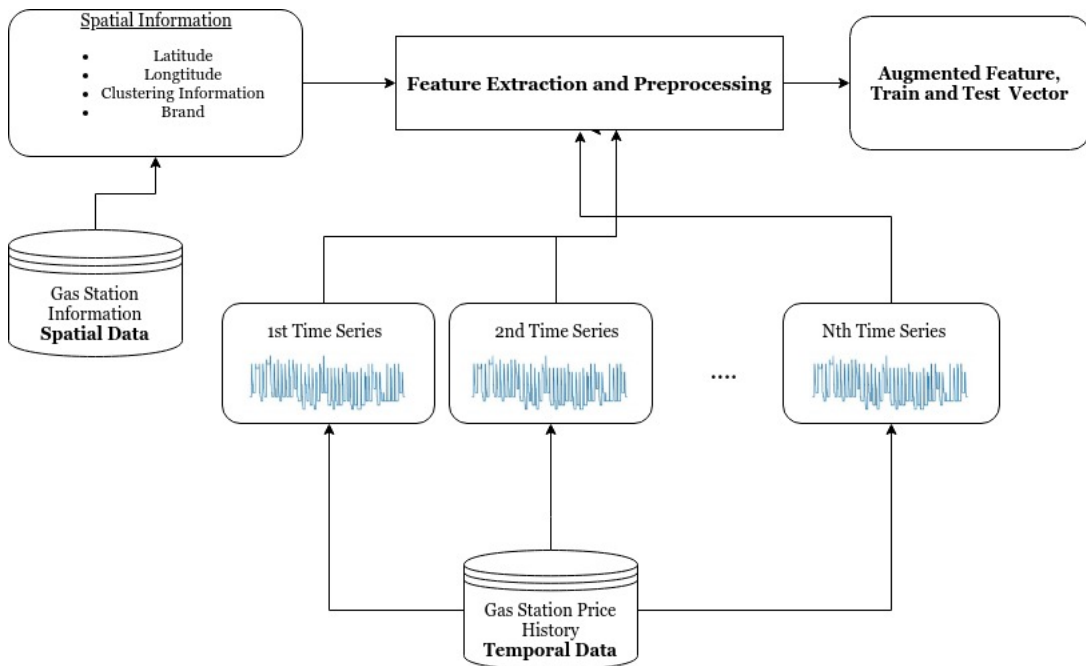


Figure 2.10: Diagram of general prediction model workflow

Feature Vector

In the next figure, we can see the features that were integrated for the general prediction model:

Table 2.2: Features for general prediction model

	Feature	Value Type
1	Latitude	Float
2	Longitude	Float
3	Brand	Alpharithmetic String
4	Neighborhood Cluster	Integer
5	NN-Metric	Float

Latitude and Longitude:

These are the geographic coordinates that specify the north–south and the east–west position of a point on the Earth’s surface respectively. These are angular measurements, both expressed in degrees.

Feature Brand:

Due to the fact that brand is a string, we had to encode this feature into an integer label, in the range of 1 – 78. We have observed 77 different brand names in the data and the last label belongs to the set of gas stations, which are independent.

Neighborhood Cluster:

In order to group the gas stations geographically, we had to find clusters of gas stations across Germany, in which different gas stations are close. For this reason, we applied the **DBSCAN** algorithm for the clustering. Specifically, assuming as maximum neighborhood distance = 1.5km, we have found 7278 clusters.

NN-Metric:

This metric expresses the competitiveness level of each gas station, according to the number of the neighbor gas stations, which are close to each other. After the clustering through DBSCAN, we computed and assigned to each gas station the NN-Metric as follows:

$nn - metric = \frac{1}{\#(cluster_neighbors)}$ for each gas station, where *cluster_neighbors* is the number of neighbors of each gas station.

Below we can see the map of Germany, where all gas stations are depicted:

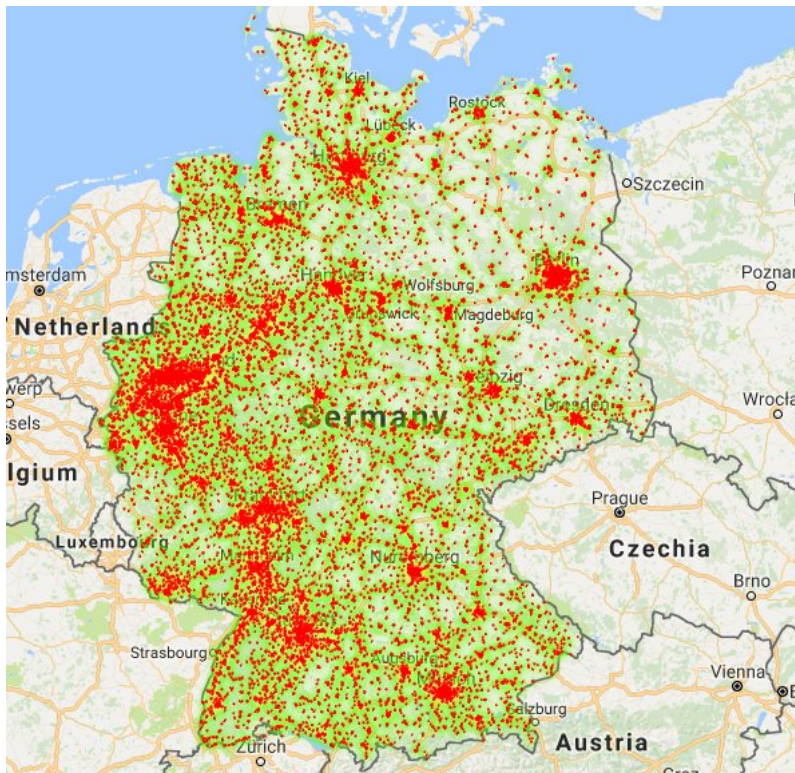


Figure 2.11: Map of gas stations in Germany

2.5 Implementation Details

Now, we present the information about the implementation of the experimentation and prediction.

First of all, we note that the fuel price data are stored in a public Postgre database. So, we used PostgreSQL and psycopg2 (a PostgreSQL adapter for Python), in order to extract and process the raw data.

Moreover, we used Python 3.4 for the prediction phase and specifically the scikit-learn package for the machine learning algorithms. Tensorflow and Keras Library was used for the parameterization of Multilayer Perceptron and the examination of a few configurations in Recurrent Neural Networks, which did not show higher accuracy than the classic MLP.

Chapter 3

Route Optimization

3.1 Introduction

After dealing with the fuel price prediction problem, now we are ready to continue with the optimization phase. In this chapter, we are going to define the refueling problem and the algorithm of refueling cost minimization. Specifically, we define the **Fixed-Route Refueling problem**, where we are provided by a specific vehicle route and our goal is to find the best planning of vehicle stops, so as to minimize the refueling cost. In this thesis, we examine the simple case, where the route is described as a graph, where its nodes are the gas stations, of which it comes by. After implementing the optimization algorithm, we applied pretrained prediction models, so as to forecast the fuel prices of the examined gas stations for the requested time of arrival of the vehicle. Consequently, the unification process of prediction and optimization is described.

3.2 Related Work

Fuel optimizers are decision models that are increasingly recognized as effective fuel management tools, especially by truckload carriers. [30] These fuel optimizers implement route planning algorithms for refueling cost minimization and these algorithms belong to a wider set of routing problems in transportation networks. Specifically, as we will see in the next section, this problem is related to the capacitated lot-sizing problems and the vehicle routing problems. Research work has been made by Shieu-Hong Lin [26],[27] in the field of optimal refueling policies and by Samir Khuller, Azarakhsh Malekian and Julian Mestre [29] in the field of vehicle routing. We are going to focus on the approaches of Shieu-Hong Lin and implement a linear time algorithm in the specific case of *Simple Fixed Path Vehicle Routing Problem*.

3.3 Theoretical Background

In the general case, the vehicle routing problems are related to computing the shortest path or tour visiting a set of locations and satisfying a set of limitations. For this work and experimentation, we are going to assume that we want a predefined route and the set of visiting locations is the set of the gas stations, from which the vehicle comes by.

In the figure below, the general setting of our problem is shown:

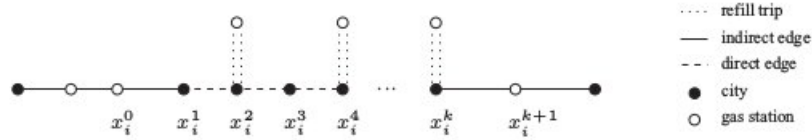


Figure 3.1: Route Setting

As we can see, we have n nodes, of which the first and the last one denote the starting position and the destination of the vehicle respectively. In our case, we are going to assume that the distance between the cities and the gas stations is **zero**, so the route is characterized from the gas stations that the vehicle drives by.

As we have mentioned before, this special case that we focus on is called *The Simple Fixed-Route Refueling Problem* and is a special case of the inventory-capacitated lot-sizing problem (ICLSP).

ICLSP

In the inventory-capacitated lot-sizing problem (ICLSP), the demand for a single commodity in each of the consecutive n stages of time must be met by production in that stage or from the accumulated inventory, and the computing task is to determine the amounts of production over the stages to minimize the total setup, production, and inventory holding cost subject to inventory capacity bounds over the stages. ICLSP is solvable by dynamic programming in $O(n^3)$ time for cases with zero setup costs and concave cost functions for production and inventory holding. [33]

In our case, we have a stationary inventory capacity, zero setup costs, zero inventory holding costs and linear cost functions for production. The vehicle travels along a fixed route with a series of fuel stations on it. The movement of the vehicle is composed of stages moving from one fuel station to the next, and fuel is the *commodity in demand* in these stages. Varying amounts of fuel are needed in these stages due to the differences of terrains and mileage between adjacent stations. Fuel must be purchased (*produced*) in each stage with varying prices from the fuel stations and the capacity of the fuel tank imposes a *stationary upper bound* on the inventory of fuel. The computing task is to determine the amounts of fuel purchased at these stations to reach the destination with minimal total fuel cost.

3.3.1 The Simple Fixed-Route Refueling Problem

Now, we are ready to define typically the problem and the algorithm that solves it in linear time. This algorithm is provided by the corresponding paper of Shieu-Hong Lin [26] and is a greedy algorithm. We are going to omit the proof of correctness, as there is a detailed presentation of it in the previous paper.

Problem Setting:

We consider a vehicle with a fixed fuel tank capacity operates along a fixed route with a series of fuel stations $\mathbf{S} = \{S_0, S_1, \dots, S_n\}$. The vehicle starts with some U_0 units of fuel station S_0 in the beginning of the route and progressively move from current station S_i to next station S_{i+1} for $0 \leq i < n$ until ending in the destination station S_n in the end of the route. We refer to i as the index of station S_i . The vehicle can refuel any nonnegative units of fuel at each fuel station as long as it does not exceed the capacity of the fuel tank.

Problem Instance:

We are going to define the problem instance, according to our parameters. For the simple fixed-route refueling problem, we have a five-tuple $\langle C, U_0, S, P, G \rangle$, regarding a vehicle initially with U_0 units of fuel in a fuel tank of capacity C along a fixed route with a series of fuel stations $S = \langle S_0, S_1, \dots, S_{n-1} \rangle$ and two vectors $P = \langle P_0, P_1, \dots, P_n \rangle$ and $G = \langle G_0, G_1, \dots, G_{n-1} \rangle$, where P_i is the unit fuel price charged at fuel station S_i and G_i is the number of fuel units consumed to move the vehicle from station S_i to station S_{i+1} , $C \in \mathfrak{R}^+$, $U_0 \in \mathfrak{R}^+ \cup \{0\}$ and for $0 \leq i < n$, $P_i \in \mathfrak{R}^+$ and $G_i \in \mathfrak{R}^+$.

Output:

Our goal is to produce a valid refueling policy, i.e a vector $F = \langle f_0, f_1, \dots, f_{n-1} \rangle$, $f_i \in \mathfrak{R}^+ \cup \{0\}$, that minimizes the total fuel cost $\sum_{0 \leq i < n} P_i * f_i$. The validity of the policy occurs, when the vehicle progressively moves to the ending station, never out of fuel and never exceeding the tank capacity.

3.3.2 Algorithm

Given a problem instance $I = \langle C, U_0, S, P, G \rangle$ and in order to construct the greedy algorithm, that computes an optimal valid refueling policy, we are going to define two functions, the $FAR(i)$ and the $LOW(i)$.

Firstly, we define the function $FAR(i)$, which describes the index of the farthest station that the vehicle can reach from S_i if starting with a full tank of fuel at fuel station S_i moving toward the destination station S_n without any refueling afterward. Below, we present the pseudocode for $FAR(i)$.

Algorithm 1: Computing $FAR(\cdot)$

Data: a problem instance $I = \langle C, U_0, S, P, G \rangle$ regarding n stations S_0, S_1, \dots, S_n in \mathbf{S}

Result: an array of n integers $FAR(i), 0 \leq i < n$

```
1  $tank = C$ ;  
2  $i, here = 0$ ;  
3 while  $i < n$  do  
    /* Check reachability of  $S_{here+1}$  with a full tank from  $S_i$  */  
4     while  $tank \geq G_{here}$  and  $here < n$  do  
5          $tank = tank - G_{here}$ ;  
6          $here = here + 1$ ;  
7      $FAR(i) = here$ ;  
    /* Decide the tank level when reaching  $S_{here}$  from  $S_{i+1}$  */  
8     if  $here > i$  then  
9          $tank = tank + G_i$ ;  
10        current section becomes this one;  
11    else  
12         $tank = C$ ;  
13         $here = i + 1$ ;  
14     $i = i + 1$ ;
```

Moreover, we define the function $LOW(i)$, which describes the index of the first station with a lower fuel price that the vehicle would encounter when moving from S_i to S_n . Below, we present the pseudocode for $LOW(i)$.

Algorithm 2: Computing $LOW(\cdot)$

Data: a problem instance $I = \langle C, U_0, S, P, G \rangle$ regarding n stations S_0, S_1, \dots, S_n in \mathbf{S}

Result: an array of n integers $LOW(i), 0 \leq i < n$

```
1  $i = n - 1$ ;  
2 while  $i \geq 0$  do  
3     if  $P_{i+1} < P_i$  then  
4          $LOW(i) = i + 1$ ;  
5         goto step 11;  
6     else  
7          $next = LOW(i + 1)$ ;  
8         while  $P_{next} \geq P_i$  do  
9              $next = LOW(next)$ ;  
10         $LOW(i) = next$ ;  
11         $i = i - 1$ ;  
12 return the array of values  $LOW(i), 0 \leq i < n$ ;
```

We note that the previous ones are two linear time algorithms for the computation of the functions. We can, also, see here that the amount of fuel left when arriving at the destination station S_n is of no concern here.

After defining the functions $FAR(i)$, $LOW(i)$, we are going to present the final Algorithm 3. This algorithm simulates the vehicle movement from the starting fuel station toward the destination station by going through a series of greedy decision-making steps. The following is the highlight of the greedy decision-making rules along with a brief explanation of the variables and the notation used in the algorithm.

The variable i is used to keep the index of the station the algorithm is exploring at the moment. The variable $fuel$ is used to record the amount of fuel currently in the tank when the algorithm simulates the vehicle movement.

Algorithm 3: Finding an optimal refueling policy

Data: a problem instance $I = \langle C, U_0, S, P, G \rangle$ regarding n stations S_0, S_1, \dots, S_n in \mathbf{S}

Result: an optimal refueling policy as a vector \mathbf{F}

```

1 determine  $FAR(i)$  and  $LOW(i)$  for every  $i$  in the range of  $[0, n - 1]$  using Algorithm 1
  and 2;
2 initialize  $F = \langle f_0, f_1, \dots, f_{n-1} \rangle$  to  $\langle 0, 0, \dots, 0 \rangle$ ;
3  $fuel, i = U_0, 0$ ;
4 while  $i < n$  do
5   if  $FAR(i) = i$  then
6      $\lfloor$  stop and report that no valid refueling policy exists;
7   if  $LOW(i) > FAR(i)$  then
8     /* Fill in the entire tank to the full capacity */
9      $f_i = C - fuel$ ;
10     $fuel = C - G_i$ ;
11     $i = i + 1$ ;
12  if  $LOW(i) \leq FAR(i)$  then
13    /* Fill in bare minimum to reach  $S_{low}$  */
14    if  $\sum_{i \leq j < LOW(i)} G_j > fuel$  then
15       $f_i = \sum_{i \leq j < LOW(i)} G_j - fuel$ ;
16    else
17       $f_i = 0$ ;
18    /* Move all the way to station  $S_{low}$  without further refueling */
19     $fuel = fuel + f_i - \sum_{i \leq j < LOW(i)} G_j$ ;
20     $i = LOW(i)$ ;
21 return the refueling policy vector  $F = \langle f_0, f_1, \dots, f_{n-1} \rangle$ 

```

According to the algorithm presented above, the three decision-making conditions are the following:

1. If $FAR(i) = i$, we are not able to move from the current fuel station S_i to the immediately next station even with a full tank of fuel. In this case, there is no valid refueling policy.
2. If $LOW(i) > FAR(i)$, no fuel station reachable within the range of a full tank of fuel from the current fuel station S_i has a lower fuel price. In this case, we greedily fill the tank to full capacity at the current fuel station and then move on to the next station.
3. If $LOW(i) < FAR(i)$, the closest fuel station with a lower fuel price is reachable within the range of a full tank of fuel from the current fuel station S_i . In this case, we fill the minimum amount of fuel needed at the current fuel station to reach that closest station with a lower unit fuel price.

3.3.3 The case of "zero knowledge"

We remind that our initial purpose is to examine how a fuel optimizer can be applied in combination with predicted values of fuel prices. For this reason, it is very helpful to compare the efficiency of the algorithm with the case, that we do not have knowledge on the fuel prices.

In the case that we cannot have a clue about the fuel prices, our problem is simple. Given the vehicle route and the distances of the gas stations, the problem instance is defined as follows:

Problem Instance

We have a four-tuple $\langle C, U_0, S, G \rangle$, regarding a vehicle initially with U_0 units of fuel in a fuel tank of capacity C along a fixed route with a series of fuel stations $S = \langle S_0, S_1, \dots, S_{n-1} \rangle$ and the vector $G = \langle G_0, G_1, \dots, G_{n-1} \rangle$, where G_i is the number of fuel units consumed to move the vehicle from station S_i to station S_{i+1} , $C \in \mathfrak{R}^+$, $U_0 \in \mathfrak{R}^+ \cup \{0\}$ and for $0 \leq i < n$ $G_i \in \mathfrak{R}^+$.

As we can see the only difference is that we do not know the price vector P .

The refueling policy is computed as it is shown in the following simple algorithm:

Algorithm 4: Finding a refueling policy without having knowledge of fuel prices

Data: a problem instance $I = \langle C, U_0, S, G \rangle$ regarding n stations S_0, S_1, \dots, S_n in \mathbf{S}

Result: a refueling policy as a vector \mathbf{F}

```

1 determine  $FAR(i)$  for every  $i$  in the range of  $[0, n-1]$  using Algorithm 1;
2 initialize  $F = \langle f_0, f_1, \dots, f_{n-1} \rangle$  to  $\langle 0, 0, \dots, 0 \rangle$ ;
3  $fuel, i = U_0, 0$ ;
4 while  $i < n$  do
5     if  $FAR(i) = i$  then
6         | stop and report that no valid refueling policy exists;
7     if  $fuel < \sum_{i \leq j < n-1} G_j$  then
8         | /* Fill in the entire tank to the full capacity */
9         |  $f_i = C - fuel$ ;
10        | /* Move all the way to station  $S_{far}$  without further refueling */
11        |  $fuel = fuel + f_i - \sum_{i \leq j < FAR(i)} G_j$ ;
12        |  $i = FAR(i)$ ;
13 return the refueling policy vector  $F = \langle f_0, f_1, \dots, f_{n-1} \rangle$ 

```

As we can see, we use again the function $FAR(i)$, that we have already computed in the previous section.

Brief Explanation:

Starting from the beginning station S_0 , we do not refuel, until we reach the farthest gas station, that our vehicle can reach with the initial tank capacity (computed with the function $FAR(i)$).

Then, we fill up the tank to the full capacity and we continue with the same procedure, until we reach our destination S_n .

This policy describes the normal behavior of a driver, given the spatial information of the gas stations.

3.4 Methodology

After explaining the theory behind the route optimization algorithm, that we are going to use, we are now ready to describe the methodology for our experiments and the structure of the recommender system.

3.4.1 Route Definition

During the experimentation, the route was defined as follows:

- Vehicle Information
 - Velocity
 - Tank Capacity
 - Fuel Consumption
- Gas Station Information
 - Station ID
 - Distances between the consecutive gas stations
 - Estimated Time of Arrival (ETA) for every gas station

Assumptions:

For the sake of simplicity, we made the following assumptions:

Velocity = $75 \frac{km}{h}$.

Fuel consumption = $7 \frac{liters}{100km}$

Tank Capacity = 65 liters.

As we can see, we used constant values for velocity, fuel consumption and tank capacity, in order to simplify the experimentation. These values correspond to characteristics of an average vehicle.

3.4.2 Route Benchmarking

We are going to use for our examination a set of 20 routes across Germany.

Past Routes: In order to compare the different models and evaluate the contribution of the prediction in the optimization phase, we make the following assumption:

The routes should have a past starting time. By these means, we can access the actual recorded fuel prices, that a vehicle driver observes, when she meets the current gas station. Thus, we can measure the optimal refueling policy, i.e the refueling policy, that the optimizer gives as output, when the fuel prices are **known**.

The format of the routes, as these are given to the recommender system, contains the following characteristics:

Table 3.1: Route Format

Field	Value Type
Station-ID	String
Brand	String
Latitude	Float
Longitude	Float
ETA	Timestamp

That means that every record of gas station, as a node of each route has four fields, Station-ID, Brand, Latitude, Longitude and Estimated Time of Arrival (ETA). The last field is the timestamp that denotes the accurate time point that the vehicle will meet the gas station.

Unification of Optimization with Prediction

Using the fields of each record, we can feed the pretrained prediction models, in order to get estimated fuel prices for the *Estimated Time of Arrival*. In the next figure, we show the two different structures of prediction for **local** and **general** model:

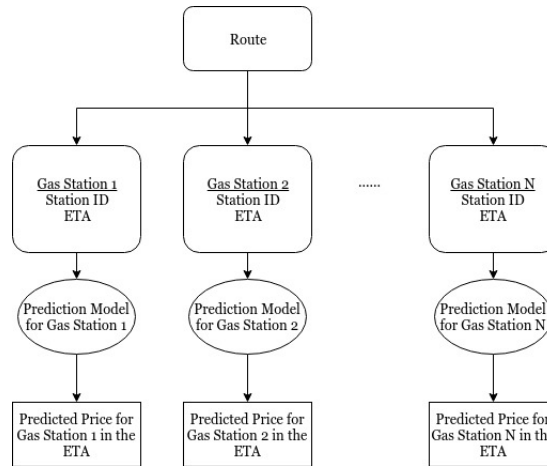


Figure 3.2: Route Definition with Local Model Prediction

As we can see in the figure above, we use N pretrained prediction models, in order to estimate the fuel prices for the *Estimated Time of Arrival*.

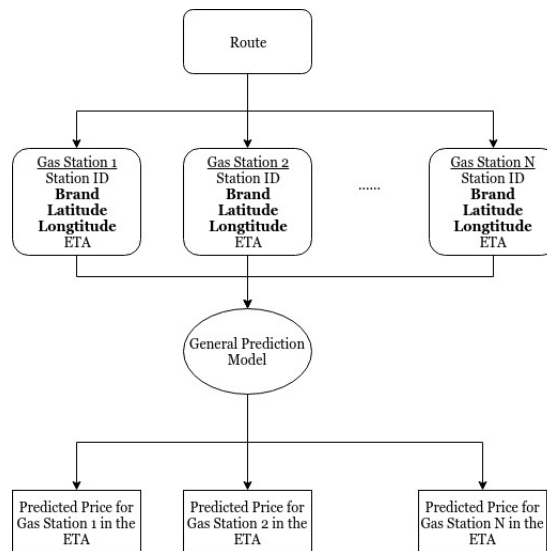


Figure 3.3: Route Definition with General Model Prediction

Here, we use **one** prediction model to estimate the fuel prices for all gas stations!

For our experimentation, we used different configurations of routes, including different locations across the land of Germany. These routes are characterized by different features. Specifically, the different configurations of the examined routes are described in the following table:

Table 3.2: Route Information

route information	distance (km)	travel time (minutes)	number of gas stations
min	100	80	20
max	1000	750	300

As we can see from the range of the kilometers and the travel times, these routes can describe trips from private or transportation vehicles.

Specifically, the set of the examined routes is as follows:

- **Small routes:** 5 different routes with total distance $< 200km$
- **Medium routes:** 5 different routes with $200km < \text{total distance} < 400km$
- **Long routes:** 5 different routes with $400km < \text{total distance} < 800km$
- **Very Long routes:** 5 different routes with $800km < \text{total distance} < 1000km$

Next, we show two examples of the examined routes in the land of Germany:

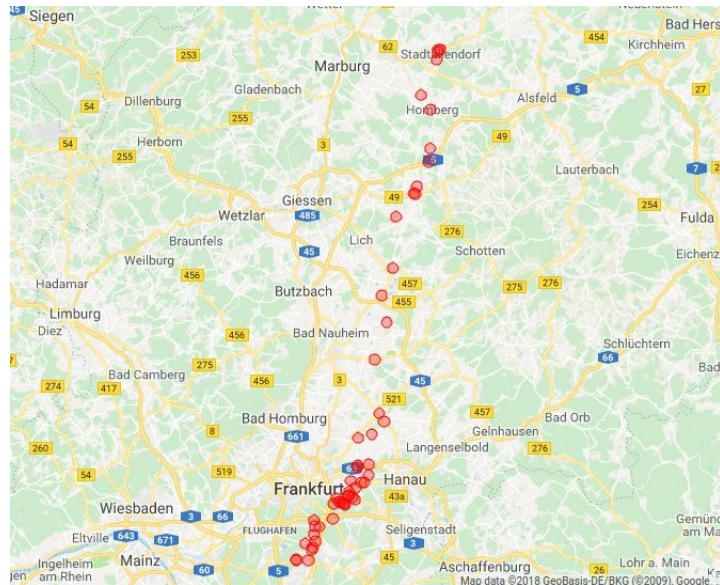


Figure 3.4: Examined Route with 50 gas stations

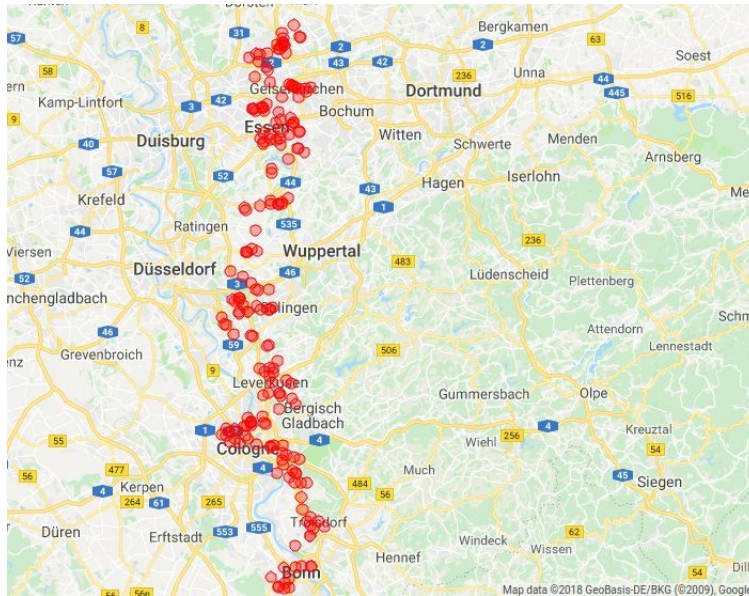


Figure 3.5: Examined Route with 150 gas stations

3.5 Evaluation of the prediction-based optimization

For our experimentation, we are going to use the set of routes, as this was defined previously. In order to evaluate the contribution of the prediction in the optimization step, we need to compute the total refueling cost for each route according to 3 cases, as shown in the next figure:

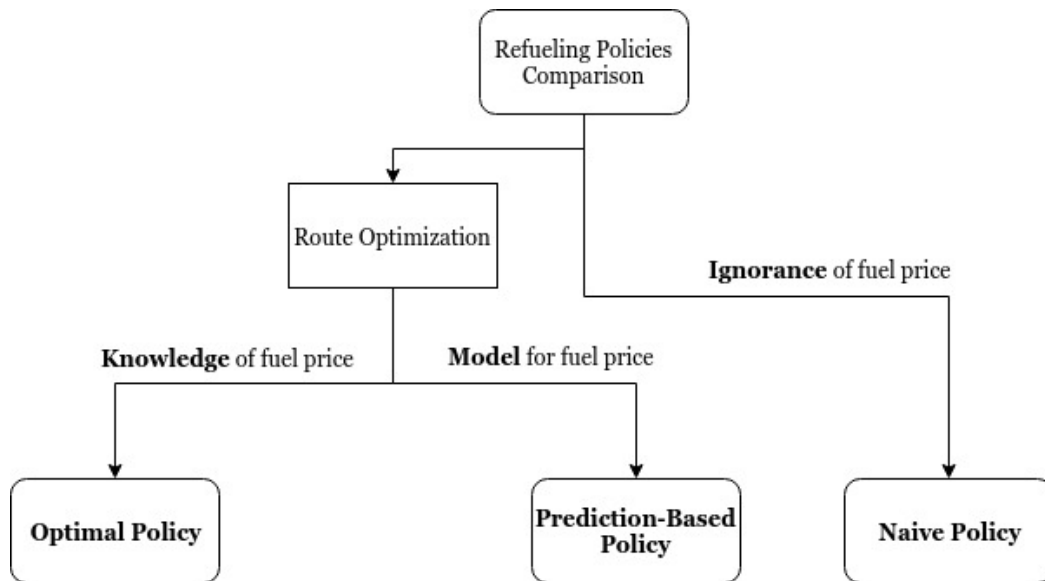


Figure 3.6: Refueling Policies Comparison

These 3 cases are analyzed in the next section:

3.5.1 Policies

We define the following variables:

- **Refueling policy:** $F = \{f_0, \dots, f_{n-1}\}$
- **Actual Fuel Prices:** c_i
- **Total refueling cost:** $R = \sum_0^{n-1} f_i * c_i$

, where i denotes the i th gas station of each route.

Using these variables we define the following 3 policies:

1. **Optimal Refueling Policy:** This is the output vector of our optimizer, if we could know *a priori* the fuel price of every gas station, that the vehicle is going to meet.
2. **Prediction-Based Refueling Policy:** This is the output vector of our optimizer, after the prediction of the fuel price in the specific timestamp for every gas station, that the vehicle is going to meet.
3. **Naive Policy:** This is the vector that the algorithm of the section 3.3 provides us, which simulates the normal behavior.

For these policies, the correspondent refueling costs are the following:

1. **Refueling Cost for Optimal Policy:** $R_{optimal} = \sum_0^{n-1} f_i^{optimal} * c_i$, where the policy vector has been generated according to the actual values.
2. **Refueling Cost for Prediction-Based Policy:** $R_{predicted} = \sum_0^{n-1} f_i^{predicted} * c_i$, where the policy vector has been generated according to the predicted values.
3. **Refueling Cost for Naive Policy:** $R_{naive} = \sum_0^{n-1} f_i^{naive} * c_i$, where the policy vector has been generated according to a setting, where we do not have knowledge for the fuel prices, so we assume **same price** for all gas stations as input to the algorithm.

As we can see from above, we use the **actual** fuel prices for the comparison of the three different policies.

3.6 Implementation Details

:

For the optimization phase we used the following:

- **Gas Station Information Extraction:** PostgreSQL and psycopg2.
Psycopg is a PostgreSQL adapter for the Python programming language.
- **Optimization Algorithm:** Python 3.4
- **Recommender System:** Python 3.4

Chapter 4

Results

4.1 Introduction

In this chapter we present the results from the two phases, the **prediction** and the **optimization**. Firstly, we focus on the prediction accuracy of the different estimators in the **local** and the **global** model. Specifically, we present and comment on the estimator performance for the 2 different approaches of the used features.

Consequently, we present the performance of the combination of prediction and optimization for a set of routes, where the total refueling cost is evaluated.

4.2 Prediction - Local Model

Firstly, we are going to show the prediction accuracy of the different estimators for the case of single time series per model (aka. the local model). For the whole set of experimentations below, we used the configuration below:

- Training/Testing Set = 1000 randomly chosen gas stations across Germany ($\approx 8\%$ of the whole set of gas stations in Germany)
- Validation Scheme = The scheme as described in section 2.4.3.
- Target Variable = E5 Price * 1000. We do not retain the raw measurement E5 Fuel Price, so as to make the Root Mean Squared Errors more visualizable.
- Scaling Method = Standardization, as described in section 2.4.3.

4.2.1 Embedding Vector

During our experimentation using the TSA approach, we had to define the length of the embedding vector and consequently the size of the feature vector. We remind that the embedding vector is the vector of the lagged variables of the time series. That means we use as features the previously observed fuel prices.

In the next figure, we present the prediction errors of the 4 estimators (with fixed configurations) for different sizes of embedding vectors.

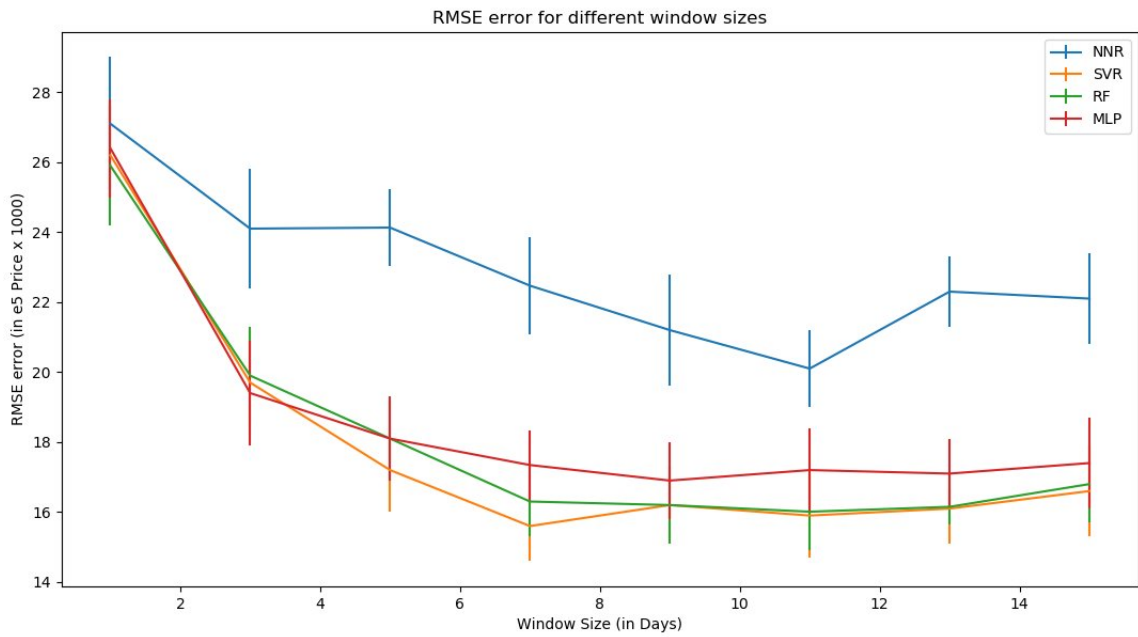


Figure 4.1: Influence of embedding vector

As we can see from the figure above, we have a decrease of RMSE until the 7-day window and then this decrease becomes smaller. Considering that, growing window size we have a bigger feature vector and consequently slower prediction, we can choose as an optimal look-back window size $p = 7$ days (which correspond to one week). That means that the feature vector size (for the TSA approach) should be $24 * 7 = 168$ data points. Thus, for the next measurements and experiments of the TSA approach, we are going to use the $24 * 7 = 168$ previously observed values as the feature vector.

4.2.2 Prediction Horizon

In this section, we examine how the recursive strategy affects the multi- step forecasting, when we follow the Time Series Analysis approach. Specifically, we use the Support Vector Machine Estimator in order to present the Root Mean Squared Error, as this occurred for different prediction horizons:

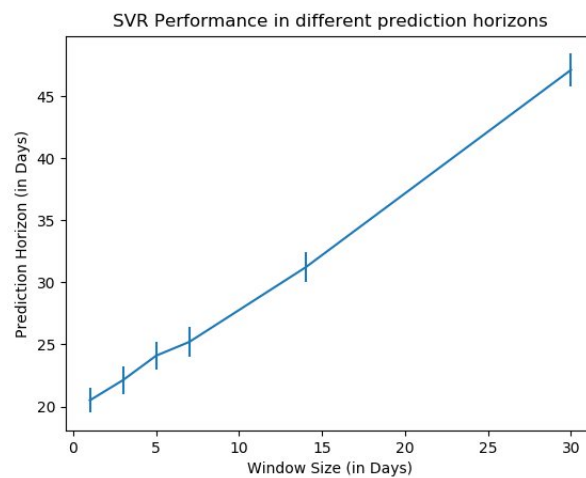


Figure 4.2: RMSE performance for different prediction horizons

As we can see, the recursive strategy integrates an accumulated error in the predictions. Specifically, while on the 1-day prediction horizon we can get an average $RMSE = 20.8$ units, on the 30-days prediction horizon we get an average $RMSE = 45.23$. This is an indication, that using the TSA approach and multi-step forecasting, could not be useful, if we need prediction in the long term. This fact will be verified in the next section, when we compare the performance of DaF and TSA approach.

4.2.3 Parameter Tuning

In order to compare the behavior of the different machine learning estimators in the problem of fuel price prediction, we firstly had to find the best free parameters of each one. For this reason, we proceeded into a grid search on the set of free parameters. We note that the set of the optimal free parameters can be **different** for the two approaches, the static features approach (where we export features from the date) and the time series analysis approach.

Time Series Analysis Approach

We present below the results for the TSA approach. We note that the parameter tuning for the MLP is shown in the **Appendix**, due to the large number of combination of parameters.

Firstly, we show the performance of k-Nearest Neighbor:

Table 4.1: k-NN Parameter Tuning - TSA - Local Model

# Neighbors	Uniform		Inverse Distance	
	RMSE	MAPE	RMSE	MAPE
10	35.91	2.57%	26.81	2.13%
20	30.15	2.31%	24.13	1.92%
50	28.12	2.29%	22.47	1.79%
100	28.01	2.26%	22.51	1.80%

We can see that the **weight** function: Inverse Distance and $k = 50$ is the best configuration for this approach.

Next, we present the performance of Random Forest:

Table 4.2: RF Parameter Tuning - TSA

# Estimators	RMSE	MAPE
10	19.67	1.43%
20	19.34	1.28%
50	18.23	1.25%
100	16.30	1.13%
200	16.34	1.18%
500	16.33	1.15%

We can see that 100 regression trees is the best configuration.

The results for the SVR are presented below:

Table 4.3: SVR Parameter Tuning - TSA

Epsilon	Cost C	RMSE	MAPE
0.005	0.1	16.03	1.17%
0.005	0.5	15.98	1.14%
0.005	1.0	18.11	1.25%
0.005	5.0	23.67	1.62%
0.005	10.0	25.13	2.01%
0.01	0.1	16.01	1.17%
0.01	0.5	15.88	1.13%
0.01	1.0	16.97	1.20%
0.01	5.0	20.16	1.33%
0.01	10.0	23.18	1.61%
0.05	0.1	17.32	1.23%
0.05	0.5	15.61	1.07%
0.05	1.0	17.21	1.22%
0.05	5.0	20.67	1.67%
0.05	10.0	22.78	1.82%
0.1	0.1	19.14	1.42%
0.1	0.5	17.89	1.39%
0.1	1.0	17.93	1.48%
0.1	5.0	19.20	1.55%
0.1	10.0	21.51	1.81%
0.2	0.1	27.65	2.32%
0.2	0.5	25.63	2.28%
0.2	1.0	25.96	2.31%
0.2	5.0	26.77	2.37%
0.2	10.0	26.50	2.32%

As we can see from the table above, the pair of optimal parameters for **SVR** is $C = 0.5$ and $\epsilon = 0.05$. Specifically, we can see that both the average RMSE and MAPE metrics are the lowest for this configuration.

Date as Features Approach

We follow the same experimentation for DaF approach. We note again that the parameter tuning for the MLP is shown in the **Appendix**, due to the large number of combination of parameters.

Table 4.4: k-NN Parameter Tuning - DaF - Local Model

# Neighbors	Uniform		Inverse Distance	
	RMSE	MAPE	RMSE	MAPE
10	47.16	3.54%	43.15	3.21%
20	45.65	3.49%	39.91	3.13%
50	44.21	3.41%	41.29	3.18%
100	44.32	3.45%	42.18	3.25%

Table 4.5: RF Parameter Tuning - DaF

# Estimators	RMSE	MAPE
10	38.13	2.78%
20	32.02	2.57%
50	33.14	2.61%
100	32.24	2.59%
200	32.78	2.62%
500	32.81	2.58%

Table 4.6: SVR Parameter Tuning - DaF

Epsilon	Cost C	RMSE	MAPE
0.005	0.1	40.13	2.94%
0.005	0.5	41.12	3.01%
0.005	1.0	44.78	3.18%
0.005	5.0	42.91	3.33%
0.005	10.0	44.65	3.41%
0.01	0.1	39.01	2.96%
0.01	0.5	39.76	2.88%
0.01	1.0	38.41	2.81%
0.01	5.0	40.67	2.97%
0.01	10.0	39.54	2.88%
0.05	0.1	40.18	2.91%
0.05	0.5	35.55	2.71%
0.05	1.0	34.21	2.65%
0.05	5.0	34.58	2.66%
0.05	10.0	36.78	2.82%
0.1	0.1	35.14	2.78%
0.1	0.5	34.78	2.67%
0.1	1.0	34.71	2.66%
0.1	5.0	36.20	2.81%
0.1	10.0	38.15	2.99%
0.2	0.1	41.78	3.13%
0.2	0.5	44.12	3.43%
0.2	1.0	42.01	3.36%
0.2	5.0	45.19	3.51%
0.2	10.0	48.54	3.67%

We can see with bold letters the best configuration for each estimator.

4.2.4 Performance Comparison of Machine Learning Estimators

Selecting the optimal parameters that we found in the previous section, we can now compare the performance of the different estimators for both feature approaches.

We are going to present first in the next table, the configuration of the optimal parameters for each estimator:

Table 4.7: Optimal parameters for each estimator - Local Model

	Random Forest	SVR	K-Nearest Neighbor	MLP
DaF	20 estimators	$e=0.05, C=1.0$	IDW, 20 Neighbors	50 nodes
TSA	100 estimators	$e=0.05, C=0.5$	IDW, 50 Neighbors	200 nodes, 50 nodes

The figure below shows the performance of Random Forest, Support Vector Regression, K-Nearest Neighbors and Neural Network:

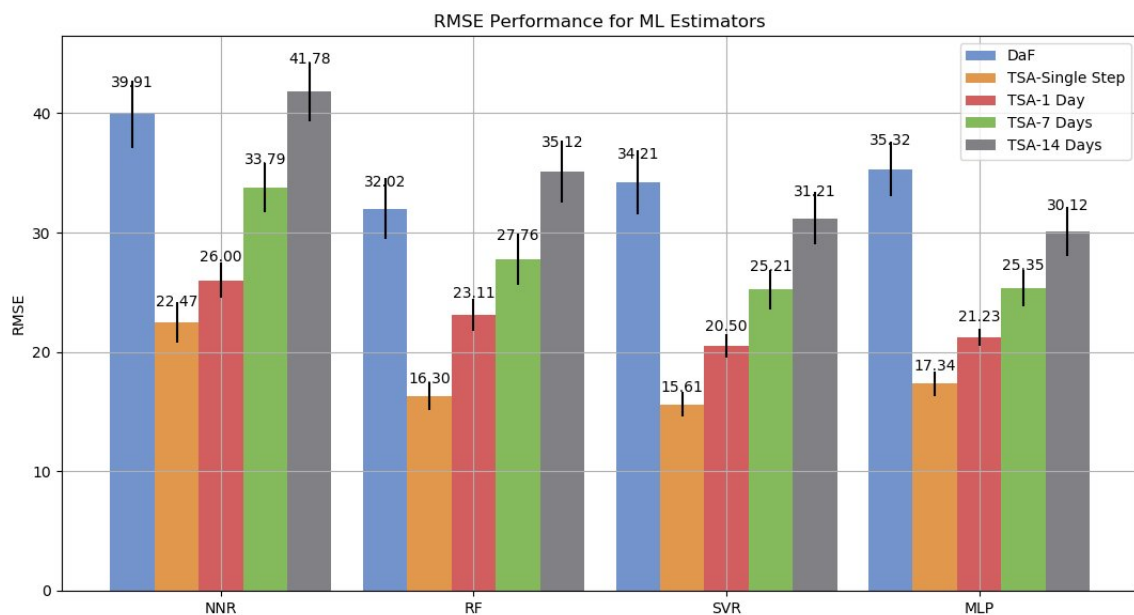


Figure 4.3: RMSE performance for best configurations of NNR, RF, SVR, MLP

Remarks:

- In the approach, where we exploit date information as features, **Random Forest** has the best average prediction accuracy ($RMSE = 32.02$ fuel price units), while the worst performance is shown by K-Nearest Neighbors.
- In the time series analysis approach, **Support Vector Regression** outperforms the rest of the estimators. Specifically, SVR achieves the lowest $RMSE = 15.61$ for a single step prediction. Second best performance is achieved by Random Forest with $RMSE = 16.30$. Again, the worst performance was shown by NNR with $RMSE = 22.47$
- It is noteworthy, using a prediction horizon of 14 days (in TSA approach), the RMSE is getting close (and even surpasses) the RMSE of the DaF approach. That means that an estimator with DaF approach is more appropriate, when we want to make a prediction in the long term.

4.2.5 Visualisation of SVR performance for different forecasting steps

After examining the performance of the different estimators and configurations, we can visualize the performance of the predictor for multiple steps of prediction. We are going to use the gas station with ID = 005056ba-7cb6-1ed2-bceb-60512dc8cd1b. This dataset has 26774 samples, from July 2014 until June 2017.

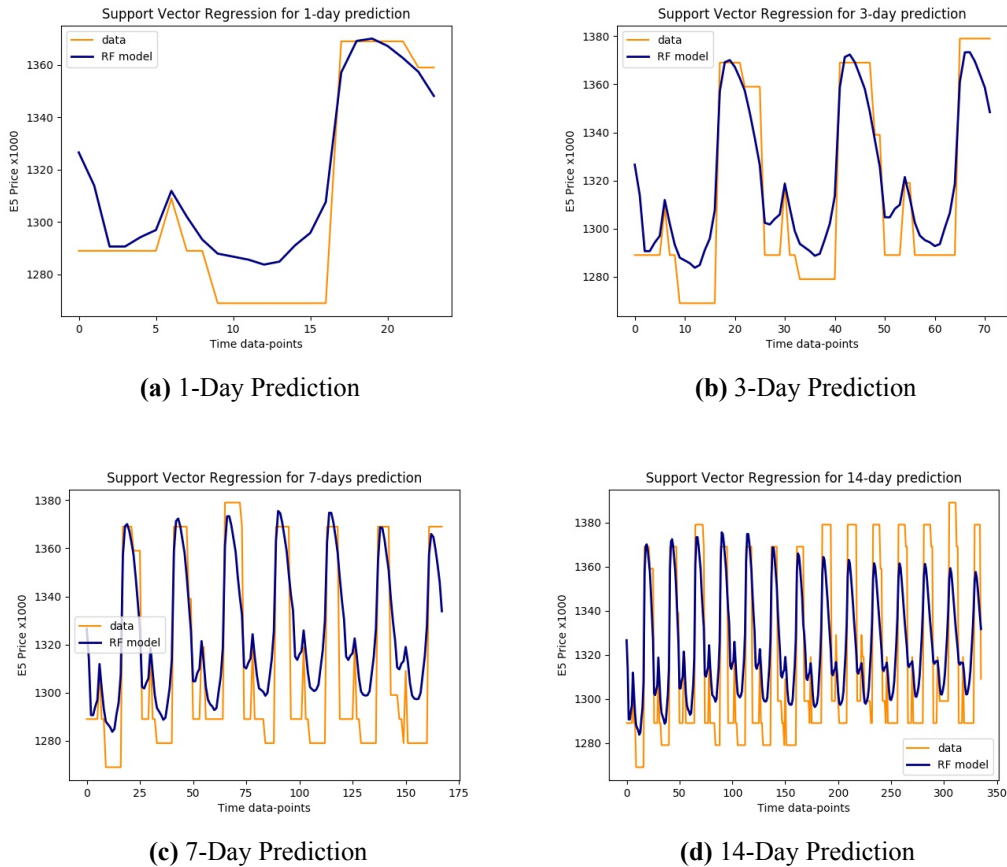


Figure 4.4: Prediction of SVR regression for 1, 3, 7, 14 days

Below we present the results for long term prediction. We can see that the recursive strategy for multiple-step prediction and the nature of the SVM do not respond well for long term forecasting. Specifically, we can observe a convergence of the estimator.

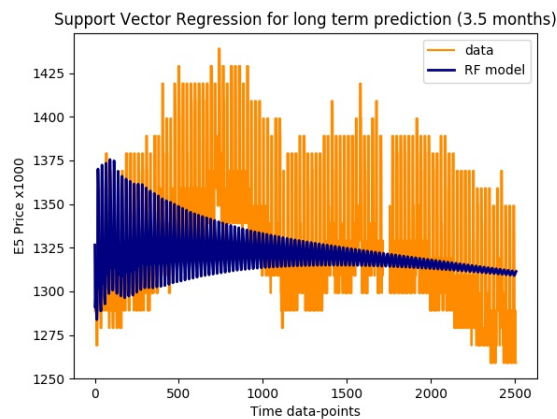


Figure 4.5: SVR Prediction for 3.5 months

From this figure, we can observe the inability of time series approach in combination with SVR to make long-term predictions.

4.3 Prediction - General Model

Now, we are ready to show the results for the general model. In this case, we had the following configuration:

- Training Set = 2500 randomly chosen gas stations across Germany ($\approx 19\%$ of the whole set of gas stations in Germany)
- Testing Set = 250 randomly chosen gas stations across Germany ($\approx 2\%$ of the whole set of gas stations in Germany)
- Validation Scheme = The scheme as described in section 2.4.3.
- Target Variable = E5 Price * 1000. We do not retain the raw measurement E5 Fuel Price, so as to make the Root Mean Squared Errors more visualizable.
- Scaling Method = Standardization, as described in section 2.4.3.

As in the local model, firstly we made a grid-search parameterization for the different estimators in order to find the optimal configurations.

4.3.1 Parameter Tuning

We are going to show again the parameter tuning for the different estimators for the general model of prediction. Due to the fact that we integrate into the feature spatial information, we can **not** assume that the optimal parameters for the local model are the same for the spatiotemporal model.

Time Series Analysis Approach

We present again below the results for the TSA approach.

Table 4.8: k-NN Parameter Tuning - TSA - General Model

# Neighbors	Uniform		Inverse Distance	
	RMSE	MAPE	RMSE	MAPE
10	51.16	4.02%	44.56	3.61%
20	50.67	3.96%	44.14	3.56%
50	54.18	4.17%	41.23	3.17%
100	55.91	4.32%	41.67	3.22%

Table 4.9: RF Parameter Tuning - TSA - General Model

# Estimators	RMSE	MAPE
10	41.65	3.01%
20	35.14	2.82%
50	33.67	2.71%
100	33.15	2.68%
200	33.22	2.67%
500	35.16	2.74%

The results for the SVR are presented below:

Table 4.10: SVR Parameter Tuning - TSA - General Model

Epsilon	Cost C	RMSE	MAPE
0.005	0.1	37.69	3.13%
0.005	0.5	35.16	3.01%
0.005	1.0	42.16	3.31%
0.005	5.0	38.15	3.05%
0.005	10.0	41.16	3.14%
0.01	0.1	37.65	3.03%
0.01	0.5	37.18	3.01%
0.01	1.0	37.12	3.02%
0.01	5.0	36.51	2.91%
0.01	10.0	33.19	2.72%
0.05	0.1	33.18	2.71%
0.05	0.5	33.56	2.75%
0.05	1.0	33.91	2.76%
0.05	5.0	32.81	2.65%
0.05	10.0	32.84	2.66%
0.1	0.1	33.14	2.71%
0.1	0.5	32.98	2.68%
0.1	1.0	32.40	2.65%
0.1	5.0	32.64	2.66%
0.1	10.0	32.71	2.71%
0.2	0.1	32.55	2.68%
0.2	0.5	33.15	2.72%
0.2	1.0	32.78	2.68%
0.2	5.0	33.18	2.71%
0.2	10.0	33.01	2.69%

Date as Features Approach

We present again below the results for the DaF approach.

Table 4.11: k-NN Parameter Tuning - DaF - General Model

# Neighbors	Uniform		Inverse Distance	
	RMSE	MAPE	RMSE	MAPE
10	62.65	4.98%	63.76	5.01%
20	60.71	4.78%	57.14	4.52%
50	61.56	4.81%	53.21	4.16%
100	64.12	5.06%	54.13	4.18%

Table 4.12: RF Parameter Tuning - DaF - General Model

# Estimators	RMSE	MAPE
10	46.43	3.21%
20	45.13	3.12%
50	45.34	3.14%
100	43.55	3.04%
200	43.71	3.07%
500	43.69	3.08%

Table 4.13: SVR Parameter Tuning - DaF - General Model

Epsilon	Cost C	RMSE	MAPE
0.005	0.1	48.64	3.54%
0.005	0.5	48.14	3.48%
0.005	1.0	48.51	3.52%
0.005	5.0	47.98	3.41%
0.005	10.0	47.22	3.38%
0.01	0.1	45.18	3.29%
0.01	0.5	45.23	3.29%
0.01	1.0	46.13	3.36%
0.01	5.0	43.82	3.21%
0.01	10.0	45.67	3.13%
0.05	0.1	42.55	3.11%
0.05	0.5	42.13	3.10%
0.05	1.0	41.11	3.03%
0.05	5.0	41.33	3.05%
0.05	10.0	41.32	3.05%
0.1	0.1	41.67	3.07%
0.1	0.5	41.54	3.05%
0.1	1.0	42.19	3.11%
0.1	5.0	41.95	3.09%
0.1	10.0	43.25	3.24%
0.2	0.1	41.87	3.08%
0.2	0.5	42.34	3.14%
0.2	1.0	42.67	3.56%
0.2	5.0	42.78	3.58%
0.2	10.0	42.55	3.58%

4.3.2 Performance Comparison of Machine Learning Estimators

Following the same experimentation, as in the Local Prediction model, we had the comparison results, as they are shown in the figure below. We note again that we selected the optimal parameters, as these were given by the parameter tuning for both approaches.

These optimal parameters are shown in the following table:

Table 4.14: Optimal parameters for each estimator - General Model

	Random Forest	SVR	K-Nearest Neighbor	MLP
DaF	100 estimators	$e=0.05, C=1.0$	IDW, 50 Neighbors	50 nodes
TSA	100 estimators	$e=0.1, C=1.0$	IDW, 50 Neighbors	200 nodes, 50 nodes

Using these optimal parameters, we now can show the performance comparison:

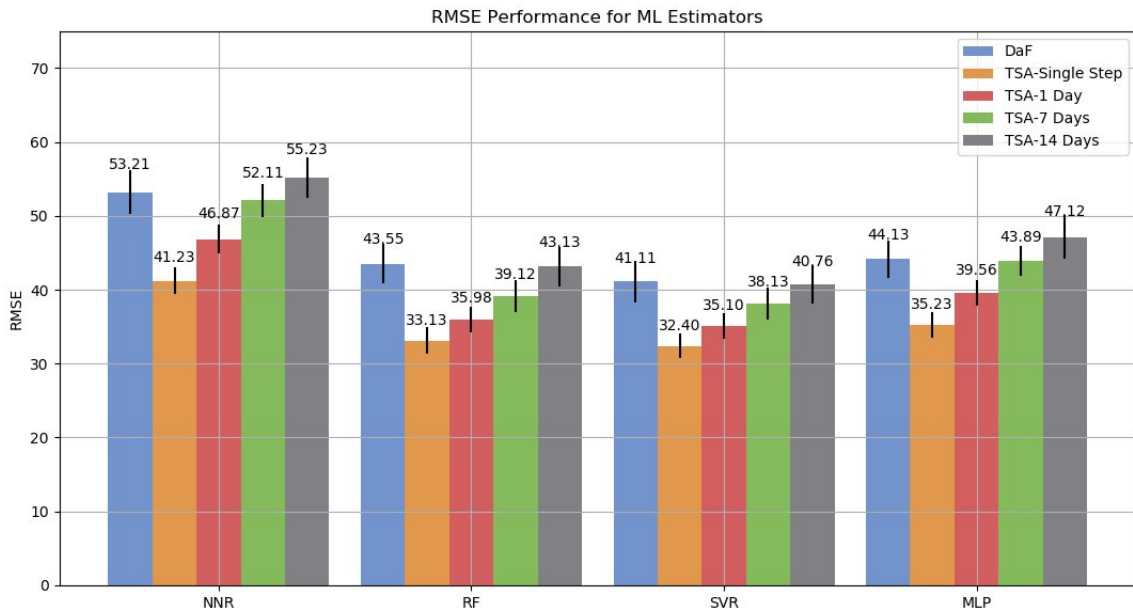


Figure 4.6: RMSE performance for best configurations of NNR, RF, SVR, MLP

Remarks:

- A first comment that we can make here is the lower accuracy that the estimators have in general in comparison with the local model. We can claim that local model prediction outperforms general model.
- In contrast to the local model, we observe that Support Vector Regression outperforms the other estimators in the DaF approach with $RMSE = 41.11$. The Random Forest estimator follows with $RMSE = 43.55$ and the highest RMSE was made by once again k-NN ($RMSE = 53.21$).
- As in the local model, when we use the TSA approach, **Support Vector Regression** again outperforms the rest of the estimators with $RMSE = 32.40$.

4.4 Prediction and Optimization

At this point, having found the optimal parameters of every examined estimator and having shown their prediction accuracy, we are ready to examine the performance of the combination of predictor and optimizer.

4.4.1 Optimal Policy

In the next diagram, we show the refueling cost minimization, using the optimization algorithm and assuming that we know the actual fuel prices of the gas stations, that a vehicle will meet.

We note that for this estimation, we compute the **percentage decrease** of the refueling cost. This can be described as follows:

$$\text{Percentage Decrease} = \frac{R_{\text{examined}} - R_{\text{naive}}}{R_{\text{naive}}} * 100\%,$$

where R_{examined} is the refueling cost, as given using the examined policy, either the optimal or the predicted and R_{naive} is the refueling cost, when we do not use the optimization algorithm. By this way, we normalize the refueling cost decrease, as the different routes, that were used, have different kilometers and hence different vehicle refueling needs.

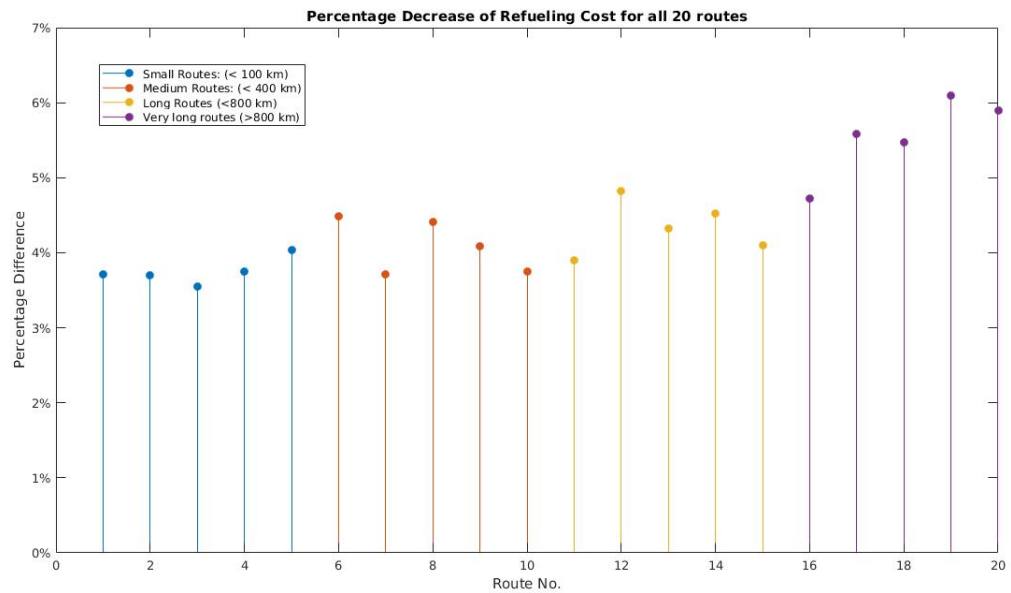


Figure 4.7: Percentage Decrease for all 20 routes

As we can see from the figure above, as long as, we have longer routes, the improvement of using our optimizer is getting greater.

4.4.2 Local Model

Here, we have the case of the local model prediction, using the two different feature approaches, **DaF** and **TSA**. For every examined route, we trained one prediction model for each gas station .

DaF Approach

The first diagram presents the percentage decrease for the different estimators according to the different lengths of routes.

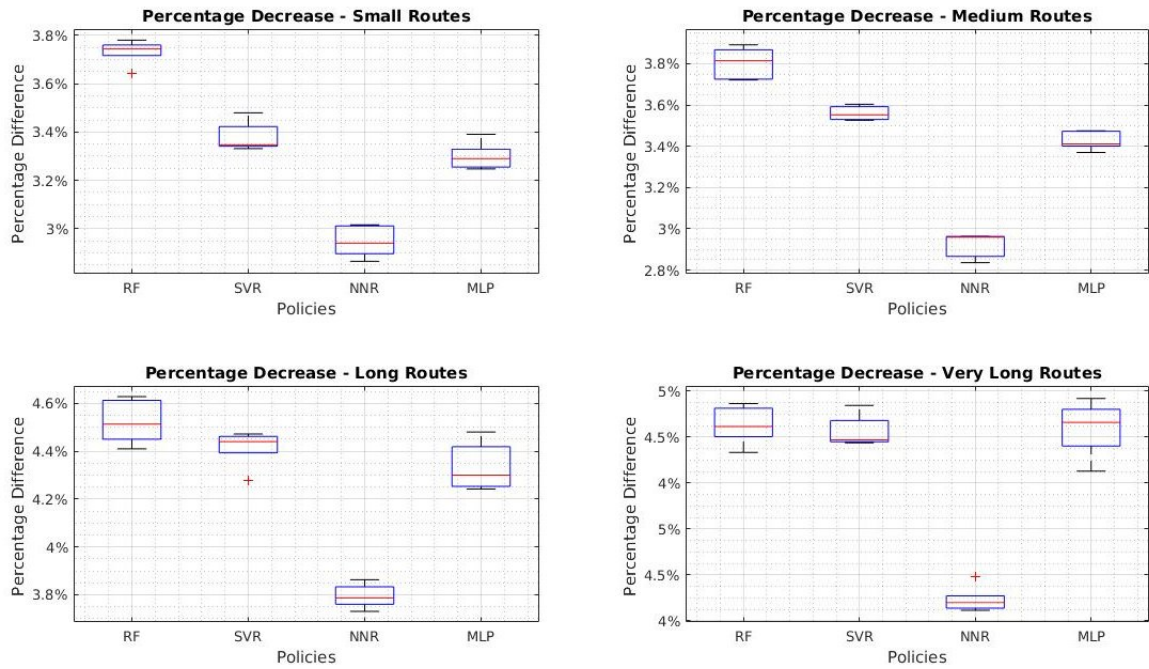


Figure 4.8: Percentage Decrease for DaF Approach per route type

The second diagram show the average percentage decrease for the different estimators:

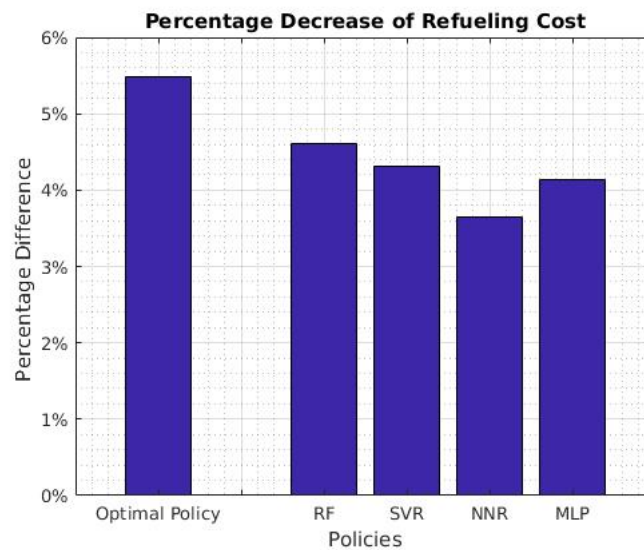


Figure 4.9: Average Percentage Decrease for DaF Approach

As we can see from the figure above, the Random Forest estimator outperforms the others in the DaF approach with an average percentage difference of 4.60% and that means, we have a difference of 0.88% from the optimal policy. This is, also, valid if we check the statistical difference for the different types of routes in figure 4.7.

TSA Approach

Here, we have the results after the same experimentation using the TSA approach:

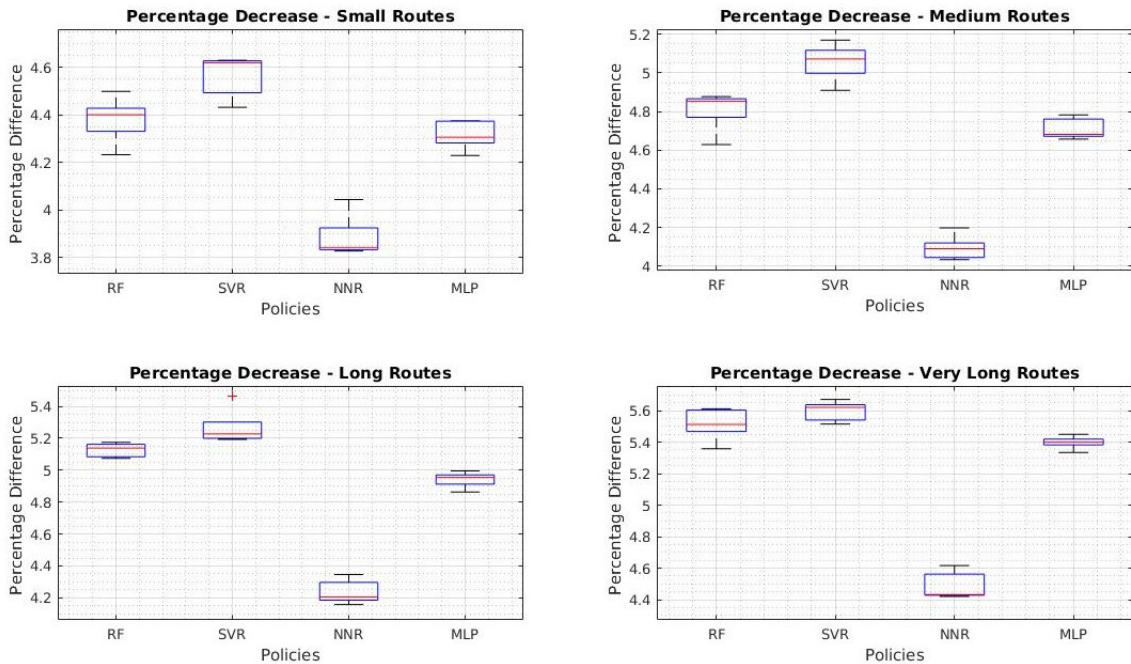


Figure 4.10: Percentage Decrease for TSA Approach

The second diagram show the average percentage decrease for the different estimators:

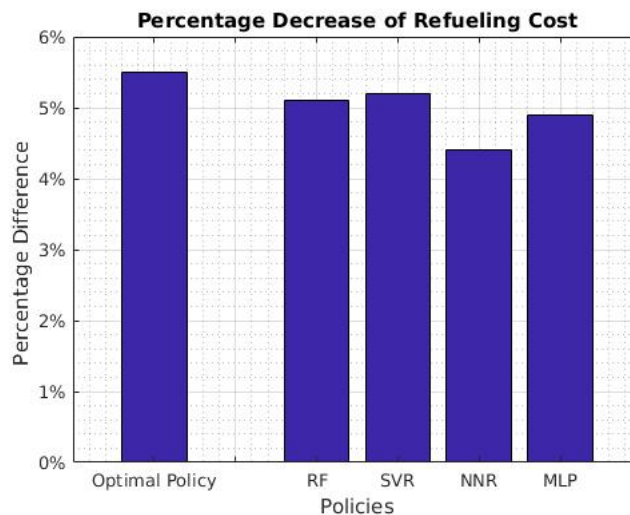


Figure 4.11: Average Percentage Decrease for TSA Approach

In the TSA approach, we can see that we have the largest percentage decrease using the **Support Vector Regression** (a difference of $5.48\% - 5.20\% = 0.28\%$ from the optimal policy).

4.4.3 Global Model

In this case, we have results for the global model prediction, using the two different feature approaches, **DaF** and **TSA**. Specifically, we used the same model for every gas station of each route.

DaF Approach

Again following the same procedure as in local model, we present the results firstly for the DaF Approach:

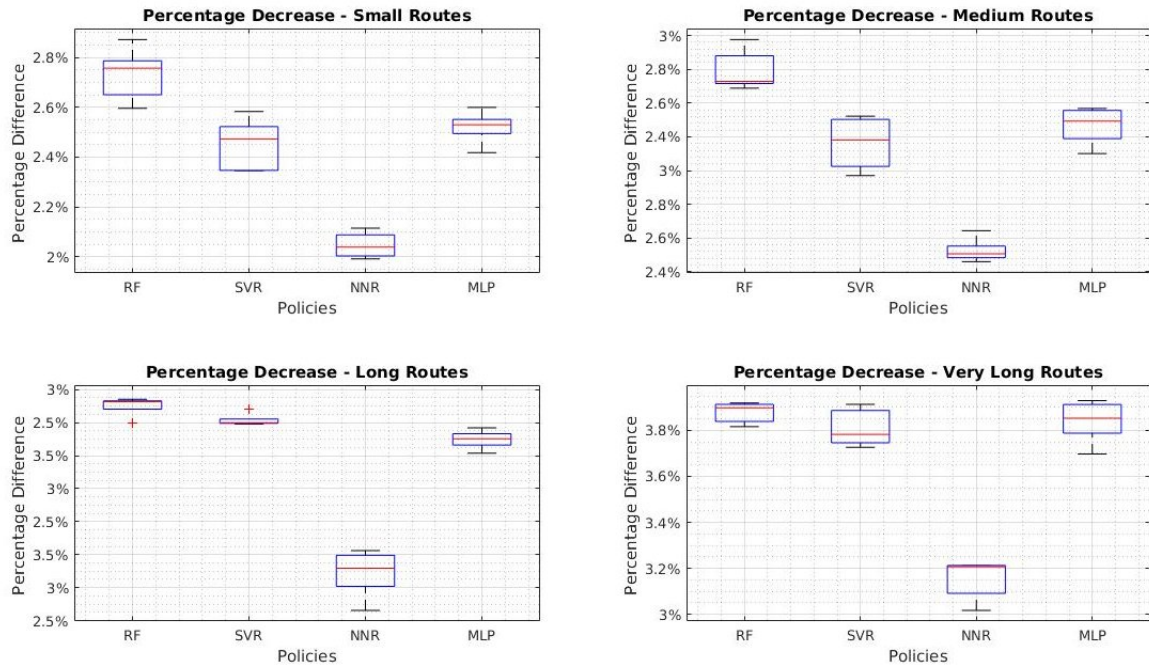


Figure 4.12: Percentage Decrease for DaF Approach

The second diagram show the average percentage decrease for the different estimators:

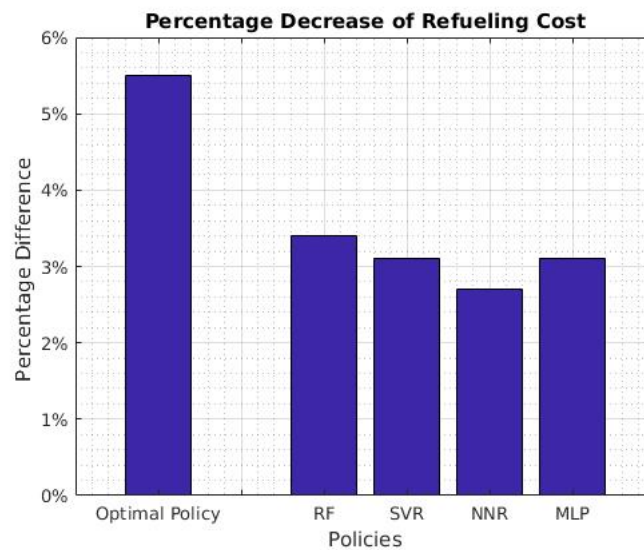


Figure 4.13: Average Percentage Decrease for DaF Approach

As in the local model, we can see again that Random Forest outperforms the other estimators in the DaF approach. We can achieve a 3.42% average percentage decrease and that means, we have a difference of 2.06% from the optimal policy. The second best performance is shown both in SVR and Neural Network, where using SVR we achieve a 3.12% average percentage decrease (a difference of 2.36% from the optimal policy), while using Neural Network we achieve a 3.15% average percentage decrease (a difference of 2.33% from the optimal policy)

TSA Approach

Once again, we give the results for the TSA approach in the global model :

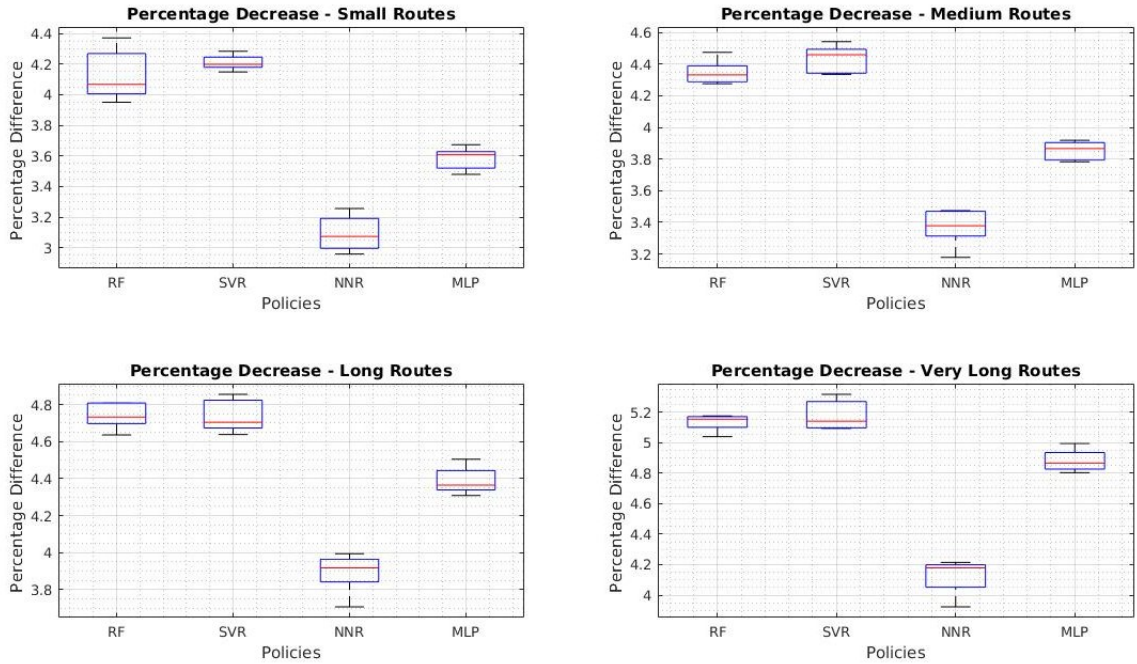


Figure 4.14: Percentage Decrease for TSA Approach

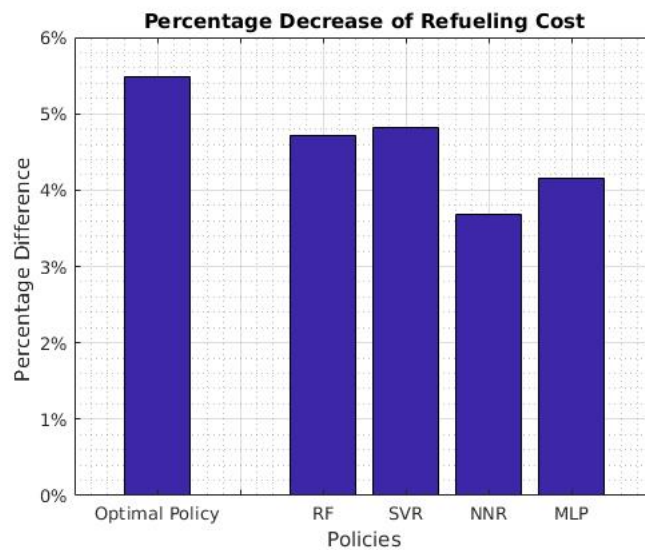


Figure 4.15: Average Percentage Decrease for TSA Approach

In the TSA approach, we can see that Support Vector Regression outperforms the other estimators with an average percentage decrease of 4.82% (with a difference of 0.66% from the optimal policy).

Chapter 5

Conclusions and Further Work

5.1 Conclusions and Discussion

In the present work, we propose a recommendation system, which aims to minimize the refueling cost for a given vehicle route. This system employs pretrained prediction models, in order to acquire knowledge about the fuel prices of the gas stations. Specifically, we focus on routes and gas stations in Germany, where the fuel prices are fluctuating intensively during the day. For the prediction, we conducted experimentations on different machine learning estimators and different approaches.

We developed 2 types of approaches. In the first type, we care about the number of gas stations, that a prediction model expresses. Specifically, on the one hand we develop **local** models, which are personalized for every gas station and on the other hand **general** models, which exploit spatiotemporal data, in order to describe patterns from the whole set of gas stations. The second type of approaches concerns the feature vector, that we are going to use. Here, on the one hand we follow the **DaF** (Date as Feature) approach, where we use "static" time information as features of our model, while on the other hand we follow the **TSA** (Time Series Analysis) approach, where we use the previously observed prices as features for our model.

As soon as we have trained appropriate prediction models for the fuel prices, we use them into our optimizer, in order to get estimations of prices for specific date and time, according to the given route. Based on these estimations, we suggest the optimal refueling policy, in order to minimize the vehicle refueling cost.

5.1.1 Prediction Accuracy

Observing the whole experimentation in the previous chapter, we can make the general remark that the winners estimators are **Support Vector Regression** and **Random Forest**. Specifically, in the Local Model, SVR outperformed the others in the TSA approach with single-step $RMSE = 15.61$ units, while RF outperformed the others in the DaF approach with $RMSE = 32.01$ units. On the other hand, in the General Model, we conclude that in both DaF and TSA approach SVR outperformed the others with $RMSE = 41.11$ units and single-step $RMSE = 32.40$ respectively.

Moreover, it was made clear that the local model can achieve higher accuracy than the general model. However, we should not forget the memory efficiency, that the general model provides.

5.1.2 Financial Time Series - Fuel Prices

It is obvious that a financial time series like the fuel price history cannot be accurately modeled using only temporal data or lagged variables, as we did in our case. Some other very crucial factors that affect the fuel and gasoline price are the following:

- The crude oil price

- The refining costs and profits for the transformation of crude oil into gasoline and diesel, so that they are compatible with the vehicles.
- The distribution and the marketing
- The taxes

In the case that we need an accurate model of prediction for the gasoline/diesel prices evolution in the long term, we should integrate into our model such extraneous features, which indeed affect mainly the trend of the fuel prices.

However, our goal is to develop a model which can give us good estimations of the fuel prices in the **short term**, in order to get reliable recommendations from our optimizer.

5.1.3 Contribution of Prediction into Optimization

Researching and experimenting into the prediction accuracy of different estimators had as a result to provide us a good model for future fuel prices. By these means, we obtained the appropriate knowledge for our route optimizer, so as to minimize the refueling cost. As the results shown, using the optimization algorithm, which was described in the chapter 3, and knowing a priori the fuel prices for each node of the route, we can achieve an average 5.48% decrease of the refueling cost in comparison with that when we do not have knowledge about the prices.

The important remark is that using the different prediction models, we can achieve an average decrease, which reaches at 5.20% (using the optimal parameters of SVR and the TSA approach for the local model) of the refueling cost. That means that equipped with the appropriate prediction model and without knowing the current fuel price, we can achieve a difference of 0.28% from the optimal policy.

Moreover, it is important to note that in the case of the general model, where we have a single model to predict the fuel prices of all gas stations in each route, we can achieve a percentage decrease of 3.42% for the DaF approach (using the optimal parameters of Random Forest) and 4.82% for the TSA approach.

5.2 Further Work

Through this work, a few interesting remarks and issues for further examination arose. Some of them are the following:

- In the case of Time Series Analysis approach, we used the **recursive** strategy for multi-step forecasting. As we saw in Chapter 4, this strategy combined with the estimator performance achieves a low accuracy in the long term due to the accumulation errors. It would be very interesting to research on the impact of other multi-step ahead forecasting strategies, as the Direct strategy, where we use different models for different prediction horizons or the MIMO strategy, where we preserve the stochastic dependence between the predicted values [22].
- Regarding the Machine Learning estimators that we have used, we can note that there are modern algorithms that can preserve and interpret even better the memory that a model for a time series dataset needs. One characteristic result is the **Recurrent Neural Networks** and specifically the **Long-Short Term Memory** Neural Networks. These structures contain recursive computational units, which could lead to better prediction accuracy. Although a few configurations of them were examined, we did not observe a better performance than the **Multilayer Perceptron**. A further experimentation and parameter tuning of LSTM would be very useful to examine how these structures can improve the accuracy of the simple Neural Networks.

- As we noted on the introduction of this chapter, regarding the features that were used, it would be interesting to examine further how further factors affect the fuel prices. Moreover, regarding the general prediction model, we suggest that an exploitation of the dependence between the different time series themselves could provide better accuracy. Structures of Neural Networks, such as Convolutional Neural Networks are appropriate for time series clustering and classification.
- In the optimization phase, we described the greedy algorithm that we have used for the optimization. According to this algorithm, we define the route, according to the gas stations, from which a vehicle drives by. In a more general context, it is useful to apply the same experimentation, extending the *Simple Fixed-Path Vehicle Routing Problem* to the general **Fixed-Path Vehicle Routing Problem**, where the gas stations are not necessarily nodes of the route. [29]
Furthermore, we made assumptions on the velocity, fuel consumption and fuel capacity of the vehicle that we used in the route benchmarking. It is very interesting to examine how this recommendation system responds to real-life applications, where we have fluctuating velocities and fuel consumptions.

Bibliography

- [1] *Fuel Price Monitor in Germany*
<https://www.spritpreismonitor.de/>
- [2] Manuel Frondel, Colin Vance, Alex Kihm.
Time Lags in the Pass-Through of Crude- Oil Prices – Big Data Evidence from the German Gasoline Market
Ruhr Economic Papers No. 573, 2015
- [3] Rob J Hyndman, George Athanasopoulos. *Forecasting: principles and practice*
<https://www.otexts.org/fpp>
- [4] Hand, D.: *Mining the past to determine the future: Problems and possibilities.*
International Journal of Forecasting, October 2008
- [5] Vapnik V: *The nature of statistical learning theory.*
Springer-Verlag, New York , 1995
- [6] S. S. Haykin: *Neural Networks and Learning Machines.*
Prentice Hall, 2009
- [7] Kyoung-jae Kim: *Financial time series forecasting using support vector machines*
Elsevier Neurocomputing, Volume 55, p. 307 – 319, 2003
- [8] Hasbi Yasin, Rezzy Eko Caraka, Tarno and Abdul Hoyyi
Prediction of Crude Oil Prices using Support Vector Regression (SVR) with grid search – cross validation algorithm
Global Journal of Pure and Applied Mathematics, Volume 12, Number 4, pp. 3009–3020, 2016
- [9] T. Cheng, J. Wang and X. Li: *The Nonlinear Integration of Spatial and Temporal Forecasting by Support Vector Machines*
Fuzzy Systems and Knowledge Discovery, FSKD 2007. Fourth International Conference, 2007
- [10] Hamel, L.: *Knowledge Discovery With Support Vector Machine.*
United State Of America: A John Wiley and Sons, INC., Publication, 2009.
- [11] Smola A.J, *Regression estimation with support vector learning machines.*
Master's thesis, Technische Universitaet Muenchen., 1996
- [12] Xin-Yao Qian, Shan Gao.
Financial Series Prediction: Comparison Between Precision of Time Series Models and Machine Learning Methods
<https://arxiv.org/abs/1706.00948>
- [13] Timo Koskela, Mikko Lehtokangas, Jukka Saarinen, and Kimmo Kaski.
Time Series Prediction with Multilayer Perceptron, FIR and Elman Neural Networks.
In Proceedings of the World Congress on Neural Networks, 1996.

- [14] Guoqiang Zhang, B. Eddy Patuwo and Michael Y. Hu
Forecasting with artificial neural networks: The state of the art
International Journal of Forecasting Volume 14,p. 35–62, 1998
- [15] Juan Manuel Gimeno Illa, Javier Bejar Alonso and Miquel Sanchez Marre.
Nearest-Neighbours for Time Series.
SpringerLink Applied Intelligence, Volume 20,p. 21–35, 2004
- [16] Adela Sasu. *K-Nearest Neighbor Algorithm for univariate time series prediction*
Bulletin of the Transilvania University of Braşov, Vol 5, No. 2, 2012
- [17] Nesreen K. Ahmed, Amir F. Atiya, Neamat El Gayar, and Hisham El-Shishiny.
An Empirical Comparison of Machine Learning Models for Time Series Forecasting
Econometric Reviews, Volume 29, Issue 5–6, p. 594–621, 2010
- [18] Enzo Busseti, Ian Osband, Scott Wong.
Deep Learning for Time Series Modeling
<http://cs229.stanford.edu/proj2012/BussetiOsbandWong-DeepLearningForTimeSeriesModeling.pdf>
- [19] F.A. Gers, J. Schmidhuber, F. Cummins
Learning to forget: continual prediction with LSTM
ICANN '99, p. 850 – 855, 1999
- [20] Kline, D.M.: *Methods for multi-step time series forecasting with neural networks.*
Peter Zhang, G. (ed.) Neural Networks in Business Forecasting, pp. 226–250. Information Science Publishing (2004)
- [21] Gianluca Bontempi, Souhaib Ben Taieb, Yann-Aël Le Borgne.
Machine Learning Strategies for Time Series Forecasting
Chapter in Lecture Notes in Business Information Processing, January 2013
- [22] Souhaib Ben Taieb, Gianluca Bontempi, Amir F. Atiya, Antti Sorjamaa.
A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition
Expert Systems with Applications 39,p. 7067–7083, 2012
- [23] Christoph Bergmeir, Rob J Hyndman, Bonsoo Koo.
A Note on the Validity of Cross-Validation for Evaluating Autoregressive Time Series Prediction
Computational Statistics and Data Analysis, Volume 120, April 2018,p. 70-83
- [24] Martin Ester, Hans-Peter Kriegel, Jiirg Sander and Xiaowei Xu :*A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*
KDD-96 Proceedings,AAAI, 1996
- [25] Derya Birant, Alp Kut: *ST-DBSCAN: An algorithm for clustering spatial–temporal data*
Elsevier, Data and Knowledge Engineering 60,p. 208–221, 2007
- [26] Shieu-Hong Lin, Nate Gertsch, and Jennifer R. Russell.
A linear-time algorithm for finding optimal vehicle refueling policies
Operations Research Letters, Volume 35, Issue 3, p. 290-296, 2007
- [27] Shieu-Hong Lin: *Finding Optimal Refueling Policies in Transportation Networks*
AAIM, Algorithmic Aspects in Information and Management, p. 280-291, 2008

- [28] Shieu-Hong Lin: *Multi-Objective Constrained Vehicle Refueling Planning: Complexity and Polynomial-Time Approximation Schemes*
 Proceedings of the 2015 IEEE IEEM, 2015
- [29] Samir Khuller, Azarakhsh Malekian, Julian Mestre.
To Fill or Not to Fill: The Gas Station Problem
 ACM Transactions on Algorithms, Volume 7 Issue 3, Article No. 36, 2011
- [30] Yoshinori Suzuki: *A decision support system of dynamic vehicle refueling*
 Elsevier, Decision Support Systems 46, p. 522–531, 2009
- [31] Yoshinori Suzuki: *A decision support system of vehicle routing and refueling for motor carriers with time-sensitive demands*
 Elsevier, Decision Support Systems 54, p. 758-767, 2012
- [32] Yoshinori Suzuki: *A variable-reduction technique for the fixed-route vehicle-refueling problem*
 Elsevier, Computers and Industrial Engineering 67,p. 204–215, 2014
- [33] S. F. Love.: *Bounded production and inventory models with piecewise concave costs*
 Management Science, 20:313-318, 1973
- [34] Szilárd Aradi, Péter Gáspár and Tamás Bécsi: *A Predictive Optimization Method for Energy-Optimal Speed Profile Generation for Trains*
 14th IEEE International Symposium on Computational Intelligence and Informatics,p. 19–21 November, 2013
- [35] Jörn Boehnke: *Pricing Strategies, Competition, and Consumer Welfare Evidence from the German and Austrian Retail Gasoline Market*,2014
- [36] *U.S Energy Information Administration* <https://www.eia.gov/>

Appendix A

Parameter Tuning

A.1 Multilayer Perceptron

We present below the parameter tuning results for the Multi-Layer Perceptron for the 4 different combination of approaches:

A.1.1 Local Model

Firstly, we show the examination of the optimal parameters for the Local Model:

DaF Approach

Table A.1: MLP for DaF Approach - Local Model

Table A.2: 1 Hidden Layer

Number of Nodes	RMSE	MAPE
10	37.45	2.88%
50	35.32	2.76%
100	36.45	2.81%
200	36.78	2.82%

Table A.3: 2 Hidden Layers

Number of Nodes	RMSE	MAPE
[10,10]	37.21	2.85%
[50,10]	37.02	2.85%
[100,10]	37.78	2.91%
[200,10]	37.81	2.90%
[50,50]	37.89	2.93%
[100,50]	37.67	2.89%
[200,50]	38.01	2.95%
[100,100]	37.65	2.87%
[200,100]	38.34	3.03%
[200,200]	38.45	3.10%

For the DaF approach, we achieved the best results for 1 hidden layer and 50 nodes in this hidden layer.

TSA Approach

Table A.4: MLP for TSA Approach - Local Model

Table A.5: 1 Hidden Layer

Number of Nodes	RMSE	MAPE
10	22.59	1.21%
50	22.01	1.17%
100	21.49	1.15%
200	21.33	1.11%
500	21.54	1.14%

Table A.6: 2 Hidden Layers

Number of Nodes	RMSE	MAPE
[10,10]	19.45	1.14%
[50,10]	19.12	1.13%
[100,10]	18.78	1.13%
[200,10]	18.45	1.12%
[500,10]	19.12	1.14%
[50,50]	18.21	1.11%
[100,50]	17.88	1.14%
[200,50]	17.34	1.04%
[500,50]	17.67	1.09%
[100,100]	18.41	1.12%
[200,100]	17.43	1.06%
[200,200]	18.13	1.08%
[500,200]	18.56	1.11%
[500,500]	18.74	1.09%

Table A.7: 3 Hidden Layers for MLP - TSA - Local Model

Number of Nodes	RMSE	MAPE
[10,10,10]	19.54	1.21%
[50,10,10]	19.45	1.17%
[100,10,10]	19.67	1.24%
[200,10,10]	19.58	1.22%
[50,50,10]	19.87	1.26%
[100,50,10]	19.32	1.23%
[200,50,10]	19.01	1.13%
[100,100,10]	19.45	1.17%
[200,100,10]	18.65	1.09%
[200,200,10]	18.74	1.10%
[50,50,50]	19.78	1.24%
[100,50,50]	19.74	1.22%
[200,50,50]	19.43	1.16%
[100,100,100]	19.61	1.19%
[200,100,100]	19.48	1.16%
[200,200,100]	19.65	1.20%
[200,200,200]	19.73	1.23%

As we can see in the TSA approach, the lowest RMSE occurred for 2 hidden layers and 100 nodes for the first hidden layer and 50 nodes on the second hidden layer.

A.1.2 General Model

Now, we present the results for the general model:

DaF Approach

Table A.8: MLP for DaF Approach - General Model

Table A.9: 1 Hidden Layer

Number of Nodes	RMSE	MAPE
10	47.08	3.51%
50	46.45	3.46%
100	44.13	3.38%
200	44.21	3.41%

Table A.10: 2 Hidden Layers

Number of Nodes	RMSE	MAPE
[10,10]	48.78	3.62%
[50,10]	47.23	3.51%
[100,10]	47.56	3.57%
[200,10]	47.33	3.55%
[50,50]	48.81	3.61%
[100,50]	48.34	3.59%
[200,50]	48.21	3.54%
[100,100]	47.78	3.49%
[200,100]	47.67	3.48%
[200,200]	47.93	3.54%

Regarding the DaF approach in the general model, we have 1 hidden layer and 100 nodes in this hidden layer, as the structure of the MLP, which achieved the best performance.

TSA Approach

Table A.11: MLP for TSA Approach - General Model

Table A.12: 1 Hidden Layer

Number of Nodes	RMSE	MAPE
10	38.41	2.98%
50	38.18	2.96%
100	36.34	2.77%
200	36.17	2.75%
500	36.21	2.75%

Table A.13: 2 Hidden Layers

Number of Nodes	RMSE	MAPE
[10,10]	36.74	2.84%
[50,10]	36.67	2.81%
[100,10]	36.21	2.78%
[200,10]	35.74	2.75%
[500,10]	36.56	2.78%
[50,50]	36.13	2.77%
[100,50]	35.78	2.76%
[200,50]	35.23	2.69%
[500,50]	35.66	2.74%
[100,100]	36.12	2.75%
[200,100]	36.04	2.76%
[200,200]	36.59	2.81%
[500,200]	36.34	2.79%
[500,500]	36.36	2.80%

Table A.14: 3 Hidden Layers for MLP - TSA - General Model

Number of Nodes	RMSE	MAPE
[10,10,10]	36.89	2.87%
[50,10,10]	36.93	2.89%
[100,10,10]	36.78	2.84%
[200,10,10]	36.45	2.79%
[50,50,10]	37.15	2.95%
[100,50,10]	36.91	2.88%
[200,50,10]	36.53	2.73%
[100,100,10]	37.16	2.95%
[200,100,10]	36.32	2.69%
[200,200,10]	36.46	2.74%
[50,50,50]	37.67	3.01%
[100,50,50]	37.12	2.93%
[200,50,50]	37.01	2.92%
[100,100,100]	37.15	2.95%
[200,100,100]	37.68	3.00%
[200,200,100]	36.41	2.78%
[200,200,200]	36.88	2.91%

We can see that the TSA approach for both models suggests the same structure and specifically 2 hidden layers with 100 nodes in the first hidden layer and 50 nodes in the second one.



Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών

Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Προβλεπτικός Μηχανισμός Βέλτιστου Ανεφοδιασμού Οχημάτων

Συνδυασμός τεχνικών μηχανικής μάθησης με υπολογισμούς βελτιστοποίησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΓΕΩΡΓΙΟΣ ΔΑΣΟΥΛΑΣ

Επιβλέπων : Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2018



Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών

Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Προβλεπτικός Μηχανισμός Βέλτιστου Ανεφοδιασμού Οχημάτων

Συνδυασμός τεχνικών μηχανικής μάθησης με υπολογισμούς βελτιστοποίησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΓΕΩΡΓΙΟΣ ΔΑΣΟΥΛΑΣ

Επιβλέπων : Νεκτάριος Κοζύρης

Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 22η Μαρτίου 2018.

.....
Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.

.....
Γεώργιος Στάμου
Αναπληρωτής Καθηγητής Ε.Μ.Π.

.....
Γεώργιος Γκούμας
Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2018

.....
Γεώργιος Δασούλας

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Γεώργιος Δασούλας, 2018.
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περιεχόμενα

Περιεχόμενα	5
Κατάλογος σχημάτων	7
Περίληψη	9
Ευχαριστίες	11
1. Εισαγωγή	13
1.1 Μηχανική Μάθηση στην Πρόβλεψη	13
1.2 Τιμές καυσίμων στη Γερμανία	13
1.3 Σκοπός Διπλωματικής	14
1.4 Δομή Εργασίας	15
2. Πρόβλεψη τιμών καυσίμου	17
2.1 Εισαγωγή	17
2.2 Σχετική Εργασία	17
2.3 Θεωρητικό Υπόβαθρο	17
2.4 Μεθοδολογία	18
2.4.1 Εξαγωγή δεδομένων	18
2.4.2 Προεπεξεργασία δεδομένων	18
2.4.3 Τοπικό Μοντέλο Πρόβλεψης	19
2.4.4 Προσεγγίσεις για το Διάνυσμα Χαρακτηριστικών	21
2.4.5 Γενικό Μοντέλο Πρόβλεψης	22
3. Βελτιστοποίηση Δρομολογίων	25
3.1 Εισαγωγή	25
3.2 Σχετική Εργασία	25
3.3 Θεωρητικό Υπόβαθρο	25
3.4 Μεθοδολογία	26
3.4.1 Ορισμός Δρομολογίου	26
3.4.2 Benchmarking Δρομολογίων	26
3.5 Εκτίμηση της βελτιστοποίησης βασιζόμενης στην πρόβλεψη	29
3.5.1 Πολιτικές	30
3.6 Λεπτομέρειες Υλοποίησης	30
4. Αποτελέσματα	31
4.1 Εισαγωγή	31
4.2 Πρόβλεψη - Τοπικό Μοντέλο	31
4.2.1 Διάνυσμα Ενσωμάτωσης	31
4.2.2 Ορίζοντας Πρόβλεψης	32
4.2.3 Σύγκριση Απόδοσης των Εκτιμητών Μηχανικής Μάθησης	33
4.2.4 Οπτικοποίηση της απόδοσης του SVM	34

4.3	Πρόβλεψη - Γενικό Μοντέλο	35
4.3.1	Σύγκριση Απόδοσης των Εκτιμητών Μηχανικής Μάθησης	36
4.4	Πρόβλεψη και Βελτιστοποίηση	37
4.4.1	Βέλτιστη Πολιτική	37
4.4.2	Τοπικό Μοντέλο	38
4.4.3	Γενικό Μοντέλο	40
5.	Συμπεράσματα και Επεκτάσεις	43
5.1	Βασικά Συμπεράσματα και Συζήτηση	43
5.1.1	Ακρίβεια Πρόβλεψης	43
5.1.2	Συνεισφορά της Πρόβλεψης στην Βελτιστοποίηση	44
5.2	Πιθανές Επεκτάσεις	44
	Βιβλιογραφία	47

Κατάλογος σχημάτων

1.1	Διακύμανση Τιμών Καυσίμου σε περίοδο 7 ημερών [1]	14
1.2	Γενική Ροή Εργασίας	14
2.1	Σύνολο Δείγματος	18
2.2	Διάγραμμα Χαρακτηριστικών	18
2.3	Tables of Postgre database	18
2.4	Διάγραμμα ροής για πρόβλεψη με μηχανική μάθηση	19
2.5	Validation σχήμα για χρονικά δεδομένα	20
2.6	Διάγραμμα ροής εργασίας στο γενικό μοντέλο	22
2.7	Χάρτης πρατηρίων στη Γερμανία	24
3.1	Ορισμός Δρομολογίου για Τοπικό Μοντέλο	27
3.2	Route Definition with General Model Prediction	27
3.3	Examined Route with 50 gas stations	28
3.4	Examined Route with 150 gas stations	29
3.5	Σύγκριση Πολιτικών Ανεφοδιασμού	29
4.1	Επίδραση του διανύσματος ενσωμάτωσης	32
4.2	Σφάλμα RMSE για διαφορετικούς ορίζοντες πρόβλεψης	32
4.3	RMSE performance for best configurations of NNR, RF, SVR, MLP	33
4.4	Prediction of SVR regression for 1, 3, 7, 14 days	34
4.5	SVR Prediction for 3.5 months	35
4.6	RMSE performance for best configurations of NNR, RF, SVR, MLP	36
4.7	Percentage Decrease for all 20 routes	37
4.8	Percentage Decrease for DaF Approach per route type	38
4.9	Average Percentage Decrease for DaF Approach	38
4.10	Percentage Decrease for TSA Approach	39
4.11	Average Percentage Decrease for TSA Approach	39
4.12	Percentage Decrease for DaF Approach	40
4.13	Average Percentage Decrease for DaF Approach	40
4.14	Percentage Decrease for TSA Approach	41
4.15	Average Percentage Decrease for TSA Approach	41

Περίληψη

Σκοπός αυτής της διπλωματικής εργασίας είναι η σχεδίαση, η υλοποίηση και η πειραματολογία ενός συστήματος είσηγησης, το οποίο αποσκοπεί στη μείωση του συνολικού κόστους ανεφοδιασμού για δρομολόγια οχημάτων. Μέσω αυτής της εργασίας, προτείνουμε ένα σχήμα βελτιστοποίησης βασιζόμενη στην πρόβλεψη, το οποίο συνδυάζει αποδοτικά αλγορίθμους μηχανικής μάθησης με υπολογισμούς βελτιστοποίησης με τελικό στόχο την παροχή στο χρήστη μιας βέλτιστης πολιτικής ανεφοδιασμού. Αυτό το σύστημα μπορεί να χωριστεί σε 3 μέρη: Την ανάπτυξη μοντέλων πρόβλεψης για τιμές καυσίμων, την υλοποίηση του αλγορίθμου βελτιστοποίησης και την ενοποίηση των δύο διαδικασιών.

Στο πρόσφατο παρελθόν, αρκετή έρευνα έχει πραγματοποιηθεί στον τομέα της ανάλυσης οικονομικών χρονοσειρών μέσω μηχανικής μάθησης. Αυτό είναι αποτέλεσμα της πρόσφατης ανέλιξης της τεχνητής νοημοσύνης σε πραγματικές εφαρμογές. Επίσης, την τελευταία δεκαετία πολλοί βελτιστοποιητές καυσίμων και αλγόριθμοι βελτιστοποίησης δρομολογίων έχουν αναπτυχθεί. Παρολαυτά, αυτοί οι βελτιστοποιητές καυσίμων απαιτούν τη γνώση της τρέχουσας τιμής καυσίμου, προκειμένου να ελαχιστοποιήσουν το κόστος ανεφοδιασμού. Μέσω του παρόντος έργου, προσπαθούμε να συνδυάσουμε αυτά τα δύο πεδία και προτείνουμε έναν προβλεπτικό βελτιστοποιητή καυσίμων, ο οποίος δεν γνωρίζει *a priori* τις τιμές καυσίμων, αλλά παραταύτα αξιοποιεί μοντέλα πρόβλεψης, προκειμένου να αποκτήσουμε μια ακριβή εκτίμηση.

Στο πρώτο κομμάτι, εστιάζουμε την προσοχή μας ανεξάρτητα στο πρόβλημα της πρόβλεψης τιμών καυσίμων στη χώρα της Γερμανίας. Εξετάζουμε 4 τύπους εκτιμητών με χρήση μηχανικής μάθησης, τις Μηχανές Διανυσματικής Υποστήριξης, τα Νευρωνικά Δίκτυα, τα Τυχαία Δάση και τον αλγόριθμο των K-Κοντινότερων Γειτόνων. Συμπεραίνουμε ότι συγκεκριμένες παραμετροποιήσεις των Μηχανών Διανυσματικής Υποστήριξης και των Τυχαίων Δασών υπερτερούν σε απόδοση των υπολοίπων.

Στη συνέχεια, αντιμετωπίζουμε το πρόβλημα της βελτιστοποίησης και τη συνεισφορά της πρόβλεψης στην βελτιστοποίηση. Σημειώνουμε πως με τη χρήση των βέλτιστων μοντέλων πρόβλεψης, μπορούμε να πετύχουμε στο μέγιστο μια ποσοστιαία μείωση 5.20% του κόστους ανεφοδιασμού, που θα είχαμε αν δεν χρησιμοποιούσαμε αυτό το σύστημα είσηγησης. Αυτή η ποσοστιαία μείωση, μάλιστα, είναι μόνο 0.20% μακριά από την μικρότερη τιμή συνολικού κόστους ανεφοδιασμού, που θα μπορούσαμε να πετύχουμε αν γνωρίζαμε *a priori* τις τιμές καυσίμων.

Λέξεις κλειδιά

Μηχανική Μάθηση, Ανάλυση Χρονοσειρών, Τεχνητή Νοημοσύνη, Δρομολόγηση Οχημάτων, Βελτιστοποίηση βασιζόμενη στην πρόβλεψη

Ευχαριστίες

Η παρούσα εργασία εκπονήθηκε υπό την επίβλεψη του καθηγητή Ε.Μ.Π., κ.Νεκτάριου Κοζύρη τον οποίο και ευχαριστώ ιδιαίτερος για την συνολική του στήριξη. Επίσης, αυτό το έργο είναι αποτέλεσμα της συνεργασίας με την εταιρεία NIKI Ltd. Digital Engineering, η οποία εδράζει στα Ιωάννινα και την φίλια εταιρεία TWT GmbH Science & Innovation, η οποία εδράζει στην Στουτγκάρδη Γερμανίας, από όπου μου προτάθηκε και το συγκεκριμένο θέμα. Θέλω να ευχαριστήσω ξεχωριστά τον κύριο Florian Mittag (TWT GmbH) και τον κύριο Alexander Paar (TWT GmbH) για την τεχνική υποστήριξη και την καθοδήγηση που μου έδειξαν και την εμπιστοσύνη να αναλάβω ένα θέμα που αφορά την εταιρεία που μου επέδειξαν. Επίσης, θέλω να ευχαριστήσω και την κυρία Βιβή Ζάψα (NIKI Ltd.) για την συνολική της υποστήριξη. Επιπροσθέτως, θέλω να πω ένα μεγάλο ευχαριστώ στον κύριο Ιωάννη Κωνσταντίνου για την καθοδήγησή του.

Προχωρώντας θέλω να ευχαριστήσω την οικογένεια μου που με στήριξε σε κάθε βήμα και επιλογή μου και μου έδωσε τα κατάλληλα εφόδια.

Τελευταίο αλλά εξίσου σημαντικό ευχαριστώ θέλω να δώσω στους φίλους μου, που ήταν πάντα δίπλα μου και αποτέλεσαν κομμάτι της φοιτητικής μου ζωής και ελπίζω όχι μόνο με τον καλύτερο δυνατό τρόπο.

Γεώργιος Δασούλας,
Αθήνα, 22η Μαρτίου 2018

Κεφάλαιο 1

Εισαγωγή

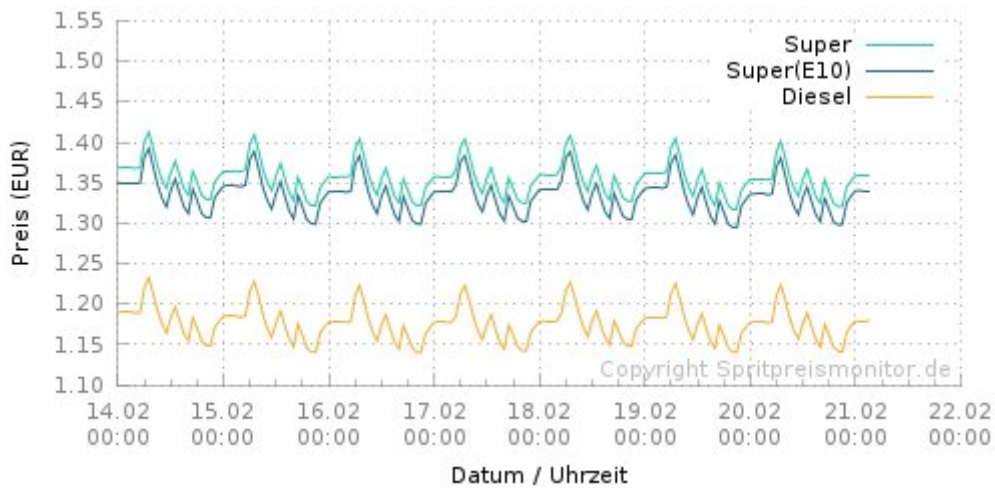
Στη σημερινή ψηφιακή εποχή, η επιστήμη της πληροφορίας και των δεδομένων έχει εισχωρήσει σε κάθε τομέα της βιομηχανίας και της παραγωγής. Ένα πολύ χαρακτηριστικό παράδειγμα είναι η αυτοκινητοβιομηχανία. Τα αυτοκίνητα γίνονται "εξυπνότερα" με το πέρασμα των χρόνων τεχνολογικής εξέλιξης. Παρολαυτά, πολλές κατασκευαστικές διαδικασίες στις αυτοκινητοβιομηχανίες είναι ακόμα κατά μεγάλο βαθμό εξαρτώμενες από ανθρώπινες αποφάσεις βασισμένες στην εμπειρία. Η ανέλιξη των Μεγάλων Δεδομένων μαζί με την αξιοποίηση της μηχανικής μάθησης στις εταιρείες αυτοκινητοβιομηχανίας οδηγούν σε ένα αυξημένο επίπεδο ακρίβειας στην λήψη αποφάσεων και στην υψηλότερη αποδόση. Ένα σχετικό πρόβλημα που θα συζητήσουμε και θα αναλύσουμε στην παρούσα εργασία είναι ο συνδυασμός αλγορίθμων μηχανικής μάθησης και υπολογισμών βελτιστοποίησης στην ελαχιστοποίηση του κόστους ανεφοδιασμού. Συγκεκριμένα, θα ερευνήσουμε δομές εκτιμητών μηχανικής μάθησης για την πρόβλεψη τιμών σε χρονοσειρές, έτσι ώστε να αποκτήσουμε γνώση για πρότυπα διακύμανσης των τιμών καυσίμων στη χώρα της Γερμανίας.

1.1 Μηχανική Μάθηση στην Πρόβλεψη

Αν κοιτάξουμε την έρευνα των προηγούμενων δεκαετιών, μπορούμε να παρατηρήσουμε πως υπάρχει πληθώρα εργασιών για την μοντελοποίηση χρονοσειρών μέσω κλασσικών στατιστικών μεθόδων. Τέτοιες στατιστικές μέθοδοι είναι οι Box-Jenkins ARIMA και GARCH. Παρολαυτά, αργότερα η *Μηχανική Μάθηση* αποτέλεσε έναν εξελισσόμενο τομέα έρευνας της επιστήμης υπολογιστών και στατιστικής και πέτυχε σημαντικές συνεισφορές σε προβλήματα σχετιζόμενα με πρόβλεψη. Η *Μηχανική Μάθηση* αξιοποιεί τεχνητούς αλγορίθμους για την μάθηση προτύπων, που προκύπτουν από υπάρχοντα δεδομένα και παράγουν ένα μοντέλο που είναι ικανό να προβλέψει νέες τιμές με την ίδια δομή.

1.2 Τιμές καυσίμων στη Γερμανία

Από το 2013, όλα τα 14.200 πρατήρια καυσίμων στην Ομοσπονδιακή Δημοκρατία της Γερμανίας άρχισαν να καταγράφουν τις τιμές καυσίμων για όλους τους τύπους (βενζίνη, πετρέλαιο, LPG, CNG) σε κεντρική αρχή καταγραφής (*Benzinpreisstelle*). Γι' αυτή την περίοδο, όλα τα ιστορικά δεδομένα τιμών καυσίμων είναι δημοσίως προσβάσιμα. Αυτή η βάση δεδομένων επιτρέπει την ανάλυση συσχέτισης των τοποθεσιών πρατηρίων, χρόνων παράδοσης, ημερομηνιών, αργιών με την εξέλιξη τιμών καυσίμου. Όπως μπορούμε να δούμε από την εικόνα παρακάτω, οι τιμές βενζίνης και πετρελαίου στην Γερμανία κυμαίνονται έντονα. Υπάρχουν 4 με 9 κινήσεις τιμών κατά μέσο όρο για κάθε μέρα της εβδομάδας. Όπως φαίνεται, τα επίπεδα τιμών εξαρτώνται σημαντικά από την ώρα της ημέρας.

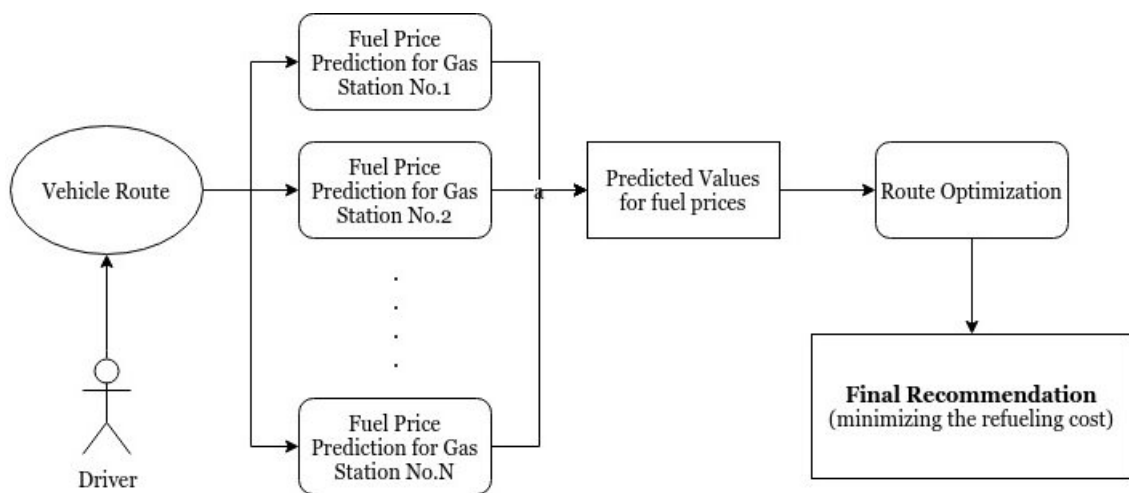


Σχήμα 1.1: Διακύμανση Τιμών Καυσίμου σε περίοδο 7 ημερών [1]

Εκτός από την έντονη μεταβλητότητα που δείχνουν οι τιμές καυσίμου κατά τη διάρκεια της μέρας, μπορούμε να παρατηρήσουμε ότι σε σύντομο χρονικό διάστημα ένας επαναληπτικός κύκλος τιμών ακολουθείται. Αυτό μας δίνει την ευκαιρία να ψάξουμε για πρότυπα σε αυτές τις οικονομικές χρονοσειρές.

1.3 Σκοπός Διπλωματικής

Ο σκοπός της παρούσας εργασίας είναι ο σχεδιασμός και η εκτίμηση ενός συστήματος εισήγησης, το οποίο αποσκοπεί στην ελαχιστοποίηση του κόστους ανεφοδιασμού για δεδομένα δρομολόγια οχημάτων. Συγκεκριμένα, για δεδομένο δρομολόγιο οχήματος θέλουμε να ορίσουμε τις στάσεις σε πρατήρια καυσίμων, έτσι ώστε να ελαχιστοποιήσουμε το κόστος ανεφοδιασμού. Έτσι, λοιπόν, είναι προφανές ότι χρειαζόμαστε να γνωρίζουμε τις τιμές των καυσίμων κάθε πρατηρίου, που το αμάξι συναντάει κατά μήκος του δρομολογίου του. Για αυτή τη γνώση, θα εφαρμόσουμε αλγορίθμους μηχανικής μάθησης προκειμένου να προβλέψουμε τις μελλοντικές τιμές. Σε αυτό το πλαίσιο, θα αντιμετωπίσουμε 2 προβλήματα, το πρόβλημα της πρόβλεψης τιμών καυσίμων στη Γερμανία και το πρόβλημα της βελτιστοποίησης δρομολογίων. Η γενική ροή της εργασίας οπτικοποιείται στην παρακάτω εικόνα:



Σχήμα 1.2: Γενική Ροή Εργασίας

1.4 Δομή Εργασίας

Το έργο αυτό χωρίζεται σε δύο βασικούς κλάδους, τη φάση της **Πρόβλεψης** και τη φάση της **Βελτιστοποίησης**.

- Στο κεφάλαιο 2 εστιάζουμε στην πρόβλεψη τιμών καυσίμου με χρήση τεχνικών μηχανικής μάθησης. Αρχικά, παρουσιάζεται σχετική έρευνα, το θεωρητικό υπόβαθρο και προαπαιτούμενη γνώση. Στη συνέχεια, αναλύονται οι μέθοδοι και οι προσεγγίσεις που χρησιμοποιήθηκαν στο παρόν έργο και τέλος παρουσιάζονται λεπτομέρειες υλοποίησης.
- Στο κεφάλαιο 3 προχωράμε στο κομμάτι της βελτιστοποίησης, όπου γίνεται ανάλυση και υλοποίηση του κεντρικού μας αλγορίθμου. Συγκεκριμένα, ξεκινάμε όμοια με το κεφάλαιο 2 εισάγοντας τα θεωρητικά στοιχεία και στη συνέχεια αναλύουμε τη μεθοδολογία και υλοποίηση. Επίσης, περιγράφουμε τη διαδικασία κατά την οποία τροφοδοτούμε το βελτιστοποιητή μας με τα αποτελέσματα της πρόβλεψης και παρουσιάζουμε τον τρόπο με τον οποίο έγιναν οι εκτιμήσεις μας.
- Στο κεφάλαιο 4 παραθέτουμε και σχολιάζουμε τα αποτελέσματα και από τις 2 φάσεις του έργου. Στο τέλος, παρουσιάζουμε τα βασικά συμπεράσματα του παρόντος έργου.
- Στο κεφάλαιο 5 καταγράφουμε τα συνολικά συμπεράσματά μας και παρουσιάζουμε θέματα για περαιτέρω έρευνα που προέκυψαν στη διάρκεια αυτής της εργασίας.
- Στο παράρτημα, παρουσιάζουμε μια λεπτομερή περιγραφή της παραμετροποίησης.

Κεφάλαιο 2

Πρόβλεψη τιμών καυσίμου

2.1 Εισαγωγή

Στόχος μας είναι να δημιουργήσουμε κατάλληλα μοντέλα πρόβλεψης των τιμών καυσίμου στη Γερμανία, τα οποία να μπορούν να δώσουν ως έξοδο τις προβλεπόμενες τιμές καυσίμου για την ζητούμενη ώρα και ημερομηνία, που θα ζητήσει ο βελτιστοποιητής μας. Η παρούσα εργασία επικεντρώνεται σε στατικά μοντέλα, δηλαδή εκτιμητές, οι οποίοι περνούν από ένα στάδιο εκπαίδευσης και στη συνέχεια χρησιμοποιούνται για πρόβλεψη μελλοντικών τιμών. Όσον αφορά τα χρησιμοποιούμενα χαρακτηριστικά, ακολουθήθηκαν δύο προσεγγίσεις. Κατά την πρώτη προσέγγιση, χρησιμοποιήθηκαν μόνο "στατικά" χαρακτηριστικά του χρόνου, όπως η ώρα, η εβδομάδα, μέρες αργίας κλπ. Κατά τη δεύτερη προσέγγιση, ακολουθώντας την υπάρχουσα βιβλιογραφία των χρονοσειρών στη μηχανική μάθηση, χρησιμοποιήσαμε ως χαρακτηριστικά τις προηγούμενες τιμές καυσίμων. Στην προσέγγιση μέσω χρονοσειρών, μετρήθηκε και συγκρίθηκε η απόδοση των εκτιμητών σε μικρό και μεγάλο βάθος χρόνου. Επίσης, λόγω της ανάγκης να δημιουργηθεί ένα γενικό μοντέλο, που να περιγράφει πρότυπα όλων των πρατηρίων, εφαρμόστηκε μέθοδος συσταδοποίησης, προκειμένου να εξαχθούν χωρητικές πληροφορίες.

2.2 Σχετική Εργασία

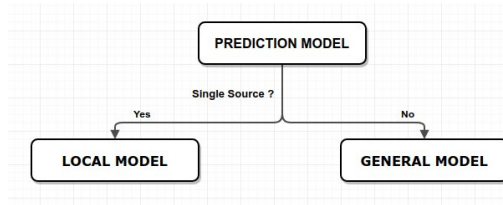
Η πρόβλεψη τιμών οικονομικών χρονοσειρών αποτελεί ανέκαθεν θέμα ενδιαφέροντος στον χώρο της στατιστικής και επιστήμης υπολογιστών. Παρόλο που κλασικά μοντέλα χρονοσειρών όπως ARIMA και GARCH έχουν ερευνηθεί και αποδειχτεί ότι είναι αποτελεσματικά στην πρόβλεψη, οι αποδόσεις τους δεν είναι ακόμα πλήρως ικανοποιητικές. Η Μηχανική Μάθηση, ως ένα εξελισσόμενο πεδίο έρευνας τα τελευταία χρόνια, έφερε πολλές βελτιώσεις σε πρόβλημα παλινδρόμησης (πρόβλεψης) και κατηγοριοποίησης και επίσης είναι πολλά υποσχόμενο στην πρόβλεψη οικονομικών χρονοσειρών. Οι Μηχανές Διανυσματικής Υποστήριξης και διαφορετικές δομές Νευρωνικών Δικτύων χρησιμοποιούνται ευρέως στην πρόβλεψη χρονοσειρών και προβλήματα παλινδρόμησης.[7] [8] [9] [13] Την τελευταία δεκαετία, αναδρομικές δομές Νευρωνικών Δικτύων, όπως τα LSTM και η Βαθεία Μάθηση χρησιμοποιούνται συχνά για την ανάπτυξη μοντέλων σε χρονικά δεδομένα.[18] [19]

2.3 Θεωρητικό Υπόβαθρο

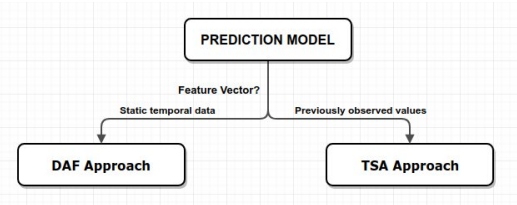
Καταρχάς, θα παρουσιάσουμε την θεωρία για τους αλγόριθμους μηχανικής μάθησης που χρησιμοποιήθηκαν για την πρόβλεψη τιμών καυσίμου. Στη συνέχεια, θα περιγράψουμε την προσέγγιση μηχανικής μάθησης για χρονοσειρές, εισάγοντας τις **αυτοπαλινδρομητικές διαδικασίες** και τις στρατηγικές **πρόβλεψης πολλαπλού βήματος**. Τέλος, θα παρουσιάσουμε τον αλγόριθμο συσταδοποίησης DBSCAN για το γενικό μοντέλο. Όλες οι παραπάνω πληροφορίες για το θεωρητικό υπόβαθρο παραλείπονται στην ελληνική έκδοση και προτρέπουμε τον αναγνώστη να ανατρέξει στην αγγλική έκδοση.

2.4 Μεθοδολογία

Θα περιγράψουμε στη συνέχεια τη μεθοδολογία και τη δομή της πειραματολογίας που ακολουθήσαμε. Συγκεκριμένα, περιγράφουμε τη διαδικασία της εξαγωγής δεδομένων και χαρακτηριστικών, την εκπαίδευση μοντέλου και το validation σχήμα που χρησιμοποιούμε. Οι επόμενες εικόνες δείχνουν τις διαφορετικές προσεγγίσεις που ακολουθήθηκαν: Ακολου-



Σχήμα 2.1: Σύνολο Δείγματος



Σχήμα 2.2: Διάνυσμα Χαρακτηριστικών

θώντας αυτή την κατηγοριοποίηση, κατά την πειραματολογία σε αυτή την εργασία συγκρίνουμε τους 4 συνδυασμούς αυτών των προσεγγίσεων.

2.4.1 Εξαγωγή δεδομένων

Τα δεδομένα, πάνω στα οποία στηριχθήκαμε για την εκπαίδευση των μοντέλων προέρχονται από τις 2 βάσεις δεδομένων σε Postgre της γερμανικής ιστοσελίδας <https://www.tankerkoenig.de/>. Η πρώτη βάση δεδομένων παρέχει πληροφορίες για το χρονικό διάστημα 8.6.2014 - 2.5.2016, ενώ η δεύτερη για το χρονικό διάστημα 2.5.2016 - σήμερα. Σε κάθε μία από τις δύο βάσεις δίνονται δύο tables, ένα για τις γενικές πληροφορίες του κάθε πρατηρίου και ένα για την ιστορία τιμών διαφόρων τύπων καυσίμου για κάθε πρατήριο. Τα πεδία των δύο tables φαίνονται στο παρακάτω διάγραμμα:

Tabelle gas_station

Column	Type	Modifiers
id	uuid	not null
version	character varying(10)	not null
version_time	timestamp without time zone	not null
name	text	not null
brand	text	
street	text	
house_number	text	
post_code	text	
place	text	
public_holiday_identifier	text	
lat	double precision	not null
lng	double precision	not null
ot_json	text	not null

(a) Table of gas station information

Tabelle gas_station_information_history

Column	Type	Modifiers
id	integer	not null
stid	uuid	not null
e5	smallint	
e10	smallint	
diesel	smallint	
date	timestamp(0) with time zone	not null
changed	smallint	

(b) Table of history of gas price

Σχήμα 2.3: Tables of Postgre database

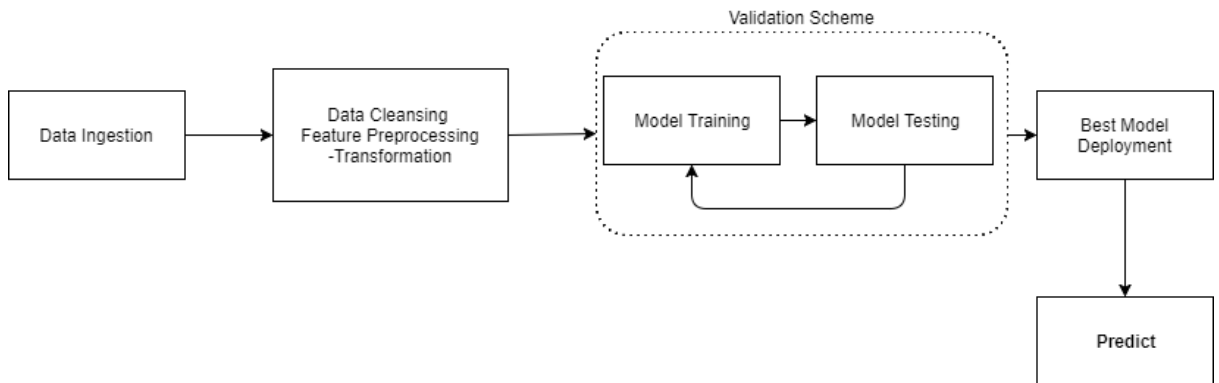
2.4.2 Προεπεξεργασία δεδομένων

Λόγω του ότι το table που αφορά τις τιμές καυσίμων περιέχει μόνο τις καταγραφές αλλαγών τιμών, που συνέβησαν μέσα σε μία μέρα, τα δεδομένα τροποποιήθηκαν ώστε να εκφράζουν ένα σταθερό 24ωρο χρονοδιάγραμμα. Αυτή η τροποποίηση έγινε με σκοπό να έχουμε ισοκατανεμημένα διαστήματα τιμών στο διάνυσμα του χρόνου. Για αυτή την τροποποίηση, σκεφτόμαστε τα εξής:

- Χρονικά μεταξύ δύο αλλαγών τιμής παρατηρούμε την ίδια τιμή καυσίμου.
- Εισάγουμε ακρίβεια 1 ώρας για τις μετρήσεις μας.

2.4.3 Τοπικό Μοντέλο Πρόβλεψης

Σε πρώτη φάση, ερευνήθηκε η περίπτωση, που ένα μοντέλο περιγράφει την εξέλιξη τιμών καυσίμου ενός μόνο πρατηρίου-οντότητας. Εδώ, χρησιμοποιούμε μόνο χρονικά δεδομένα, που αφορούν το εκάστοτε πρατήριο και είναι προφανές ότι κάθε πρατήριο χρειάζεται ένα εξατομικευμένο μοντέλο πρόβλεψης. Αυτή η περίπτωση απαιτεί το κοινώς γνωστό workflow μηχανικής μάθησης για την πρόβλεψη, όπως φαίνεται στην παρακάτω εικόνα :



Σχήμα 2.4: Διάγραμμα ροής για πρόβλεψη με μηχανική μάθηση

Καθαρισμός Δεδομένων/ Επεξεργασία Χαρακτηριστικών

Προεπεξεργασία Χαρακτηριστικών: Όπως παρατηρήσαμε από τα πειράματα, ο μετασχηματισμός των χαρακτηριστικών είναι αρκετά χρήσιμος για την απόδοση των εκτιμητών. Για το λόγο αυτό, χρησιμοποιούμε 2 διαφορετικούς μετασχηματισμούς, σύμφωνα με τον εκάστοτε εκτιμητή:

- **Min-Max** μετασχηματισμός, που ορίζεται ως: $x' = \frac{x - \min(x)}{\max(x) - \min(x)}$. Αυτός ο μετασχηματισμός μεταφέρει κάθε χαρακτηριστικό στο διάστημα $[0, 1]$.
- **Κανονικοποίηση**, που ορίζεται ως: $x' = \frac{x - \bar{x}}{\sigma}$, όπου x είναι το αρχικό διάνυσμα χαρακτηριστικών, \bar{x} είναι ο αριθμητικός μέσος του διανύσματος, και σ η τυπική του απόκλιση. Αυτός ο μετασχηματισμός αφαιρεί τη μέση τιμή και κλιμακώνει σε μοναδιαία διασπορά.

Validation Σχήμα

Από τη στιγμή που διαχειριζόμαστε χρονοσειρές και χρονικά δεδομένα, δεν μπορούμε να εφαρμόσουμε το κλασικό σχήμα cross-validation, λόγω του ότι χρειαζόμαστε τη χρονική εξάρτηση μεταξύ διαφορετικών χρονικών στιγμών. Για το λόγο αυτό ακολουθούμε ένα διαφορετικό σχήμα :

Συγκεκριμένα, ακολουθούμε τα παρακάτω βήματα:

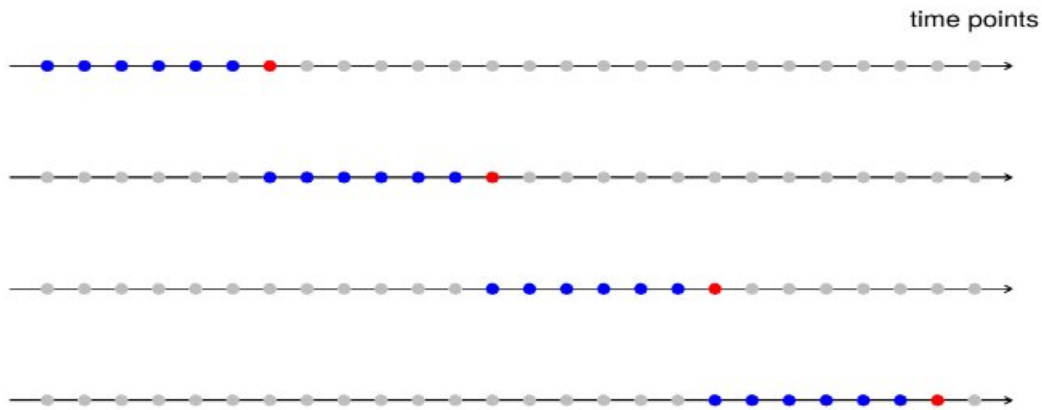
Παράμετροι:

n: αριθμός των επαναλήψεων κοπής

Διαδικασία:

1. Κόβουμε το σύνολο σε n διαδοχικά και μη επικαλυπτόμενα υποσύνολα με ίδιο μέγεθος.
2. Καθένα από αυτά τα υποσύνολα χωρίζεται στη συνέχεια σε **train** και **test** υποσύνολο.

3. Για καθένα από τα n υποσύνολα εκπαιδεύουμε ένα μοντέλο, βασισμένοι στον εξεταζόμενο εκτιμητή και προβλέπουμε τις επόμενες τιμές στο test υποσύνολο. Μετράμε την μετρική σφάλματος.
4. Ύστερα από τις n επαναλήψεις, υπολογίζουμε το μέσο όρο της μετρικής σφάλματος.



Σχήμα 2.5: Validation σχήμα για χρονικά δεδομένα

Εκπαίδευση και Εξέταση Μοντέλου

Σε αυτή τη φάση θα βρούμε την βέλτιστη διαμόρφωση παραμέτρων του προβλέπτη που θα χρησιμοποιήσουμε στο γενικό σύστημα. Για την επιλογή αυτής της διαμόρφωσης θα ερευνήσουμε την απόδοση 4 τύπων εκτιμητών, τροποποιημένοι για πρόβλεψη χρονοσειρών. Αυτοί είναι οι παρακάτω:

- Μηχανές Διανυσματικής Υποστήριξης
- K-Κοντινότεροι Γείτονες
- Τυχαία Δάση
- Νευρωνικά Δίκτυα - Multilayer Perceptron

Για κάθε εκτιμητή πραγματοποιούμε παραμετροποίηση, προκειμένου να βρούμε το βέλτιστο σύνολο παραμέτρων.

Εκτίμηση Απόδοσης στην Πρόβλεψη

Κατά την πειραματολογία, χρησιμοποιήσαμε δύο μετρικές απόδοσης, οι οποίες είναι ευρέως διαδεδομένες για πρόβλημα παλινδρόμησης και συγκεκριμένα χρονοσειρών.

- Η πρώτη είναι **Root Mean Squared Error** (aka. RMSE). Το RMSE των προβλεπόμενων τιμών \hat{y}_i για παρατηρήσεις i ενός παλινδρομητικά εξαρτώμενου μεγέθους y_i υπολογίζεται για n διαφορετικές προβλέψεις ως η τετραγωνική ρίζα του μέσου των τετραγώνων των αποκλίσεων : $RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}}$
- Η δεύτερη μετρική είναι η **Mean Absolute Percentage error** (MAPE). Αυτή η μετρική εκφράζει την ακρίβεια ως ποσοστό και ορίζεται ως εξής:

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

2.4.4 Προσεγγίσεις για το Διάνυσμα Χαρακτηριστικών

Όπως έχουμε ήδη επισημάνει, θα ακολουθήσουμε 2 διαφορετικές προσεγγίσεις σύμφωνα με το διάνυσμα χαρακτηριστικών που χρησιμοποιούνται στο μοντέλο πρόβλεψης. Κατά την πρώτη προσέγγιση, που θα την αποκαλούμε **DaF** ("Date as Feature"), αξιοποιούμε δεδομένα ημερομηνιών ως χαρακτηριστικά. Κατά τη δεύτερη προσέγγιση, **TSA**, συμπεριφερόμαστε στα χρονικά δεδομένα ως χρονοσειρές, υλοποιώντας Ανάλυση Χρονοσειρών μέσω μηχανικής μάθησης.

DaF Προσέγγιση

Κατά αυτή την προσέγγιση, εκμεταλλευόμαστε τις ωριαίες και εβδομαδιαίες επαναληπτικές μεταβολές των δεδομένων. Τα χαρακτηριστικά που χρησιμοποιούμε φαίνονται στον επόμενο πίνακα:

Πίνακας 2.1: Χαρακτηριστικά για την DaF προσέγγιση

	Features	Value Range
1	Day of Month	1 - 31
2	Day of Week	1 - 7
3	Day of Year	1 - 365
4	Week of Month	1 - 5
5	Week of Year	1 - 53
6	Month of Year	1 - 12
7	Year	2014 - 2017
8	Hour	0 - 23
9	Holiday	{0,1}

Όπως φαίνεται, χρησιμοποιούμε ως χαρακτηριστικό μια σημαία διακοπών, που δηλώνει αν η εξεταζόμενη ημερομηνία είναι αργία για τη Γερμανία. Αυτό είναι σημαντικό καθώς κατά τη διάρκεια των διακοπών, οι τιμές καυσίμου ανεβαίνουν. Προκειμένου να παράγουμε αυτή την σημαία, είχαμε πρόσβαση στο δημόσιο ημερολόγιο της Γερμανίας.

TSA Προσέγγιση

Εδώ, διαχειριζόμαστε τα χρονικά δεδομένα ως αυτοπαλινδρομητικές διαδικασίες, οι οποίες περιγράφονται στην αγγλική έκδοση. Συγκεκριμένα, χρησιμοποιούμε τις προηγούμενες παρατηρημένες τιμές καυσίμου ως χαρακτηριστικά, για να παράγουμε τις νέες τιμές. Αυτές οι τιμές καλούνται *Καθυστερημένες Τιμές* και το διάνυσμα χαρακτηριστικών καλείται **Διάνυσμα Ενσωμάτωσης** ή **Look-back window**. Κατά τη διάρκεια των πειραμάτων, έπρεπε να ορίσουμε το μήκος του Διανύσματος Ενσωμάτωσης. Τα αποτελέσματα από αυτά τα πειράματα απεικονίζονται στο κεφάλαιο 4.

Διαφορές

Οι δύο προσεγγίσεις έχουν τα ακόλουθα διαφορετικά χαρακτηριστικά:

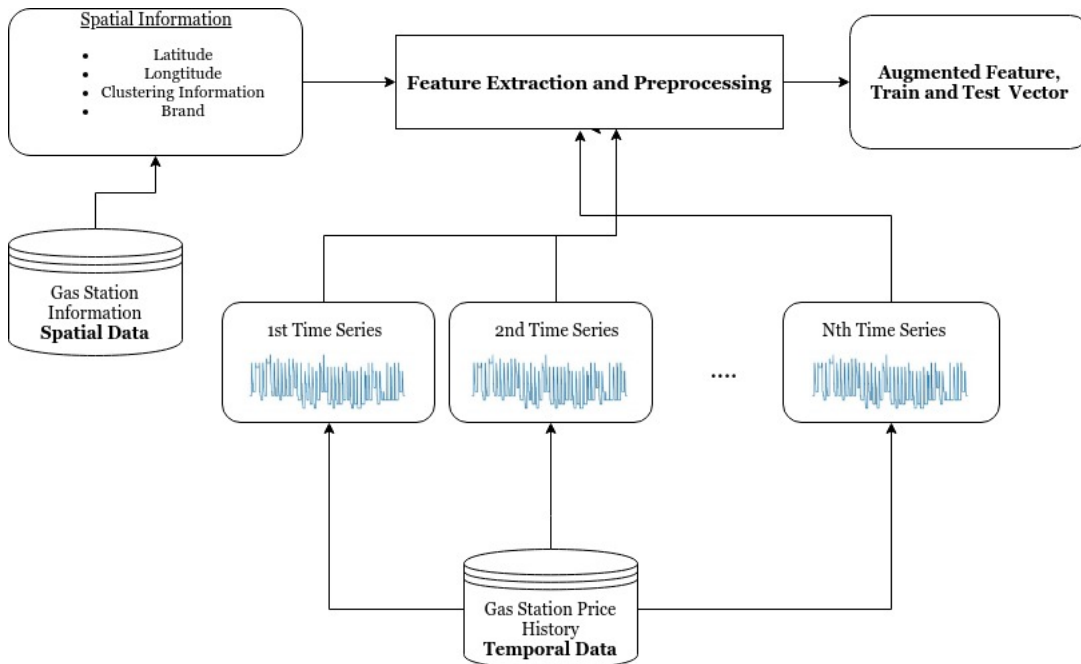
- Χρησιμοποιώντας την DaF προσέγγιση, δεν έχουμε εικόνα των εξαρτήσεων και των συσχετίσεων μεταξύ των καταγεγραμμένων τιμών μεταξύ τους. Η φύση αυτής της προσέγγισης κάνει το μοντέλο πρόβλεψης ανεξάρτητο από προηγούμενες παρατηρημένες τιμές για να προβλέψει νέες.
- Η έξοδος του προβλέπτη χρησιμοποιώντας την TSA προσέγγιση είναι η αμέσως επόμενη τιμή καυσίμου. Αυτό σημαίνει πως έχουμε ορίζοντα πρόβλεψης 1 ώρας. Έτσι,

δημιουργείται η ανάγκη πρόβλεψης πολλαπλού βήματος, όπως αυτό περιγράφεται στο θεωρητικό μέρος (στην αγγλική έκδοση). Από την άλλη μεριά, χρησιμοποιώντας την DaF προσέγγιση, είμαστε ανεξάρτητοι από πρόβλεψη πολλαπλού βήματος, καθώς μπορούμε να κάνουμε πρόβλεψη για κάθε μελλοντικό timestamp.

2.4.5 Γενικό Μοντέλο Πρόβλεψης

Κατά την προσπάθεια να δημιουργήσουμε ένα αποδοτικό τελικό προϊόν, ήρθαμε αντιμέτωποι με το γεγονός πως η πρόσεγγιση της πρόβλεψης της τιμής του καυσίμου ενός πρατηρίου με βάση μόνο την προσωπική του ιστορία, προϋποθέτει την δημιουργία τόσων μοντέλων όσων και των πρατηρίων, κάτι που θέτει θέμα οικονομίας, καθώς ο αριθμός των πρατηρίων καυσίμων στη Γερμανία φτάνει τις 14000. Για το λόγο αυτό προσπαθήσαμε να αναπτύξουμε ένα γενικό μοντέλο πρόβλεψης. Στο γενικό μοντέλο πρόβλεψης, ξεκινήσαμε εμπλουτίζοντας το μοντέλο με γεωχωρικά χαρακτηριστικά, που μπορεί να μας παρέχει η βάση δεδομένων. Ο στόχος μας είναι να δημιουργήσουμε ένα μοντέλο, που να περιγράφει τα πρότυπα αλλαγών των τιμών καυσίμου όχι μόνο ενός πρατηρίου, αλλά όλου του συνόλου πρατηρίων στη Γερμανία. Για αυτό το λόγο, χρησιμοποιώντας τους ίδιους εκτιμητές επεκτείναμε το μέγεθος του διανύσματος χαρακτηριστικών, έτσι ώστε να περιλάβουμε χωρικές πληροφορίες και μεγαλώσαμε το μέγεθος του συνόλου εκπαίδευσης, προκειμένου να εκπαιδεύσουμε το μοντέλο με περισσότερες από μία χρονοσειρές. Αυτή σημαίνει ότι διαχειριζόμαστε τα δεδομένα από διαφορετικές πηγές (πρατήρια) ως εξαρτώμενες μετρήσεις. Επίσης, έχοντας υπόψιν ότι διαφορετικές μάρκες καυσίμου μπορεί να δείξουν διαφορετική συμπεριφορά στην διακύμανση τιμών, ενσωμάτωσαμε στο σύνολο των χαρακτηριστικών και το brand του κάθε πρατηρίου.

Στην εικόνα παρακάτω, περιγράφουμε γραφικά την ενσωμάτωση χωρικών πληροφοριών στο μοντέλο.



Σχήμα 2.6: Διάγραμμα ροής εργασίας στο γενικό μοντέλο

Διάγραμμα Χαρακτηριστικών

Στην επόμενη εικόνα, μπορούμε να δούμε τα χαρακτηριστικά που ενσωματώθηκαν στο γενικό μοντέλο πρόβλεψης:

Πίνακας 2.2: Επιπρόσθετα χαρακτηριστικά γενικού μοντέλου

	Feature	Value Type
1	Latitude	Float
2	Longitude	Float
3	Brand	Alpharithmetic String
4	Neighborhood Cluster	Integer
5	NN-Metric	Float

Latitude and Longitude:

Αυτές είναι οι γεωγραφικές συντεταγμένες που καθορίζουν τη βόρεια-νότια και την ανατολική-δυτική θέση ενός σημείου στην επιφάνεια της Γης αντίστοιχα. Αυτές είναι γωνιακές μετρήσεις, εκφρασμένες σε μοίρες.

Brand Πρατηρίου:

Εξαιτίας του ότι αυτό το χαρακτηριστικό είναι ένα string, έπρεπε να το κωδικοποιήσουμε σε μια ακέραια ετικέτα στο διάστημα 1 – 78. Στο σύνολο των δεδομένων εντοπίσαμε 77 διαφορετικά brands καυσίμων και η τελευταία ετικέτα αφορά όλα τα ανεξάρτητα πρατήρια.

Neighborhood Cluster:

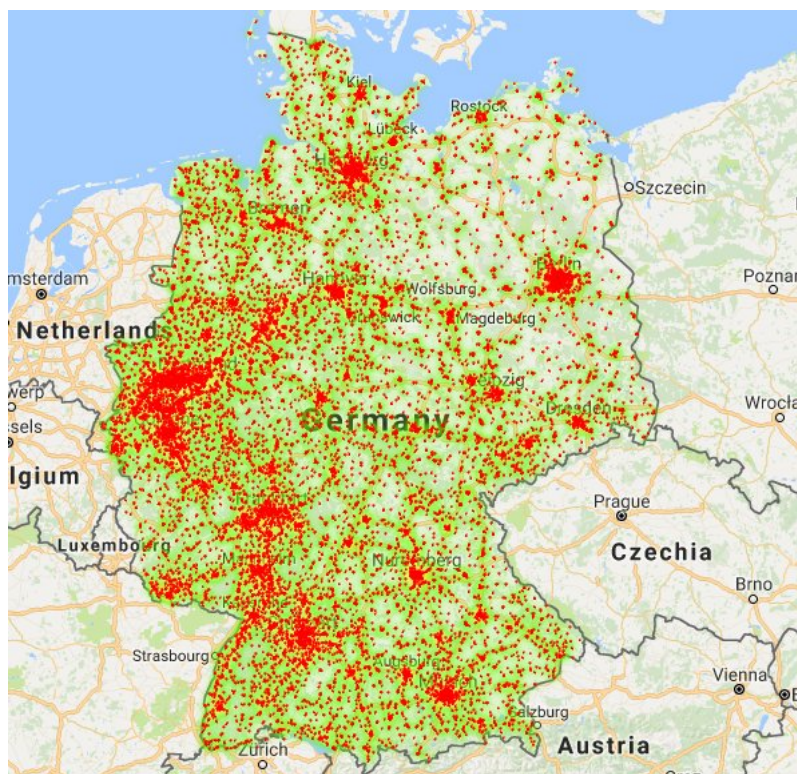
Προκειμένου να ομαδοποιήσουμε τα πρατήρια γεωγραφικά, έπρεπε να εντοπίσουμε τα clusters των πρατηρίων στο χώρο της Γερμανίας, στα οποία είναι κοντινά μεταξύ τους. Για αυτό το λόγο χρησιμοποιήσαμε τον αλγόριθμο **DBSCAN** για συσταδοποίηση. Συγκεκριμένα, υποθέτοντας ως μέγιστη τιμή γειτονίας = 1.5km, βρήκαμε 7278 clusters.

NN-Metric:

Αυτή η μετρική εκφράζει την ανταγωνιστικότητα κάθε πρατηρίου, σύμφωνα με τον αριθμό γειτονικών πρατηρίων. Ύστερα από τη συσταδοποίηση μέσω DBSCAN, υπολογίσαμε και αναθέσαμε σε κάθε πρατήριο τον αριθμό NN-Metric ως εξής:

$nn - metric = \frac{1}{\#(cluster_neighbors)}$ για κάθε πρατήριο, όπου cluster_neighbors είναι ο αριθμός των γειτόνων του.

Παρακάτω, μπορούμε να δούμε το χάρτη της Γερμανίας, όπου απεικονίζονται όλα τα πρατήρια καυσίμων:



Σχήμα 2.7: Χάρτης πρατηρίων στη Γερμανία

Κεφάλαιο 3

Βελτιστοποίηση Δρομολογίων

3.1 Εισαγωγή

Έχοντας αντιμετωπίσει το πρόβλημα της πρόβλεψης τιμών καυσίμου, είμαστε έτοιμοι να προχωρήσουμε στη φάση βελτιστοποίησης. Σε αυτό το κεφάλαιο, θα ορίσουμε το πρόβλημα του ανεφοδιασμού και τον αλγόριθμο ελαχιστοποίησης κόστους. Συγκεκριμένα, ορίζουμε το **Fixed-Route Refueling problem**, όπου μας παρέχεται το δρομολόγιο οχήματος και ο στόχος μας είναι να οργανώσουμε κατάλληλα τις στάσεις του οχήματος, έτσι ώστε να ελαχιστοποιηθεί το κόστος ανεφοδιασμού. Σε αυτή την εργασία, εξετάζουμε την απλή περίπτωση, όπου το δρομολόγιο περιγράφεται ως γράφημα, και οι κόμβοι του είναι τα πρατήρια καυσίμου, από τα οποία το όχημα περνά. Μετά την υλοποίηση του αλγορίθμου, εφαρμόσαμε προεκπαιδευμένα μοντέλα πρόβλεψης, έτσι ώστε να προβλέψουμε τις τιμές καυσίμου από τα εξεταζόμενα πρατήρια για την ζητούμενη χρονική στιγμή άφιξης του οχήματος. Στη συνέχεια, περιγράφουμε τη διαδικασία ενοποίησης της πρόβλεψης και της βελτιστοποίησης.

3.2 Σχετική Εργασία

Οι βελτιστοποιητές καυσίμου είναι μοντέλα απόφασης που είναι αναγνωρισμένα ως αποτελεσματικά εργαλεία διαχείρισης καυσίμου, ειδικά από φορτηγά οχήματα. [30] Αυτοί οι βελτιστοποιητές εφαρμόζουν οργάνωση δρομολογίων για την ελαχιστοποίηση κόστους ανεφοδιασμού και αυτοί οι αλγόριθμοι ανήκουν σε ένα γενικότερο σύνολο αλγορίθμων σε συγκοινωνιακά δίκτυα. Ειδικά, όπως θα δούμε στην επόμενη παράγραφο, το πρόβλημα είναι σχετικό με τα *capacitated lot-sizing* προβλήματα και τα *the vehicle routing* προβλήματα. Η ερευνητική δουλειά έχει πραγματοποιηθεί από τον Shieu-Hong Lin [26],[27] στον τομέα των βέλτιστων πολιτικών ανεφοδιασμού και από τους Samir Khuller, Azarakhsh Malekian and Julian Mastre [29] στον τομέα του *vehicle routing*. Θα εστιάσουμε στις προσεγγίσεις του και θα εφαρμόσουμε έναν γραμμικού χρόνου αλγόριθμο για την περίπτωση του *Simple Fixed Path Vehicle Routing Problem*.

3.3 Θεωρητικό Υπόβαθρο

Το θεωρητικό υπόβαθρο και η περιγραφή του αλγορίθμου βελτιστοποίησης παραλείπεται από την ελληνική έκδοση. Για όλες τις θεωρητικές πληροφορίες ο αναγνώστης μπορεί να ανατρέξει στην αγγλική έκδοση. Θα προχωρήσουμε λοιπόν απευθείας στη μεθοδολογία:

3.4 Μεθοδολογία

Ύστερα από την περιγραφή του αλγορίθμου που βρίσκεται στην αγγλική έκδοση παραπάνω, είμαστε έτοιμη να περιγράψουμε τη μεθοδολογία για τα πειράματά μας και τη δομή του συστήματος εισήγησης.

3.4.1 Ορισμός Δρομολογίου

Για τα πειράματα που διεξήχθησαν, το δρομολόγιο ορίστηκε ως εξής:

- Πληροφορίες Οχήματος
 - Ταχύτητα
 - Χωρητικότητα Δεξαμενής
 - Κατανάλωση Καυσίμου
- Πληροφορίες Πρατηρίου
 - ID Πρατηρίου
 - Αποστάσεις μεταξύ των διαδοχικών πρατηρίων
 - Εκτιμώμενος χρόνος άφιξης

Υποθέσεις:

Για χάριν απλότητας, κάναμε τις εξής υποθέσεις:

Ταχύτητα = $75 \frac{km}{h}$.

Κατανάλωση καυσίμου = $7 \frac{liters}{100km}$

Χωρητικότητα Δεξαμενής = 65 λίτρα.

Όπως βλέπουμε, χρησιμοποιήσαμε σταθερές τιμές ταχύτητας, κατανάλωσης καυσίμου και χωρητικότητας, έτσι ώστε να απλοποιήσουμε την πειραματολογία. Αυτές οι τιμές αντιστοιχούν σε χαρακτηριστικά ενός μέσου I.X.

3.4.2 Benchmarking Δρομολογίων

Θα χρησιμοποιήσουμε για την πειραματολογία ένα σύνολο 20 δρομολογίων στο χώρο της Γερμανίας.

Παρελθοντικά Δρομολόγια: Προκειμένου να συγκρίνουμε τα διαφορετικά μοντέλα και να εκτιμήσουμε την συνεισφορά της πρόβλεψης στην βελτιστοποίηση, κάνουμε την εξής υπόθεση:

Τα δρομολόγια θα πρέπει να έχουν παρελθοντικό χρόνο άφιξης. Με αυτό τον τρόπο, είμαστε σε θέση να έχουμε πρόσβαση στις πραγματικές τιμές που καταγράφηκαν. Έτσι, μπορούμε να εκτιμήσουμε τη βέλτιστη πολιτική ανεφοδιασμού, δηλαδή την πολιτική, που θα μας δώσει ο βελτιστοποιητής ως έξοδο, όταν γνωρίζουμε εξαρχής τις τιμές καυσίμου.

Η μορφολογία των δρομολογίων, όπως αυτά δίνονται στο σύστημα εισήγησης, περιέχει τα ακόλουθα χαρακτηριστικά:

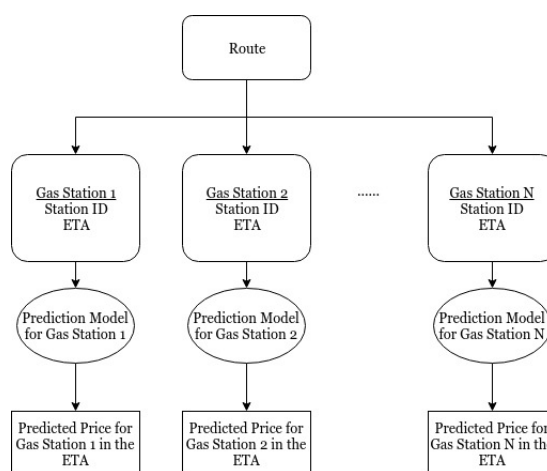
Αυτό σημαίνει πως κάθε εγγραφή πρατηρίου, ως ένας κόμβος του δρομολογίου έχει 4 πεδία, τα Station-ID, Brand, Latitude, Longitude and Estimated Time of Arrival (ETA). Το τελευταίο πεδίο είναι ένα timestamp που δηλώνει την ακριβή ώρα άφιξης του οχήματος στο συγκεκριμένο πρατήριο.

Πίνακας 3.1: Μορφολογία Δρομολογίων

Field	Value Type
Station-ID	String
Brand	String
Latitude	Float
Longitude	Float
ETA	Timestamp

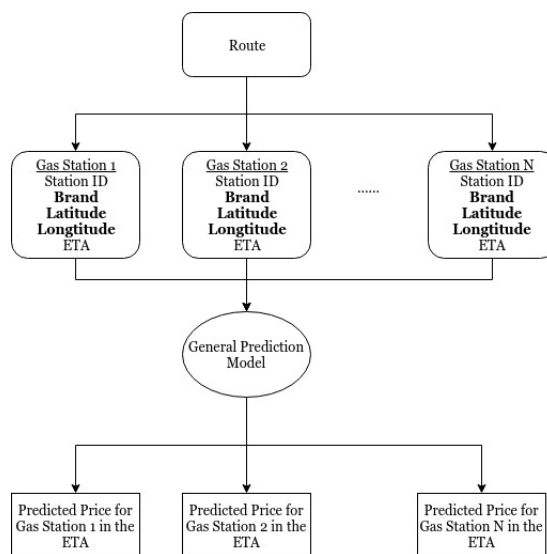
Ενοποίηση της Πρόβλεψης και της Βελτιστοποίησης

Χρησιμοποιώντας τα πεδία κάθε εγγραφής, μπορούμε να τροφοδοτήσουμε τα προεκπαιδευμένα μοντέλα πρόβλεψης, προκειμένου να αποκτήσουμε τις εκτιμώμενες τιμές καυσίμου για τον *Εκτιμώμενο Χρόνο Άφιξης*. Στην επόμενη εικόνα, βλέπουμε τις δύο δομές πρόβλεψης για το **τοπικό** και το **γενικό** μοντέλο:



Σχήμα 3.1: Ορισμός Δρομολογίου για Τοπικό Μοντέλο

Όπως βλέπουμε παραπάνω, χρησιμοποιούμε N προεκπαιδευμένα μοντέλα.



Σχήμα 3.2: Route Definition with General Model Prediction

Εδώ, χρησιμοποιούμε **ένα** μοντέλο για την εκτίμηση των τιμών για όλα τα πρατήρια.

Για τα πειράματά μας, χρησιμοποιήσαμε διαφορετικούς συνδυασμούς δρομολογίων, περιλαμβανομένων διαφορετικών τοποθεσιών στο χώρο της Γερμανίας. Αυτά τα δρομολόγια χαρακτηρίζονται από διαφορετικά χαρακτηριστικά. Συγκεκριμένα, οι διαφορετικοί συνδυασμοί των εξεταζόμενων δρομολογίων περιγράφονται στον επόμενο πίνακα:

Πίνακας 3.2: Πληροφορίες Δρομολογίων

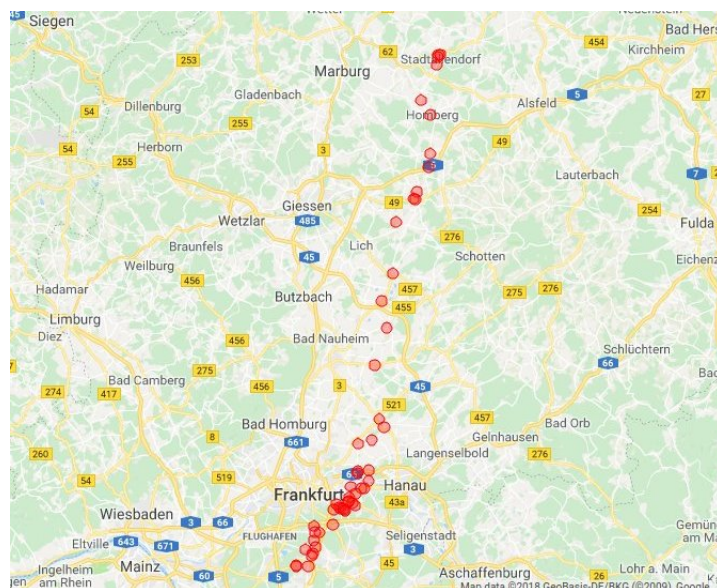
route information	distance (km)	travel time (minutes)	number of gas stations
min	100	80	20
max	1000	750	300

Όπως βλέπουμε από το εύρος των χιλιομέτρων και των χρόνων ταξιδιού, τα δρομολόγια μπορούν να περιγράψουν ταξίδια ιδιωτικών ή επαγγελματικών φορτηγών οχημάτων.

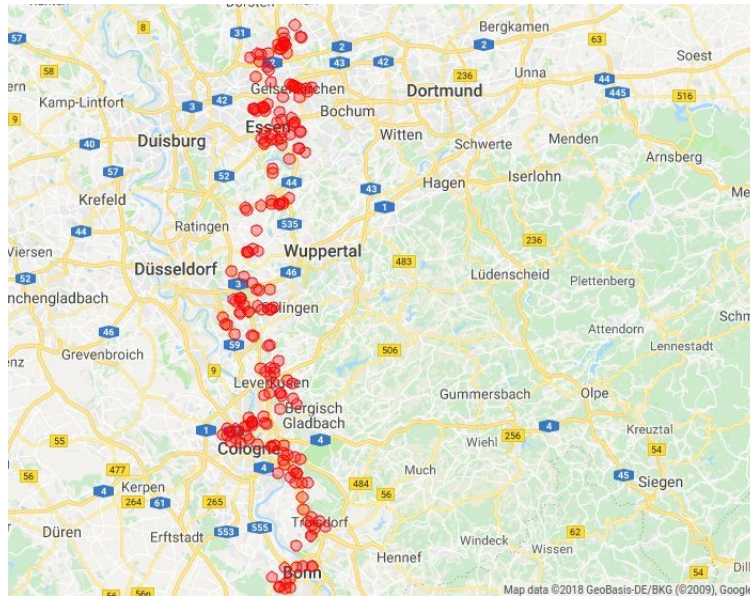
Συγκεκριμένα, το σύνολο των εξεταζόμενων δρομολογίων είναι ως εξής:

- **Μικρά Δρομολόγια:** 5 διαφορετικά δρομολόγια με συνολική απόσταση < 200km
- **Μεσαία Δρομολόγια:** 5 διαφορετικά δρομολόγια με 200km < συνολική απόσταση < 400km
- **Μεγάλα Δρομολόγια:** 5 διαφορετικά δρομολόγια με 400km < συνολική απόσταση < 800km
- **Πολύ μεγάλα Δρομολόγια:** 5 διαφορετικά δρομολόγια με 800km < συνολική απόσταση < 1000km

Στη συνέχεια παρουσιάζουμε 2 παραδείγματα των εξεταζόμενων δρομολογίων στο χώρο της Γερμανίας:



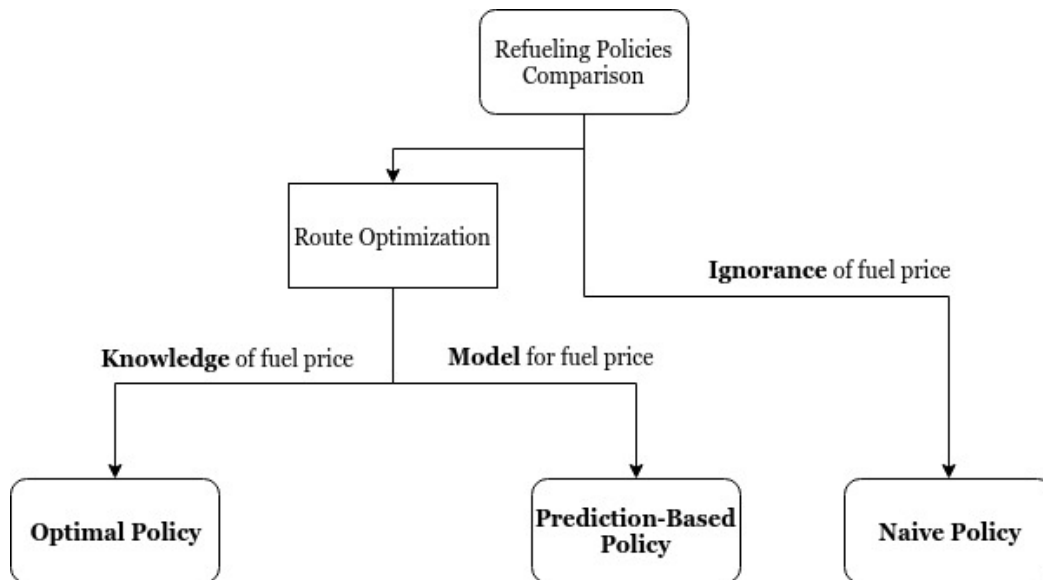
Σχήμα 3.3: Examined Route with 50 gas stations



Σχήμα 3.4: Examined Route with 150 gas stations

3.5 Εκτίμηση της βελτιστοποίησης βασιζόμενης στην πρόβλεψη

Για την πειραματολογία μας, θα χρησιμοποιήσουμε ένα σύνολο δρομολογίων, όπως αυτό ορίστηκε προηγουμένως. Προκειμένου να εκτιμήσουμε την συνεισφορά της πρόβλεψης στην βελτιστοποίηση, πρέπει να υπολογίσουμε το συνολικό κόστος ανεφοδιασμού για 3 περιπτώσεις, όπως φαίνονται παρακάτω:



Σχήμα 3.5: Σύγκριση Πολιτικών Ανεφοδιασμού

Αυτές οι 3 περιπτώσεις αναλύονται στην παρακάτω παράγραφο:

3.5.1 Πολιτικές

Ορίζουμε τις ακόλουθες μεταβλητές:

- **Πολιτική Ανεφοδιασμού:** $F = \{f_0, \dots, f_{n-1}\}$
- **Πραγματικές Τιμές Καυσίμου:** c_i
- **Συνολικό Κόστος Ανεφοδιασμού:** $R = \sum_0^{n-1} f_i * c_i$

, όπου i δηλώνει το *ιστό* πρατήριο.

Χρησιμοποιώντας αυτές τις μεταβλητές, ορίζουμε τις 3 πολιτικές:

1. **Βέλτιστη Πολιτική:** Αυτό είναι το εξαγόμενο διάνυσμα που παρέχει ο βελτιστοποιητής, αν γνωρίζουμε *a priori* τις τιμές καυσίμου του κάθε πρατηρίου.
2. **Βασιζόμενη στην πρόβλεψη Πολιτική:** Αυτό είναι το εξαγόμενο διάνυσμα που παρέχει ο βελτιστοποιητής, ύστερα από την πρόβλεψη των τιμών καυσίμου.
3. **Απλή Πολιτική:** Αυτό είναι το διάνυσμα που παρέχει η κανονική συμπεριφορά οδηγού, μη γνωρίζοντας τις τιμές καυσίμου.

Για αυτές τις πολιτικές, τα αντίστοιχα κόστη ανεφοδιασμού είναι τα ακόλουθα:

1. **Κόστος Βέλτιστης Πολιτικής:** $R_{optimal} = \sum_0^{n-1} f_i^{optimal} * c_i$,
2. **Κόστος Βασιζόμενης στην πρόβλεψη Πολιτικής:** $R_{predicted} = \sum_0^{n-1} f_i^{predicted} * c_i$
3. **Κόστος Απλής Πολιτικής:** $R_{naive} = \sum_0^{n-1} f_i^{naive} * c_i$

Όπως μπορούμε να δούμε παραπάνω, χρησιμοποιούμε τις **πραγματικές** τιμές καυσίμου για την σύγκριση των 3 πολιτικών.

3.6 Λεπτομέρειες Υλοποίησης

:

Για τη φάση της βελτιστοποίησης χρησιμοποιήσαμε τα ακόλουθα:

- **Εξαγωγή δεδομένων πρατηρίων:** PostgreSQL και psycopg2.
- **Αλγόριθμος Βελτιστοποίησης:** Python 3.4
- **Σύστημα Εισήγησης:** Python 3.4

Κεφάλαιο 4

Αποτελέσματα

4.1 Εισαγωγή

Σε αυτό το κεφάλαιο παρουσιάζουμε τα αποτελέσματα και από τις δύο φάσεις του έργου, της πρόβλεψης και της βελτιστοποίησης. Πρώτα, θα εστιάσουμε στην ακρίβεια του πρόβλεπτη μας στο μοντέλο εξατομικευμένου πρατηρίου και στο γενικό μοντέλο. Συγκεκριμένα, θα παρουσιαστούν και θα σχολιαστούν οι αποδόσεις των εκτιμητών για τις δύο προσεγγίσεις χρησιμοποιούμενων χαρακτηριστικών και για τους δύο τύπους μοντέλων. Στη συνέχεια, παρουσιάζουμε την απόδοση της συνεργασίας πρόβλεψης και βελτιστοποίησης σε ένα σύνολο δρομολογίων, όπου εκτιμάται το τελικό κόστος ανεφοδιασμού.

4.2 Πρόβλεψη - Τοπικό Μοντέλο

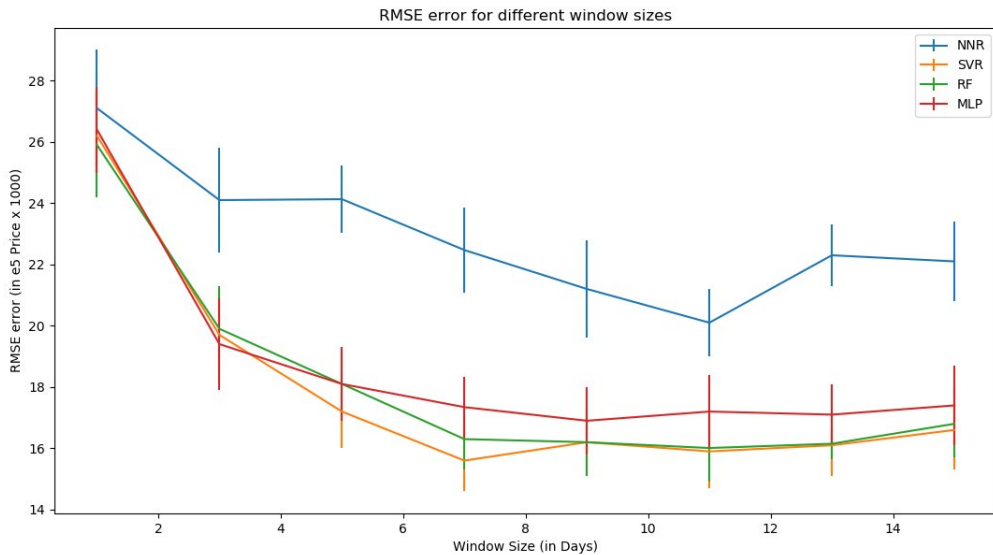
Πρώτα, θα δείξουμε την ακρίβεια πρόβλεψης για τους διαφορετικούς εκτιμητές στην περίπτωση του εξατομικευμένου μοντέλου για κάθε πρατήριο (Τοπικό Μοντέλο). Για το σύνολο των πειραμάτων, χρησιμοποιήσαμε τα εξής:

- Σύνολο Εκπαίδευσης/Εξέτασης = 1000 τυχαία επιλεγμένα πρατήρια στο χώρο της Γερμανίας (περίπου το 8% του συνολικού αριθμού πρατηρίων στην Γερμανία)
- Validation Σχήμα = Το σχήμα που παρουσιάστηκε στο μέρος 2.4.3.
- Μεταβλητή Στόχος = E5 Price * 1000. Δεν διατηρούμε την αρχική μέτρηση E5, για να κάνουμε το σφάλμα RMSE ορατότερο.
- Μέθοδος κλιμακοποίησης = Κανονικοποίηση, όπως περιγράφηκε στο 2.4.3.

4.2.1 Διάνυσμα Ενσωμάτωσης

Στη διάρκεια των πειραμάτων χρησιμοποιώντας την προσέγγιση TSA, έπρεπε να ορίσουμε το μέγεθος του διανύσματος ενσωμάτωσης και συνεπώς το μέγεθος του διανύσματος χαρακτηριστικών. Θυμίζουμε ότι το διάνυσμα ενσωμάτωσης είναι το διάνυσμα που περιέχει τις καθυστερημένες τιμές της χρονοσειράς. Αυτό σημαίνει ότι χρησιμοποιούμε ως χαρακτηριστικά τις προηγούμενες παρατηρημένες τιμές καυσίμου.

Στην επόμενη εικόνα, παρουσιάζουμε τα σφάλματα πρόβλεψης για τους 4 εκτιμητές (με σταθερές διαμορφώσεις) για διαφορετικά μεγέθη διανυσμάτων ενσωμάτωσης.

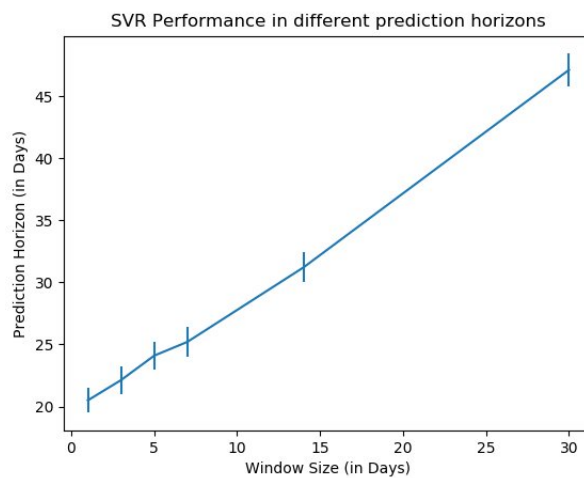


Σχήμα 4.1: Επίδραση του διανύσματος ενσωμάτωσης

Όπως μπορούμε να δούμε από την εικόνα παραπάνω, έχουμε μια μείωση του σφάλματος RMSE μέχρι ένα μέγεθος παραθύρου 7 ημερών και από εκεί και πέρα αυτή η μείωση γίνεται μικρότερη. Έχοντας υπόψη ότι μεγαλώνοντας το μέγεθος παραθύρου έχουμε ένα μεγαλύτερο διάνυσμα χαρακτηριστικών και συνεπώς αργότερη πρόβλεψη, μπορούμε να επιλέξουμε ως βέλτιστο μέγεθος παραθύρου $p = 7$ ημέρες (που αντιστοιχούν σε μία εβδομάδα). Αυτό σημαίνει ότι διάνυσμα χαρακτηριστικών (για την TSA προσέγγιση) για κάθε εγγραφή πρέπει να είναι $24 * 7 = 168$ χρονικά δεδομένα. Έτσι, για τις επόμενες μετρήσεις και πειράματα της TSA προσέγγισης θα χρησιμοποιήσουμε $24 * 7 = 168$ παρατηρημένες τιμές ως χαρακτηριστικά.

4.2.2 Ορίζοντας Πρόβλεψης

Σε αυτό το σημείο θα ασχοληθούμε με το πώς η **αναδρομική** στρατηγική επηρεάζει την πρόβλεψη πολλαπλού βήματος, όταν ακολουθούμε την ανάλυση TSA. Συγκεκριμένα, θα χρησιμοποιήσουμε το SVM ως εκτιμητή, για να παρουσιάσουμε το σφάλμα RMSE, όπως αυτό προέκυψε για διαφορετικούς ορίζοντες πρόβλεψης :



Σχήμα 4.2: Σφάλμα RMSE για διαφορετικούς ορίζοντες πρόβλεψης

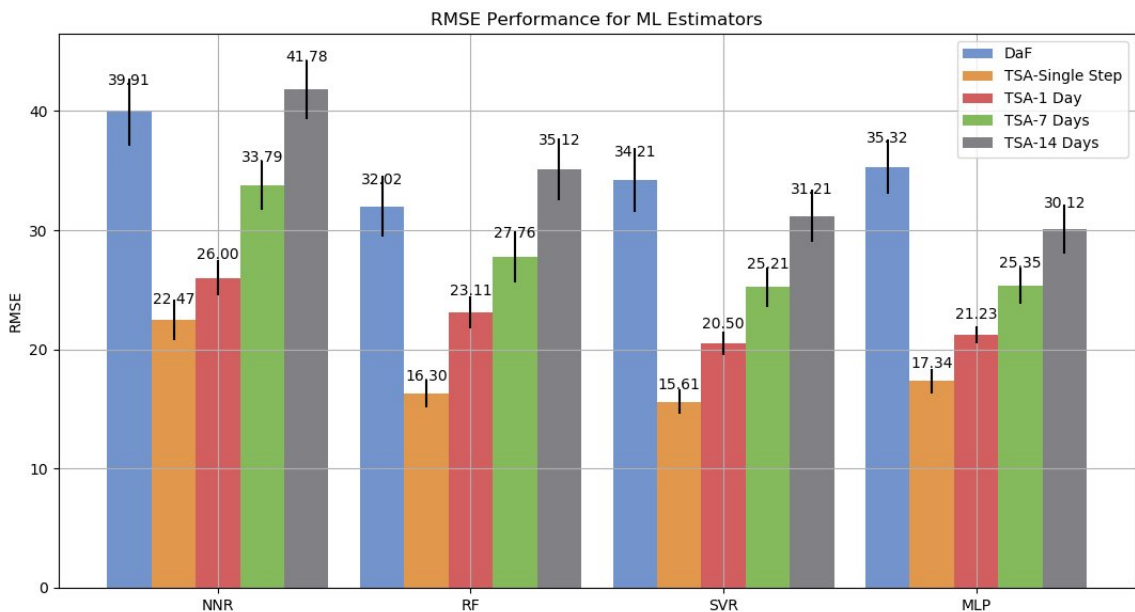
Όπως μπορούμε να δούμε, η αναδρομική στρατηγική ενσωματώνει ένα συσσωρευμένο

σφάλμα στις προβλέψεις. Συγκεκριμένα, ενώ στην πρόβλεψη 1 ημέρας μπορούμε να πετύχουμε ένα μέσο $RMSE = 20.8$ μονάδων, στις 30 ημέρες πρόβλεψης αυτό το σφάλμα εκτοξεύεται στα $RMSE = 45.23$. Αυτό είναι μια ένδειξη ότι δεν είναι αποδοτική η χρήση TSA και πρόβλεψη πολλαπλού βήματος, όταν χρειαζόμαστε μακροπρόθεσμη πρόβλεψη. Αυτό το γεγονός μπορεί να επικυρωθεί κι από την επόμενη παράγραφο, όπου συγκρίνουμε την απόδοση των δύο προσεγγίσεων DaF και TSA για το τοπικό μοντέλο.

4.2.3 Σύγκριση Απόδοσης των Εκτιμητών Μηχανικής Μάθησης

Επιλέγοντας τις βέλτιστες παραμέτρους, που βρήκαμε, μπορούμε τώρα να συγκρίνουμε την απόδοση των διαφορετικών εκτιμητών για τις 2 προσεγγίσεις.

Η εικόνα παρακάτω δείχνει την απόδοση των Random Forest, Support Vector Regression, K-Nearest Neighbors and Neural Network:



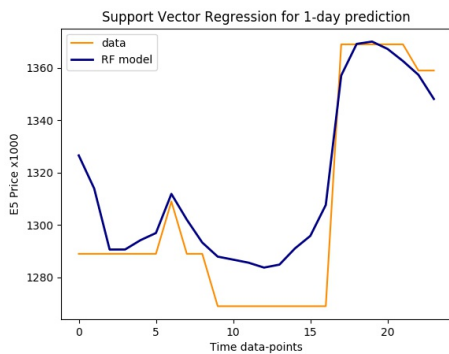
Σχήμα 4.3: RMSE performance for best configurations of NNR, RF, SVR, MLP

Σχόλια:

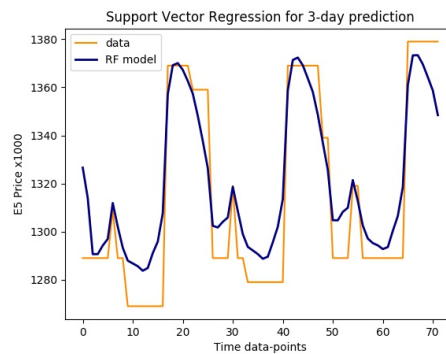
- Στην DaF προσέγγιση, ο **Random Forest** έχει την καλύτερη μέση ακρίβεια ($RMSE = 32.02$ μονάδες), ενώ η χειρότερη απόδοση παρουσιάστηκε από τον K-Nearest Neighbors.
- Στην TSA προσέγγιση, ο **Support Vector Regression** υπερτερεί των υπολοίπων εκτιμητών. Συγκεκριμένα, ο SVR πετυχαίνει το χαμηλότερο σφάλμα $RMSE = 15.61$ για πρόβλεψη μονού βήματος. Δεύτερη καλύτερη απόδοση πετυχαίνει ο Random Forest με $RMSE = 16.30$. Ξανά, η χειρότερη επίδοση εμφανίζεται από τον NNR με $RMSE = 22.47$
- Είναι άξιο προσοχής, πως χρησιμοποιώντας ως ορίζοντα πρόβλεψης τις 14 μέρες(με TSA προσέγγιση), το σφάλμα RMSE φτάνει (και σε μερικές περιπτώσεις ξεπερνάει) το σφάλμα της DaF προσέγγισης. Αυτό σημαίνει πως ο εκτιμητής με DaF προσέγγιση είναι καταλληλότερος, όταν θέλουμε να κάνουμε μακροπρόθεσμη πρόβλεψη.

4.2.4 Οπτικοποίηση της απόδοσης του SVM

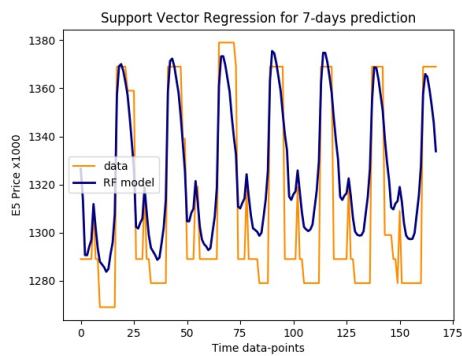
Ύστερα από την πειραματολογία των διαφορετικών εκτιμητών και διαμορφώσεων, μπορούμε να οπτικοποιήσουμε την απόδοση του προβλέπτη για πολλαπλά βήματα πρόβλεψης. Για το λόγο αυτό θα χρησιμοποιήσουμε το πρατήριο με ID = 005056ba-7cb6-1ed2-bceb-60512dc8cd1b. Αυτό το σύνολο έχει 26774 εγγραφές, από Ιούλιο 2014 μέχρι Ιούνιο 2017.



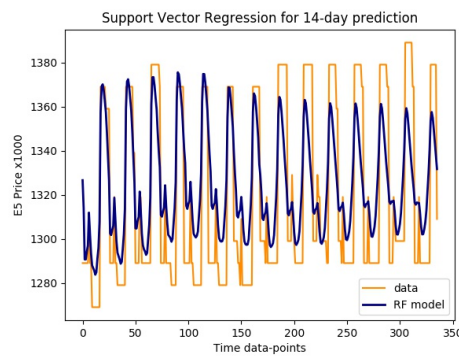
(a) 1-Day Prediction



(b) 3-Day Prediction



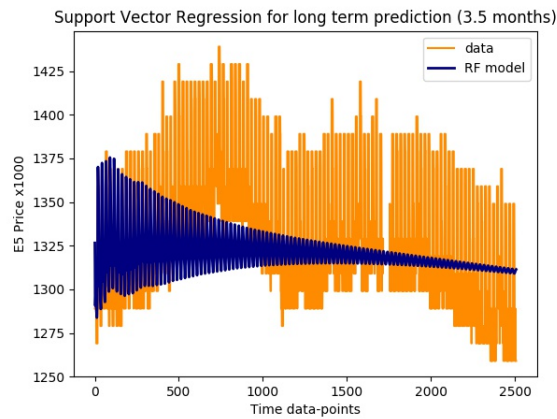
(c) 7-Day Prediction



(d) 14-Day Prediction

Σχήμα 4.4: Prediction of SVR regression for 1, 3, 7, 14 days

Παρακάτω, παρουσιάζουμε και τα αποτελέσματα για την μακροπρόθεσμη πρόβλεψη. Μπορούμε να δούμε πως η αναδρομική στρατηγική για την πρόβλεψη πολλαπλού βήματος και η φύση του SVM δεν ανταποκρίνονται καλά στην μακροπρόθεσμη πρόβλεψη. Συγκεκριμένα, βλέπουμε σύγκλιση του εκτιμητή.



Σχήμα 4.5: SVR Prediction for 3.5 months

Από την παραπάνω εικόνα, φαίνεται ξεκάθαρα η ανικανότητα της TSA προσέγγισης σε συνδυασμό με το SVM για την παροχή μακροπρόθεσμης πρόβλεψης.

4.3 Πρόβλεψη - Γενικό Μοντέλο

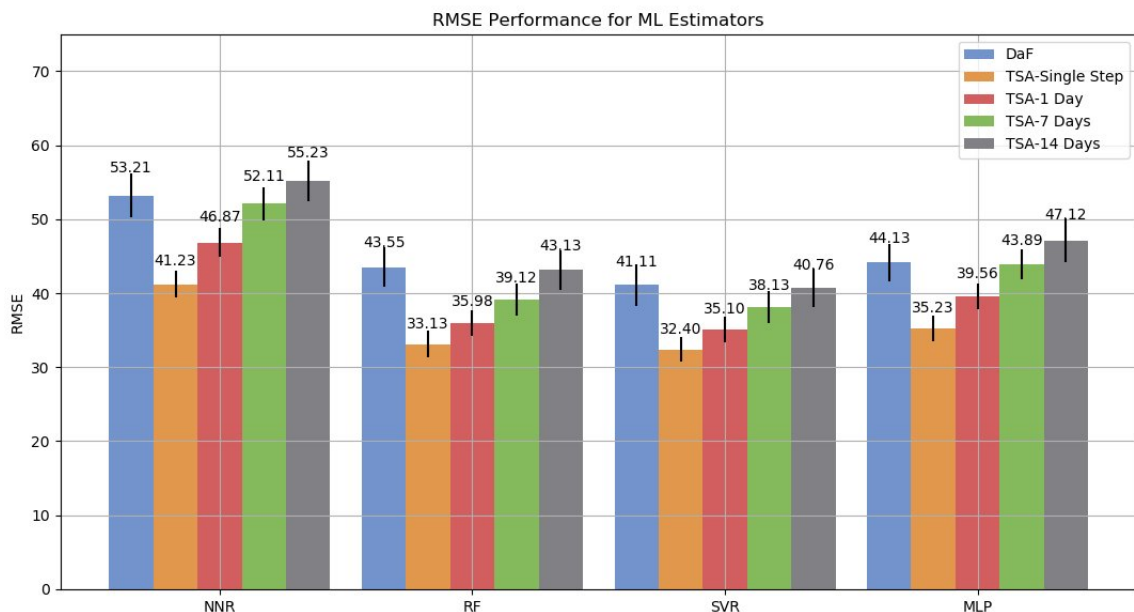
Τώρα, είμαστε έτοιμοι να παρουσιάσουμε τα αποτελέσματα για το γενικό μοντέλο. Σε αυτή την περίπτωση, χρησιμοποιήσαμε την παρακάτω διαμόρφωση:

- Σύνολο Εκπαίδευσης = 2500 τυχαία επιλεγμένα πρατήρια στο χώρο της Γερμανίας (περίπου το 19% του συνολικού αριθμού πρατηρίων στην Γερμανία)
- Σύνολο Εξέτασης = 250 τυχαία επιλεγμένα πρατήρια στο χώρο της Γερμανίας (περίπου το 2% του συνολικού αριθμού πρατηρίων στην Γερμανία)
- Validation Σχήμα = Το σχήμα που παρουσιάστηκε στο μέρος 2.4.3.
- Μεταβλητή Στόχος = E5 Price * 1000. Δεν διατηρούμε την αρχική μέτρηση E5, για να κάνουμε το σφάλμα RMSE ορατότερο.
- Μέθοδος κλιμακοποίησης = Κανονικοποίηση, όπως περιγράφηκε στο 2.4.3.

Όπως και στο τοπικό μοντέλο, πρώτα εφαρμόσαμε grid-search παραμετροποίηση για τους 4 τύπους εκτιμητών προκειμένου να βρούμε τις βέλτιστες παραμέτρους. Αυτές οι παραμετροποιήσεις παραλείπονται από την ελληνική έκδοση.

4.3.1 Σύγκριση Απόδοσης των Εκτιμητών Μηχανικής Μάθησης

Ακολουθώντας την ίδια πειραματολογία, όπως στο τοπικό μοντέλο, παρουσιάζουμε παρακάτω τα αποτελέσματα σύγκρισης. Επισημαίνουμε ξανά ότι επιλέχθηκαν οι βέλτιστες παράμετροι:



Σχήμα 4.6: RMSE performance for best configurations of NNR, RF, SVR, MLP

Σχόλια:

- Ένα πρώτο σχόλιο που μπορούμε να κάνουμε είναι η χαμηλότερη ακρίβεια που παρουσιάζουν οι εκτιμητές στο γενικό μοντέλο σε σύγκριση με το τοπικό. Μπορούμε να υποστηρίξουμε ότι η πρόβλεψη τοπικού μοντέλου υπερτερεί του γενικού σε επίπεδο ακρίβειας.
- Σε αντίθεση με το τοπικό μοντέλο, παρατηρούμε πως το SVM υπερτερεί των υπολοίπων στην DaF προσέγγιση με $RMSE = 41.11$. Ο Random Forest εκτιμητής ακολουθεί με $RMSE = 43.55$ και το μεγαλύτερο σφάλμα προέκυψε και πάλι από τον k-NN ($RMSE = 53.21$).
- Όπως και στο τοπικό μοντέλο, όταν χρησιμοποιούμε TSA προσέγγιση, ο SVM υπερτερεί ξανά των υπολοίπων με $RMSE = 32.40$.

4.4 Πρόβλεψη και Βελτιστοποίηση

Σε αυτό το σημείο, έχοντας βρει τις βέλτιστες παραμέτρους για τους εξεταζόμενους εκτιμητές και έχοντας υπολογίσει την προβλεπτική τους ικανότητα, είμαστε έτοιμοι να εξετάσουμε την απόδοση της συνεργασίας του προβλέπτη με τον εκτιμητή.

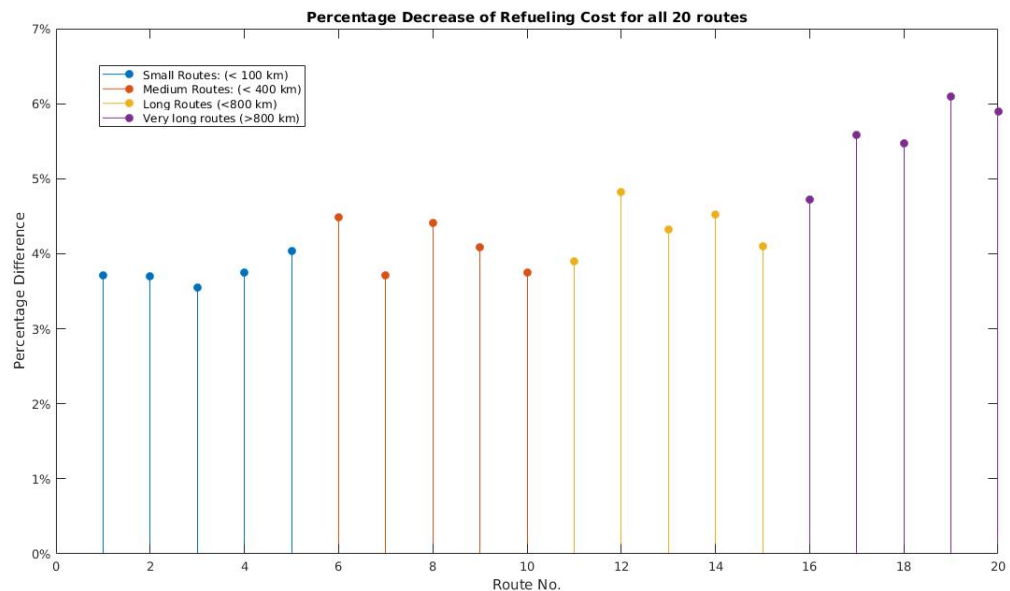
4.4.1 Βέλτιστη Πολιτική

Στο επόμενο διάγραμμα, παρουσιάζουμε την ελαχιστοποίηση κόστους ανεφοδιασμού, χρησιμοποιώντας τον αλγόριθμο βελτιστοποίησης, θεωρώντας ότι γνωρίζουμε εξ αρχής τις τιμές καυσίμου.

Επισημαίνουμε πως για αυτή την εκτίμηση, υπολογίζουμε την ποσοστιαία μείωση του κόστους ανεφοδιασμού.

Αυτό περιγράφεται ως εξής:

$$\text{Percentage Decrease} = \frac{R_{\text{examined}} - R_{\text{naive}}}{R_{\text{naive}}} * 100\% .$$



Σχήμα 4.7: Percentage Decrease for all 20 routes

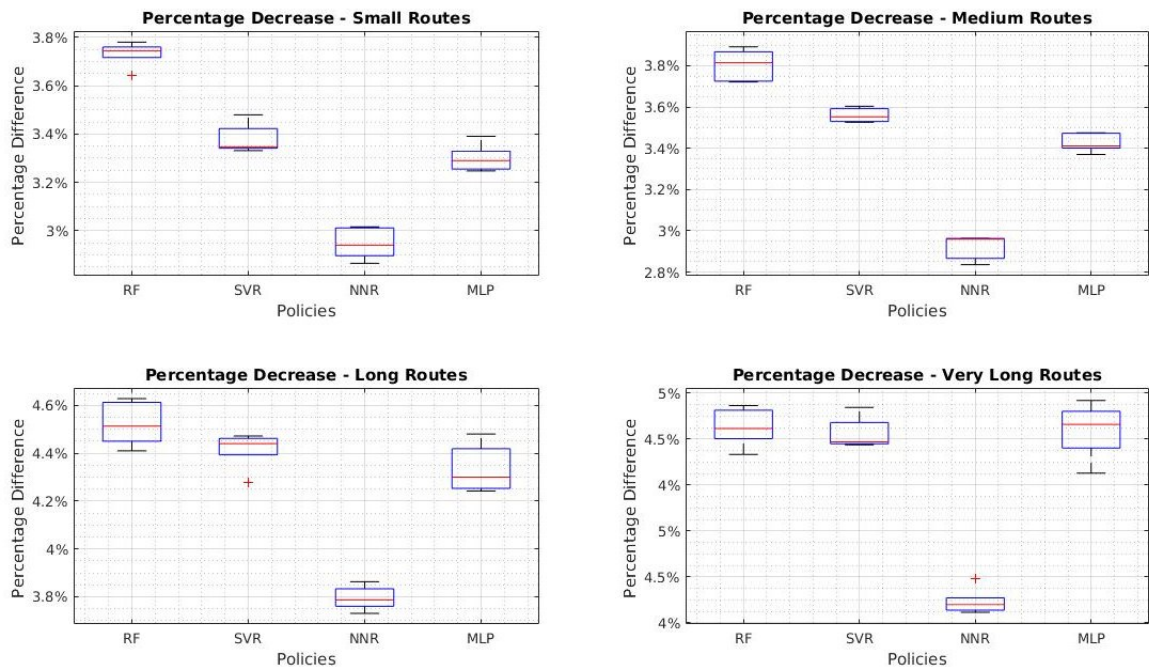
Όπως μπορούμε να δούμε από την εικόνα παραπάνω, όσο μεγαλύτερα δρομολόγια έχουμε τόσο μεγαλύτερη είναι η μείωση που πετυχαίνει ο βελτιστοποιητής.

4.4.2 Τοπικό Μοντέλο

Εδώ έχουμε την περίπτωση του τοπικού μοντέλου πρόβλεψης, χρησιμοποιώντας τις δύο προσεγγίσεις DaF και TSA. Για κάθε εξεταζόμενο δρομολόγιο, εκπαιδεύσαμε ένα μοντέλο πρόβλεψης για κάθε πρατήριο.

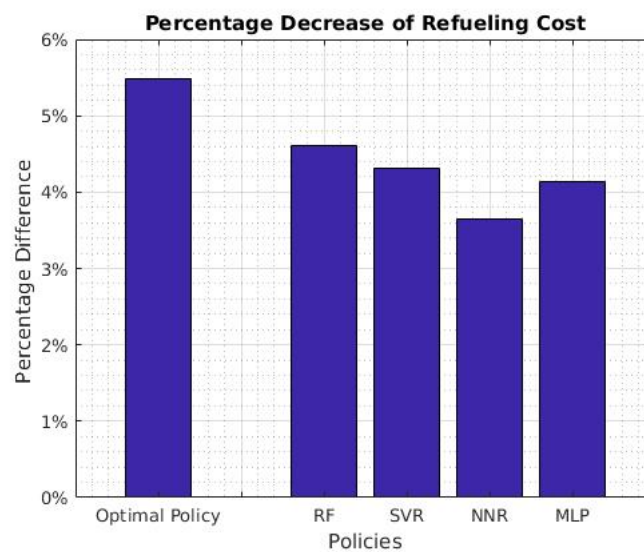
DaF Προσέγγιση

Το πρώτο διάγραμμα δείχνει την ποσοστιαία μείωση για τους διαφορετικούς εκτιμητές, σύμφωνα με τα διαφορετικά μήκη δρομολογίων.



Σχήμα 4.8: Percentage Decrease for DaF Approach per route type

The second diagram show the average percentage decrease for the different estimators:

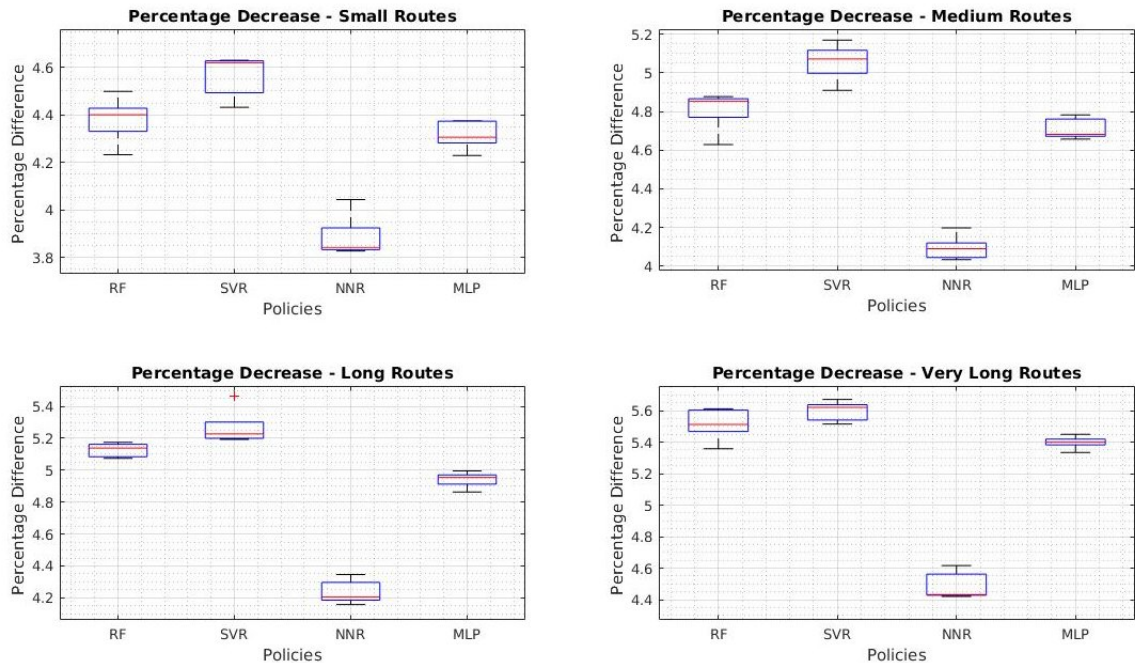


Σχήμα 4.9: Average Percentage Decrease for DaF Approach

Όπως βλέπουμε ο Random Forest εκτιμήτης υπερτερεί των υπολοίπων στη DaF προσέγγιση με μια μέση ποσοστιαία μείωση 4.60% και αυτό σημαίνει πως έχουμε μια διαφορά 0.88% από τη βέλτιστη πολιτική. Αυτό επίσης επικυρώνεται από την στατιστική διαφορά για τους διαφορετικούς τύπους δρομολογίων, όπως φαίνονται παραπάνω.

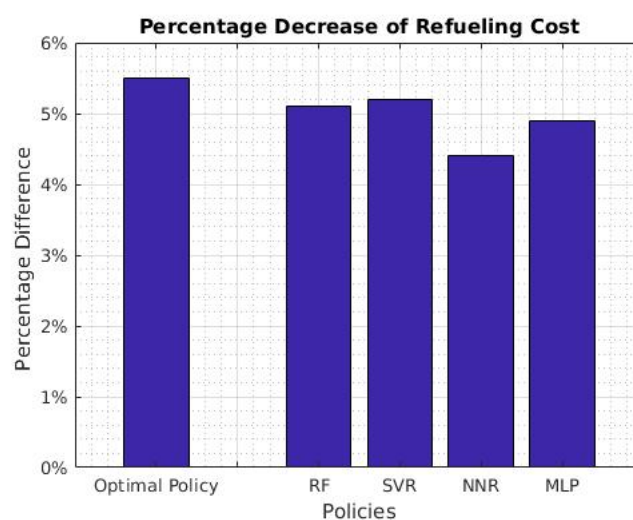
TSA Προσέγγιση

Εδώ έχουμε τα αποτελέσματα για την ίδια πειραματολογία χρησιμοποιώντας την TSA προσέγγιση:



Σχήμα 4.10: Percentage Decrease for TSA Approach

Το δεύτερο διάγραμμα δείχνει την μέση ποσοστιαία μείωση για τους διαφορετικούς εκτιμητές:



Σχήμα 4.11: Average Percentage Decrease for TSA Approach

Στην TSA προσέγγιση, βλέπουμε ότι έχουμε την μεγαλύτερη μέση ποσοστιαία μείωση

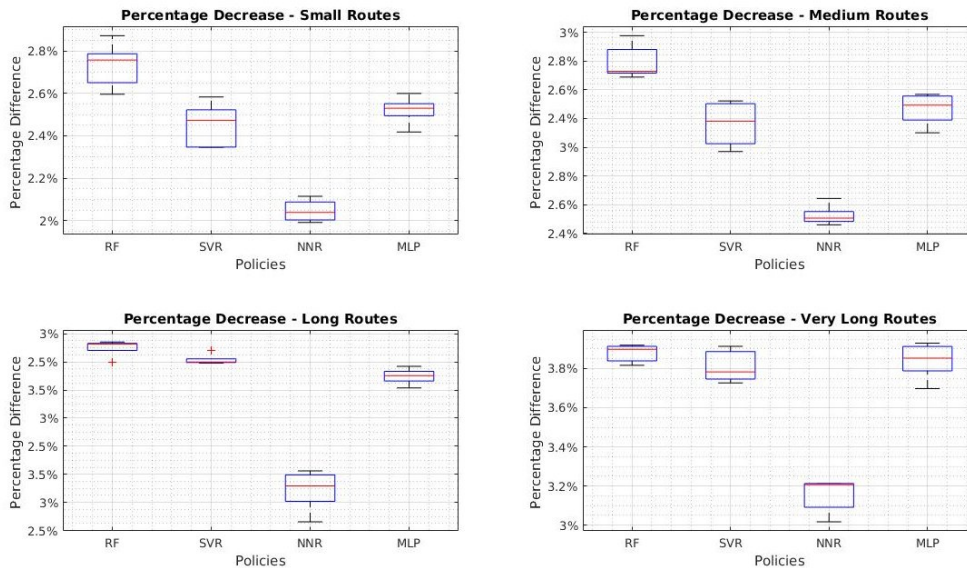
χρησιμοποιώντας το SVM (με διαφορά $5.48\% - 5.20\% = 0.28\%$ από τη βέλτιστη πολιτική).

4.4.3 Γενικό Μοντέλο

Σε αυτή την περίπτωση, έχουμε τα αποτελέσματα για την πρόβλεψη γενικού μοντέλου. Συγκεκριμένα, χρησιμοποιήσαμε το **ίδιο** μοντέλο για κάθε πρατήριο σε κάθε δρομολόγιο.

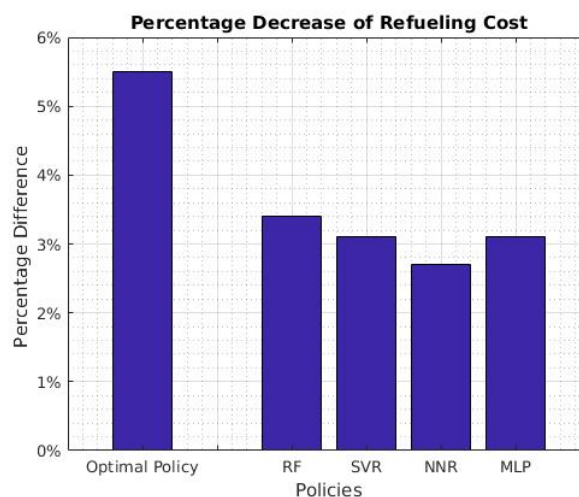
DaF Προσέγγιση

Και πάλι ακολουθώντας την ίδια διαδικασία, παρουσιάζουμε την DaF προσέγγιση:



Σχήμα 4.12: Percentage Decrease for DaF Approach

Το δεύτερο διάγραμμα δείχνει την μέση ποσοστιαία μείωση για τους διαφορετικούς εκτιμητές:



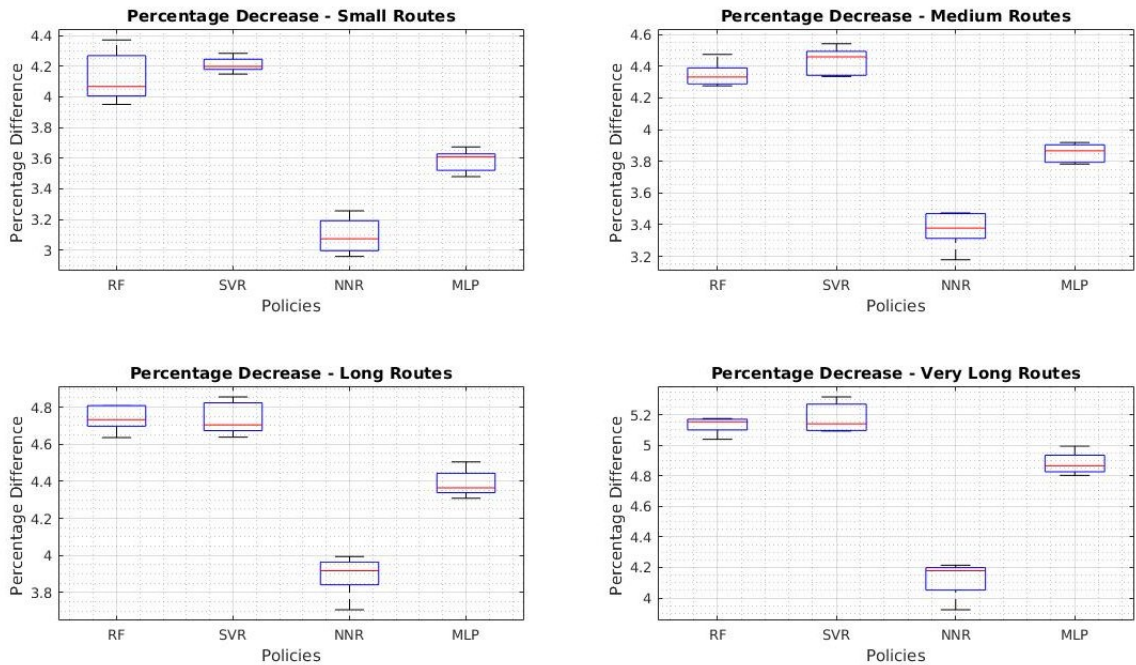
Σχήμα 4.13: Average Percentage Decrease for DaF Approach

Όπως και στο τοπικό μοντέλο, βλέπουμε ξανά ότι ο Random Forest υπερτερεί των υπολοίπων στην DaF προσέγγιση. Μπορούμε να πετύχουμε 3.42% μέση ποσοστιαία μείωση

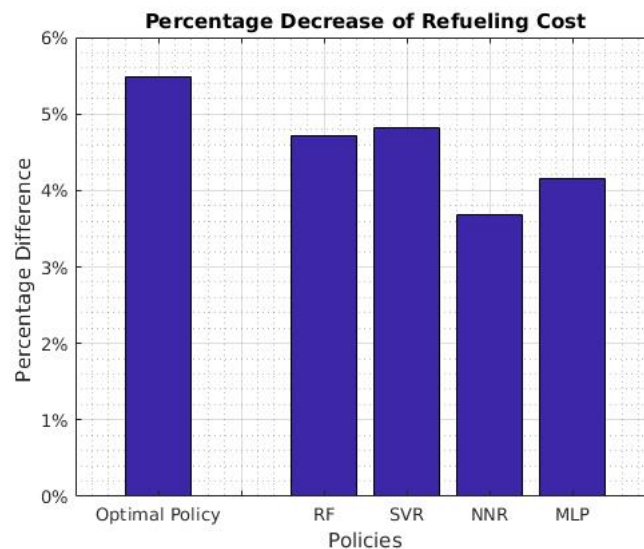
και αυτό σημαίνει πως έχουμε διαφορά 2.06% από τη βέλτιστη πολιτική.
 Η δεύτερη καλύτερη απόδοση παρουσιάζεται στο SVR και στο MLP.

TSA Προσέγγιση

Και πάλι, δίνουμε τα αποτελέσματα για την TSA προσέγγιση στο γενικό μοντέλο:



Σχήμα 4.14: Percentage Decrease for TSA Approach



Σχήμα 4.15: Average Percentage Decrease for TSA Approach

Στην TSA προσέγγιση, μπορούμε να δούμε πως το SVM υπερτερεί των υπολοίπων με μια μέση ποσοστιαία μείωση 4.82% (και διαφορά 0.66% από τη βέλτιστη πολιτική).

Κεφάλαιο 5

Συμπεράσματα και Επεκτάσεις

5.1 Βασικά Συμπεράσματα και Συζήτηση

Στην παρούσα εργασία, προτείνουμε ένα σύστημα εισήγησης, το οποίο στοχεύει στην ελαχιστοποίηση του κόστους ανεφοδιασμού για δεδομένο δρομολόγιο οχήματος. Αυτό το σύστημα αξιοποιεί προεκπαιδευμένα μοντέλα πρόβλεψης, με σκοπό την απόκτηση γνώσης για τις τιμές καυσίμου σε κάθε πρατήριο. Συγκεκριμένα, εστιάζουμε σε δρομολόγια και πρατήρια στο οδικό σύστημα της Γερμανίας, όπου οι τιμές καυσίμου διακυμαίνονται έντονα κατά τη διάρκεια της ημέρας. Για την πρόβλεψη, διεξήχθησαν πειράματα πάνω σε διαφορετικούς τύπους εκτιμητών μηχανικής μάθησης και διαφορετικές προσεγγίσεις. Αναπτύξαμε δύο τύπους προσεγγίσεων. Κατά τον πρώτο τύπο, ενδιαφερόμαστε για τον αριθμό των πρατηρίων, τα οποία περιγράφει το μοντέλο πρόβλεψης. Συγκεκριμένα, από τη μία μεριά αναπτύσσουμε **τοπικά** μοντέλα, εξατομικευμένα για κάθε πρατήριο και από την άλλη αναπτύσσουμε **γενικά** μοντέλα, τα οποία αξιοποιούν γεωχρονικά δεδομένα, προκειμένου να περιγράψουν πρότυπα από όλο το σύνολο των πρατηρίων της Γερμανίας. Κατά τον δεύτερο τύπο προσεγγίσεων, μας ενδιαφέρει το διάνυσμα χαρακτηριστικών που χρησιμοποιούμε. Σε αυτή την περίπτωση, ακολουθούμε από τη μία μεριά την **DaF** (Date as Feature) προσέγγιση, όπου χρησιμοποιούμε στατικά χρονικά δεδομένα ως χαρακτηριστικά του μοντέλου, ενώ από την άλλη μεριά ακολουθούμε την **TSA** (Time Series Analysis) προσέγγιση, όπου χρησιμοποιούμε τις προηγούμενες παρατηρημένες τιμές καυσίμων. Από τη στιγμή που έχουμε εκπαιδεύσει τα μοντέλα πρόβλεψης για τις τιμές καυσίμου, τα χρησιμοποιούμε στον βελτιστοποιητή, προκειμένου να πάρουμε εκτιμήσεις τιμών για συγκεκριμένο πρατήριο και για συγκεκριμένη ώρα και μέρα, σύμφωνα με το δεδομένο δρομολόγιο. Βασισμένοι στις εκτιμήσεις, προτείνουμε μια βέλτιστη πολιτική ανεφοδιασμού, προκειμένου να ελαχιστοποιήσουμε το κόστος ανεφοδιασμού.

5.1.1 Ακρίβεια Πρόβλεψης

Παρατηρώντας το σύνολο της πειραματολογίας στο προηγούμενο κεφάλαιο, μπορούμε να καούμε το γενικό σχόλιο ότι οι νικητές εκτιμητές είναι οι **SVM** και **Random Forest**. Συγκεκριμένα, στο τοπικό μοντέλο, ο **SVR** υπερτερεί των υπολοίπων στην TSA προσέγγιση με σφάλμα μονού βήματος $RMSE = 15.61$ μονάδων, ενώ ο **RF** υπερτερεί των υπολοίπων στην DaF προσέγγιση με $RMSE = 32.01$.

Από την άλλη μεριά, στο γενικό μοντέλο, συμπεραίνουμε ότι και στις δύο προσεγγίσεις ο **SVR** υπερτερεί των υπολοίπων με σφάλμα $RMSE = 41.11$ και $RMSE = 32.40$ αντίστοιχα. Επίσης, έγινε αντιληπτό ότι το τοπικό μοντέλο πέτυχε καλύτερη ακρίβεια από το γενικό μοντέλο. Παρολαυτά, δε θα πρέπει να ξεχνάμε την απόδοση μνήμης που προσφέρει το γενικό μοντέλο.

5.1.2 Συνεισφορά της Πρόβλεψης στην Βελτιστοποίηση

Ερευνώντας πάνω στην ακρίβεια πρόβλεψης των διαφορετικών εκτιμητών είχε ως αποτέλεσμα να έχουμε ένα ικανοποιητικό μοντέλο για μελλοντικές τιμές καυσίμου. Κατά αυτό τον τρόπο, μπορούμε να αποκτήσουμε την απαραίτητη γνώση για τον βελτιστοποιητή δρομολογίου, έτσι ώστε να ελαχιστοποιήσουμε το κόστος ανεφοδιασμού. Όπως έδειξαν τα αποτελέσματα, χρησιμοποιώντας τον αλγόριθμο βελτιστοποίησης, που περιγράφηκε στο κεφάλαιο 3, και γνωρίζοντας *a priori* τις τιμές καυσίμου για κάθε κόμβο του γραφήματος, μπορούσαμε να πετύχουμε μια μέση μείωση 5.48% στο κόστος ανεφοδιασμού σε σύγκριση με το κόστος που θα είχαμε αν δεν γνωρίζαμε τις τιμές καυσίμου. Όπως μπορούμε, όμως, να παρατηρήσουμε, χρησιμοποιώντας τα διαφορετικά μοντέλα πρόβλεψης, μπορούμε να πετύχουμε μια μέση μείωση, που φτάνει το 5.20% (χρησιμοποιώντας τις καλύτερες τιμές παραμέτρων για το SVR και την TSA προσέγγιση τοπικού μοντέλου). Αυτό σημαίνει ότι έχοντας εξοπλιστεί με το κατάλληλο μοντέλο πρόβλεψης και χωρίς να γνωρίζουμε τις τρέχουσες τιμές καυσίμου, μπορούμε να πετύχουμε μια διαφορά της τάξης του 0.28% από τη βέλτιστη πολιτική ανεφοδιασμού.

Επίσης, είναι σημαντικό να τονίσουμε πως στην περίπτωση του γενικού μοντέλου, μπορούμε να πετύχουμε μείωση 3.42% για την DaF προσέγγιση και 4.82% για την TSA προσέγγιση.

5.2 Πιθανές Επεκτάσεις

Κατά τη διάρκεια αυτής της εργασίας προέκυψαν ορισμένα θέματα για περαιτέρω έρευνα. Μερικά από αυτά είναι τα εξής:

- Στην περίπτωση της προσέγγισης TSA, χρησιμοποιήσαμε την **αναδρομική** στρατηγική για πρόβλεψη πολλαπλού βήματος. Όπως είδαμε στο κεφάλαιο 4, αυτή η στρατηγική συνδυασμένη με την απόδοση του εκτιμητή πετυχαίνει χαμηλή ακρίβεια για μακροπρόθεσμους στόχους εξαιτίας των σφαλμάτων συσσώρευσης. Θα ήταν ενδιαφέρον να ερευνηθεί η επίδραση άλλων στρατηγικών πρόβλεψης πολλαπλού βήματος, όπως η Direct ή η MIMO στρατηγική[22].
- Όσον αφορά τους εκτιμητές Μηχανικής Μάθησης που χρησιμοποιήθηκαν, μπορούμε να επισημάνουμε ότι υπάρχουν μοντέρνες υπολογιστικές δομές αλγορίθμων που διατηρούν και ερμηνεύουν αποδοτικότερα την *μνήμη* που χρειάζεται ένα σύνολο χρονοσειράς. Ένα τέτοιο χαρακτηριστικό παράδειγμα είναι τα **Αναδρομικά Νευρωνικά Δίκτυα** και συγκεκριμένα τα **LSTM**. Αυτές οι δομές περιέχουν αναδρομικές υπολογιστικές μονάδες, που μπορούν να οδηγήσουν σε υψηλότερη ακρίβεια πρόβλεψης. Παρόλο που διαμορφώσεις αυτών ερευνήθηκαν και κατά τη διάρκεια αυτής της εργασίας, δεν παρατηρήσαμε καλύτερη απόδοση από το κλασικό MLP. Μια περαιτέρω πειραματολογία και παραμετροποίησης των LSTM θα ήταν χρήσιμη για την βελτίωση ακρίβειας.
- Όπως επισημάναμε στην εισαγωγή αυτού του κεφαλαίου, σχετικά με τα χαρακτηριστικά που χρησιμοποιήθηκαν στην DaF προσέγγιση, είναι ενδιαφέρον να δούμε πως άλλοι παράγοντες επηρεάζουν τις τιμές καυσίμου. Επίσης, σχετικά με το γενικό μοντέλο πρόβλεψης, προτείνουμε ότι η αξιοποίηση της εξάρτησης μεταξύ των διαφορετικών χρονοσειρών πρατηρίων θα μπορούσε να βελτιώσει την ακρίβεια πρόβλεψης. Δομές των Νευρωνικών Δικτύων, όπως είναι τα **Συνελικτικά Νευρωνικά Δίκτυα** είναι κατάλληλα για συσταδοποίηση χρονοσειρών.
- Κατά τη φάση της βελτιστοποίησης, περιγράψαμε έναν άπληστο αλγόριθμο που χρησιμοποιείται για την οργάνωση των δρομολογίων. Σύμφωνα με αυτόν, ορίζουμε ένα δρομολόγιο με βάση τα πρατήρια από τα οποία περνάει το όχημα. Σε ένα γενικότερο

πλαίσιο, είναι χρήσιμο να εφαρμόσουμε την ίδια πειραματολογία, επεκτείνοντας το *Simple Fixed-Path Vehicle Routing Problem* στο γενικό **Fixed-Path Vehicle Routing Problem**, όπου τα πρατήρια δεν είναι απαραίτητα κόμβοι του γραφήματος.[29]

Επίσης, κάναμε υποθέσεις ως προς την ταχύτητα, την κατανάλωση καυσίμου και τη χωρητικότητα καυσίμου στο όχημα που χρησιμοποιήσαμε για το benchmarking. Είναι χρήσιμο για το τελικό προϊόν να εξετάσουμε πως το σύστημα εισήγησης ανταποκρίνεται σε πραγματικές εφαρμογές, όπου έχουμε μεταβλητές ταχύτητες και κατανάλωση καυσίμου.

Βιβλιογραφία

- [1] *Fuel Price Monitor in Germany*
<https://www.spritpreismonitor.de/>
- [2] Manuel Frondel, Colin Vance, Alex Kihm.
Time Lags in the Pass-Through of Crude- Oil Prices – Big Data Evidence from the German Gasoline Market
Ruhr Economic Papers No. 573, 2015
- [3] Rob J Hyndman, George Athanasopoulos. *Forecasting: principles and practice*
<https://www.otexts.org/fpp>
- [4] Hand, D.: *Mining the past to determine the future: Problems and possibilities.*
International Journal of Forecasting, October 2008
- [5] Vapnik V: *The nature of statistical learning theory.*
Springer-Verlag, New York , 1995
- [6] S. S. Haykin: *Neural Networks and Learning Machines.*
Prentice Hall, 2009
- [7] Kyoung-jae Kim: *Financial time series forecasting using support vector machines*
Elsevier Neurocomputing, Volume 55, p. 307 – 319, 2003
- [8] Hasbi Yasin, Rezyzy Eko Caraka, Tarno and Abdul Hoyyi
Prediction of Crude Oil Prices using Support Vector Regression (SVR) with grid search – cross validation algorithm
Global Journal of Pure and Applied Mathematics, Volume 12, Number 4, pp. 3009–3020, 2016
- [9] T. Cheng, J. Wang and X. Li: *The Nonlinear Integration of Spatial and Temporal Forecasting by Support Vector Machines*
Fuzzy Systems and Knowledge Discovery, FSKD 2007. Fourth International Conference, 2007
- [10] Hamel, L.: *Knowledge Discovery With Support Vector Machine.*
United State Of America: A John Wiley and Sons, INC., Publication, 2009.
- [11] Smola A.J, *Regression estimation with support vector learning machines.*
Master's thesis, Technische Universitaet Muenchen., 1996
- [12] Xin-Yao Qian, Shan Gao.
Financial Series Prediction: Comparison Between Precision of Time Series Models and Machine Learning Methods
<https://arxiv.org/abs/1706.00948>
- [13] Timo Koskela, Mikko Lehtokangas, Jukka Saarinen, and Kimmo Kaski.
Time Series Prediction with Multilayer Perceptron, FIR and Elman Neural Networks.
In Proceedings of the World Congress on Neural Networks, 1996.

- [14] Guoqiang Zhang, B. Eddy Patuwo and Michael Y. Hu
Forecasting with artificial neural networks: The state of the art
International Journal of Forecasting Volume 14,p. 35–62, 1998
- [15] Juan Manuel Gimeno Illa, Javier Bejar Alonso and Miquel Sanchez Marre.
Nearest-Neighbours for Time Series.
SpringerLink Applied Intelligence, Volume 20,p. 21–35, 2004
- [16] Adela Sasu. *K-Nearest Neighbor Algorithm for univariate time series prediction*
Bulletin of the Transilvania University of Braşov, Vol 5, No. 2, 2012
- [17] Nesreen K. Ahmed, Amir F. Atiya, Neamat El Gayar, and Hisham El-Shishiny.
An Empirical Comparison of Machine Learning Models for Time Series Forecasting
Econometric Reviews, Volume 29, Issue 5–6, p. 594–621, 2010
- [18] Enzo Busseti, Ian Osband, Scott Wong.
Deep Learning for Time Series Modeling
<http://cs229.stanford.edu/proj2012/BussetiOsbandWong-DeepLearningForTimeSeriesModeling.pdf>
- [19] F.A. Gers, J. Schmidhuber, F. Cummins
Learning to forget: continual prediction with LSTM
ICANN '99, p. 850 – 855, 1999
- [20] Kline, D.M.: *Methods for multi-step time series forecasting with neural networks.*
Peter Zhang, G. (ed.) Neural Networks in Business Forecasting, pp. 226–250.
Information Science Publishing (2004)
- [21] Gianluca Bontempi, Souhaib Ben Taieb, Yann-Aël Le Borgne.
Machine Learning Strategies for Time Series Forecasting
Chapter in Lecture Notes in Business Information Processing, January 2013
- [22] Souhaib Ben Taieb, Gianluca Bontempi, Amir F. Atiya, Antti Sorjamaa.
A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition
Expert Systems with Applications 39,p. 7067–7083, 2012
- [23] Christoph Bergmeir, Rob J Hyndman, Bonsoo Koo.
A Note on the Validity of Cross-Validation for Evaluating Autoregressive Time Series Prediction
Computational Statistics and Data Analysis, Volume 120, April 2018,p. 70-83
- [24] Martin Ester, Hans-Peter Kriegel, Jiirg Sander and Xiaowei Xu :*A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*
KDD-96 Proceedings,AAAI, 1996
- [25] Derya Birant, Alp Kut: *ST-DBSCAN: An algorithm for clustering spatial–temporal data*
Elsevier, Data and Knowledge Engineering 60,p. 208–221, 2007
- [26] Shieu-Hong Lin, Nate Gertsch, and Jennifer R. Russell.
A linear-time algorithm for finding optimal vehicle refueling policies
Operations Research Letters, Volume 35, Issue 3, p. 290-296, 2007
- [27] Shieu-Hong Lin: *Finding Optimal Refueling Policies in Transportation Networks*
AAIM, Algorithmic Aspects in Information and Management, p. 280-291, 2008

- [28] Shieu-Hong Lin: *Multi-Objective Constrained Vehicle Refueling Planning: Complexity and Polynomial-Time Approximation Schemes*
Proceedings of the 2015 IEEE IEEM, 2015
- [29] Samir Khuller, Azarakhsh Malekian, Julian Mestre.
To Fill or Not to Fill: The Gas Station Problem
ACM Transactions on Algorithms, Volume 7 Issue 3, Article No. 36, 2011
- [30] Yoshinori Suzuki: *A decision support system of dynamic vehicle refueling*
Elsevier, Decision Support Systems 46, p. 522–531, 2009
- [31] Yoshinori Suzuki: *A decision support system of vehicle routing and refueling for motor carriers with time-sensitive demands*
Elsevier, Decision Support Systems 54, p. 758-767, 2012
- [32] Yoshinori Suzuki: *A variable-reduction technique for the fixed-route vehicle-refueling problem*
Elsevier, Computers and Industrial Engineering 67,p. 204–215, 2014
- [33] S. F. Love.: *Bounded production and inventory models with piecewise concave costs*
Management Science, 20:313-318, 1973
- [34] Szilárd Aradi, Péter Gáspár and Tamás Bécsi: *A Predictive Optimization Method for Energy-Optimal Speed Profile Generation for Trains*
14th IEEE International Symposium on Computational Intelligence and Informatics,p. 19–21 November, 2013
- [35] Jörn Boehnke: *Pricing Strategies, Competition, and Consumer Welfare Evidence from the German and Austrian Retail Gasoline Market*,2014
- [36] *U.S Energy Information Administration* <https://www.eia.gov/>