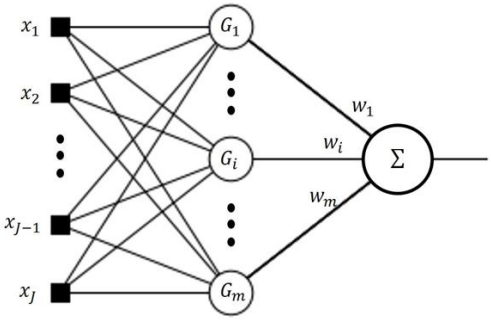
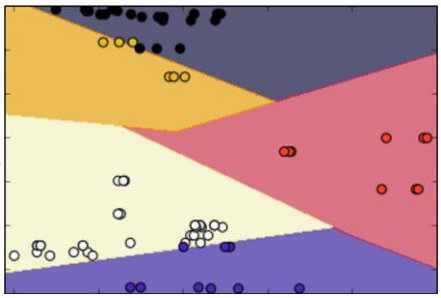
CHALMERS
UNIVERSITY OF TECHNOLOGY





Machine learning for vehicle concept candidate population & verification

Master's thesis in Applied Physics

Björn Grevholm

Department of Applied Mechanics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2017

MASTER'S THESIS IN APPLIED PHYSICS

Machine learning for vehicle concept candidate population & verification

Björn Grevholm

Department of Applied Mechanics
Division of Vehicle Engineering & Autonomous Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2017

Machine learning for vehicle concept candidate population & verification
Björn Grevholm

Master's Thesis 2017:39
ISSN 1652-8557
Department of Applied Mechanics
Division of Vehicle Engineering & Autonomous Systems
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: + 46 (0)31-772 1000

Cover:
Compilation of an image of Volvo car [1] in a wind tunnel an artificial neural network and a classification map.

Department of Applied Mechanics
Göteborg, Sweden 2017-06-15

Machine learning for vehicle concept candidate population & verification
Master's thesis in Applied Physics
Björn Grevholm
Department of Applied Mechanics
Division of Vehicle Engineering & Autonomous Systems
Chalmers University of Technology

Abstract

The aim of this M.Sc. thesis is to evaluate the potential of using machine learning to support concept phase decisions to balance the thermal properties of an automobile. With the use of computer scripts, the relevant measurement data is extracted from repositories and is used to train an artificial neural network which can identify the importance of the different parameters that are involved in tuning the vehicle thermal attributes.
After data for several car models has been used to train Machine Learning (ML) tools, this configuration used in predicting parameters affecting engine under hood thermal behaviour. A neural network based ranking procedure which may make it possible to reduce the order of concept decision space is also proposed. After several vehicle families gone through this prediction phase, a clustering of vehicle classes may allow for prediction and optimisation of new families, if errors due to assumptions and underlying mathematics are quantified.
The project has the added benefit of allowing Volvo Car Corporation (VCC) to reuse the large amount of data which are seldom used after the initial project delivery date. Measurements collected in VCC's wind tunnels are the main source of data for this thesis but the open-source script based method can be used on other type of data from other disciplines.
A possible outcome of the thesis might be recommendation for updated procedures in creating and storing data to easier integration into machine learning based investigations.

Key words: Machine learning, heat exposure protection, artificial neural networks, vehicle thermal properties, vehicle families, polynomial kernels, linear kernels, prediction, quantification of errors, regularization, radial basis functions, k-means, Support Vector Machines, Support Vector Regression, sorting, data storage, data management, wind tunnel tests

Acknowledgments

I would like to thank my supervisor Raik Orbay for the opportunity to participate in the development of this interesting way to solve engineering task and for his support throughout my project. I would also like to thank Magnus Bäckelie and all our colleagues at Volvo Car Corporation for their support and assistance. At Chalmers University of Technology I would like to express my gratitude to my examiner Simone Sebben for her valuable input and to Bernhard Mehlig and Marina Rafajlovic for years ago introducing me to machine learning and artificial neural network. At last, I would like to thank my family and friends for their help and support over the years.

Content

Abbreviation

ANN ..... Artificial Neural Network
BEV..... Battery Electric Vehicle
CAC ..... Charged Air Cooler
CC ..... City Cycle
CFD..... Computational Fluid Dynamics
CON ..... Change of NRMSE
CPU..... Central Processing Unit
FFNN..... Feedforward Neural Network
HCTR..... Hill Climb with Trailer
ICE..... Internal Combustion Engine
ML..... Machine Learning
MSE ..... Mean Squared Error
NRMSE..... Normalized Root Mean Squared Error
NVH ..... Noise Vibration and Harshness
OPM ..... Object Process Methodology
RBF ..... Radial Basis Function
RBFN ..... Radial Basis Function Network
RDE..... Real Drive Emissions
RL ..... Road Load
RLTR..... Road Load with Trailer
SHC..... Steep Hill Climb
SUV..... Sport Utility Vehicle
SVC..... Support Vector Classification
SVM ..... Support Vector Machine
SVR..... Support Vector Regression
VCC ..... Volvo Car Corporation

Notations

Roman upper case letters

$\mathbf{D}$	Matrix stabilizer
F	Function calculating the output of an ANN
G	Gaussian function
$\mathbf{G}$	Matrix with Gaussian function between input data and centers for the RBFN
$\mathbf{G}_0$	Matrix with Gaussian function between different centers for the RBFN
J	Number of element in the input data $\mathbf{x}$
K	Kernel function
N	Number of training data sets
W	Power

Roman lower case letters

b	Bias
f	Activation function
k	Number of clusters in k-means clustering algorithm
m	Number of neurons in a hidden layer.
n	The index of the data set.
r	Constant in support vector machine kernels
v	Signal from node before activation function
w	Weight (normal)
$\mathbf{w}$	Weight (vector)
w_i	Weight for input signal i
w_{ij}	Weight for input signal i to node j
$\mathbf{x}$	Input data
$\mathbf{x}(n)$	The n^{th} set of training data
x_i	The i^{th} input in $\mathbf{x}$
y	Output data

Greek upper case letters

$\mathcal{E}$	Regularized cost function
$\mathcal{E}_s$	Empiric cost function
Ω	Regularization function

Greek lower case letters

γ	Constant in support vector machine kernels
η	Learning rate
ζ	Signal from node after activation function
τ	Torque of crankshaft
$\boldsymbol{\mu}_\mathrm{i}$	Vector for the center of a radial basis function
ω	Angular velocity of powertrain crankshaft

1 Introduction

1.1 Background

The automobile industry is going through changes due to the increasing complexity of the product portfolio. The need to handle customer demands as environmentally as possible with products which have the highest possible quality and with least financial burden as possible; as well as doing this in a very short time is a real challenge.

In facing multiple challenges, traditional methods employed early in the development phases are growing too hard to manage due to the numerous degrees of freedom, as well as number of dimensions involved. Occasionally, the complexity is handled by means of prioritising concepts, which may leave plausible concept solutions out of the concept design space due to lack of knowledge on a product which in fact is aimed to arrive to the market years later.

For a new complexity there are new tools to consider. Many challenges faced by the information technology industries are handled using Machine Learning principles. Machine Learning, therefore, is a strong candidate to populate and validate concepts. In this sense the automotive industry can embrace the complexity, because innovations are through exploiting this complexity.

1.2 Aim of the study

During the early phases of product development there is a need to make decisions based on simplified models to determine in what direction the development should align. Today these decisions are made by experts based on the years on experience they possess. For this purpose, many tools are at hand at Volvo Car Corporation (VCC). For instance, the technical planning is done based on a functional disposition framework, where every system & component is connected to the customer needs instead of mere engineering performance cursors. The object process methodology is a means to establish the processes affecting objects in the above described framework. For further information the reader is advised to follow Törmänen [2]. As an example the Steep Hill Climb (SHC) test procedure is explained from the Object Process Methodology (OPM) point of view below. The schematic shows the traditional method used to determine if certain parts of the car will stay in the acceptable range when the car is subjected to the SHC-test. The schematic is depicted in object process methodology conventions.

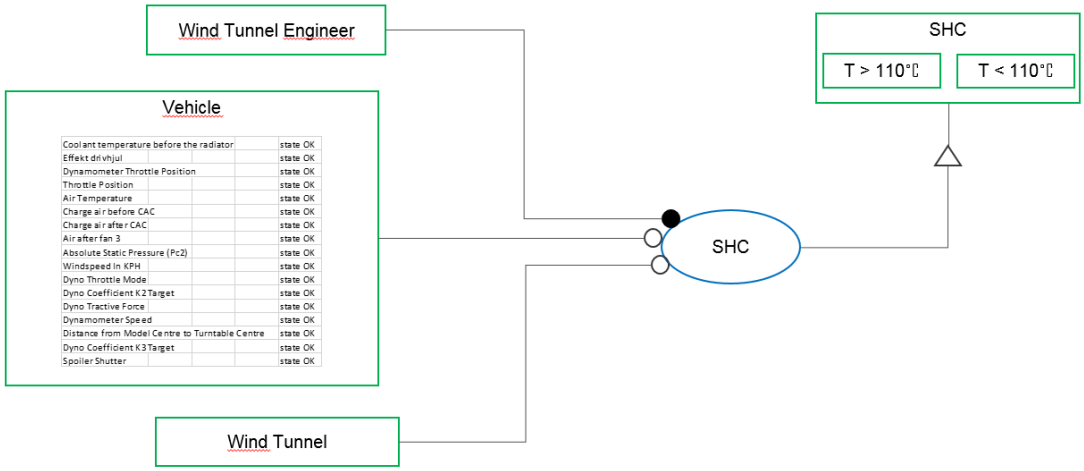


Figure 1.1. OPM schematic of SHC-test performed in a wind tunnel to determine if a critical system temperature will stay within an acceptable range.

VCC has been performing wind tunnel tests under many years leading to a huge database for several car classes. By means of machine learning (ML), the need for resource-intensive wind tunnel tests can be minimized:

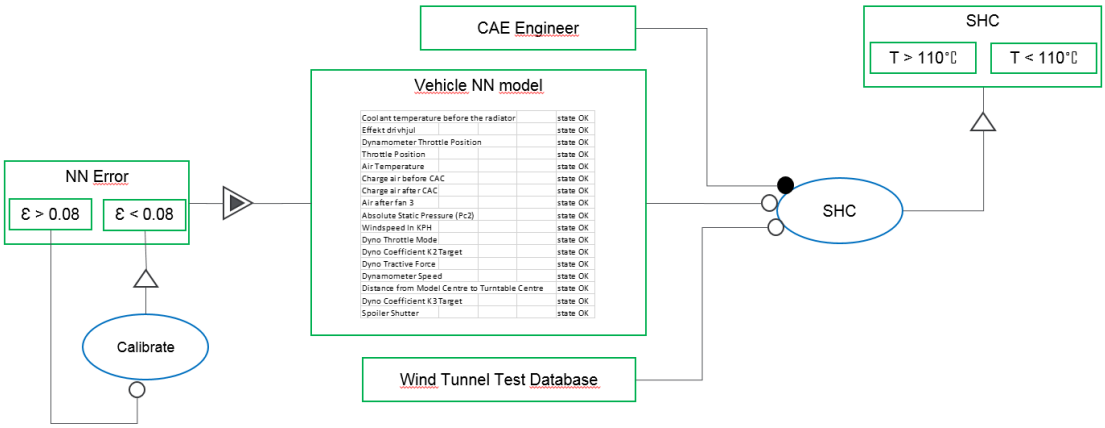


Figure 1.2. OPM schematic of ML usage to determine if a critical temperature will stay within an acceptable range.

With an appropriate ML technique, predictions of the temperature of the given car part may be performed with an acceptable margin of error, saving both time and money. This study aims to prove this as a possible method and to examine different methods for prediction.

1.3 Literature study

Compared to products which VCC is introducing nowadays to the market, vehicle models of 20 years ago are rather simple in system configuration. Vehicles are now connected and partially autonomous, they are also much more efficient. In the course of last 20 years, the industry have tested and built-up a lot of know-how. Although the high expertise involved, the complexity of the products are not allowing simple formulations to describe complex system-of-systems behaviour. As an example, propulsion systems get much more complex and the count of components constituting a whole internal combustion engine increases with every model year. To handle this new complexity, new methods are proposed. In [3], tools to handle Big-Data are recommended to tackle internal combustion engine complexity, due to the fact that lean manufacturing and other traditional 6 Sigma processes for waste reduction are still reactive strategies and will not provide break-through impacts on automotive original equipment manufacturers.

Theoretical information and instructions on how the ML tools should be constructed was taken mainly form Simon Haykin's book [4] but also from lecture materials from different universities available online [5], [6], [7]. Implementing these tools in a technical system required more technical papers [8], [9], [10]. Zhang, [8], uses machine learning to model a complex phenomenon like internal combustion engine emissions. It is shown that although it is not amenable to a simple formulation, there is a structure in the emissions data set which a computer quantifies mathematically.

During the course of the project the prospect of ranking the input parameters with different methods were explored [11], [12].

1.4 Method

This work is performed at the VCC Vehicle Propulsion- 97100 Propulsion Strategy & Innovation Department. The study is started with a literature study. Simultaneously, surveys with the staff from VCC Environment and Fluid Dynamics Centre, Thermodynamics group is conducted. In this way, related test procedures, databases, test set-ups and vehicle families are listed. Following the creation of access to databases, a sub-set of the thermodynamic test results are created locally. Principles of thermodynamic behaviour, as well as contemporary automotive systems are studied. Parallel to this, the study involved benchmarking several Python based tools for ML purposes.

Additionally an ANN tool is scripted from scratch to comprehend the process thoroughly. In the course of the study, the detail in the models are increased. Benchmarking methodology made it possible to devise the optimum error/activation functions for the task in hand. After a working ANN set-up prescribed, methods from literature are used to sort the wind tunnel test parameters in their importance. This set-up used further to cluster vehicle classes using ML.

Python version 2.7 was chosen as the primary programming language because of its open sourced nature and the ML tools available. The chosen tool for this project is the Scikit-learn module. These tools are complemented with python tools scripted from scratch by the author of the M.Sc. thesis.

2 Theory

2.1 Machine learning

Machine learning (ML) is a discipline in computational science, where algorithms make predictions by means of inferred conclusions based on a set of data, instead of using analytic or numerical approaches. The principle is based on features (which are chosen to sufficiently describe the system at hand) through which a statistical algorithm will "learn" via a process of pattern recognition. In that way the code is able to describe a system without an explicit programming [6].

2.1.1 Supervised vs unsupervised learning

Two of the most basic categories of machine learning are supervised and unsupervised learning. In both categories N sets of training data $\mathbf{x}(n), n = 1,2, \dots N$ will be studied, but only in the case of supervised learning will these sets of data have a corresponding label, y(n).

Unsupervised learning uses the distribution of the data to either clustering them into groups or determines odd cases which differ substantially from the majority of the data. Supervised learning uses algorithms to connect a new set of data with the right label y. It is referred to either classification or regression depending on the nature of the label y. Discrete labels can be separated by classification, while continuous labels are approximated with regression [13].

2.2 Artificial neural networks

Artificial neural networks (ANN) or just neural networks are computational models based on the workings of the human brain. These networks are collections of connected nodes which regulate flow of signals among each other in a very similar fashion to how brain cells work. Because of the similarities between ANN and the human brain, the nodes in the network is often referred to as neurons, a synonym for nerve cells. Many network architecture organise the neurons into layers where the first and last layers are referred to as input and output layer respectively and any layer between them are referred to as hidden layers.

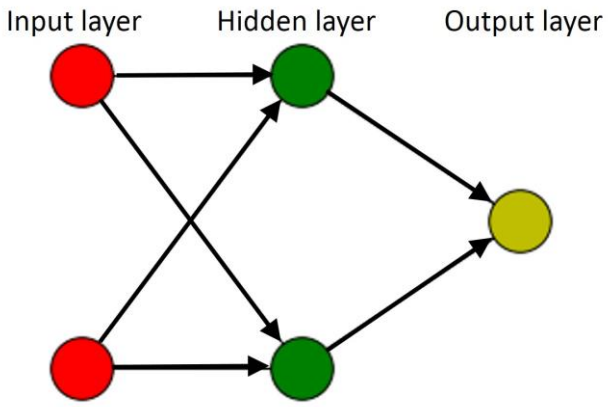


Figure 2.1 Illustration of a simple artificial neural network with one hidden layer with signals among the nodes.

Neural networks are often used in machine learning where a large number of inputs and the corresponding outputs are known, but the exact relationship between them are unknown. By initialising an ANN and training it with the known data, a relationship structure may be prescribed without knowing the underlying physics. The network can then be trained with new data and new results can be predicted. As ANN is trying to prescribe a relationship, it is important to quantify the amount of error created. In this study, methods based on literature study will be devised to handle the error.

2.2.1 Feedforward neural network

Feedforward neural network (FFNN) is arguably the simplest type of artificial neural networks. The neurons are organized into different layers where signals only can be sent in one direction. The signal x_j from the j^{th} neuron in the previous layer are multiplied with a weighting value, w_j. The sum of all the multiplications are then inserted into an activation function together with a constant bias, b and the result is fed to the next layer in the network.

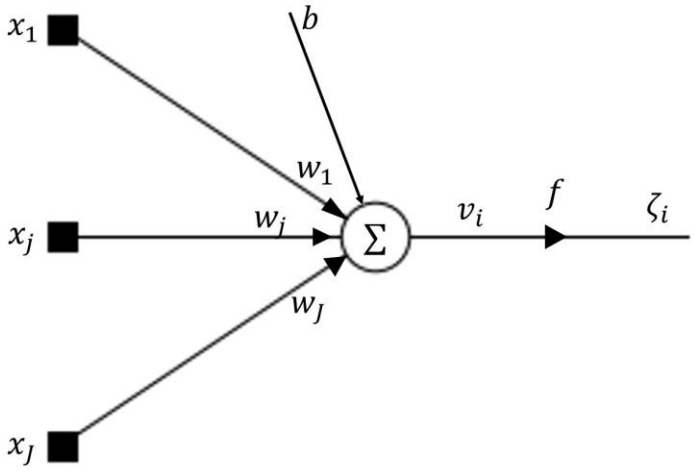


Figure 2.2. Illustration of a neuron in a feedforward neural network.

The output signal ζ_i from neuron i can be computed as

$$v_i = \sum_{j=1}^{J} w_j x_j + b, \qquad \zeta_i = f(v_i) \tag{2.1}$$

The connection among the different layers are based on the choice of activation function and the weighting value saved in the weight-vector **w**. Some authors add the bias term b to the sum, while other authors combine weight and bias by adding an extra weight and an input that is equal to 1. The properties of the network is based on the number of hidden layers, the number of neurons in those layers and the activation function used in each layer.

The choice of activation function varies depending on the range of the wanted output but easily differentiable activation functions in the following table are the most common:

Table 2.1. List of common activation function for a feedforward neural network.

Name	Formula	Range	Differential function
Logistic	$f(x) = \frac{1}{1+e^{-x}}$	$(0,1)$	$f'(x) = f(x)\big(1-f(x)\big)$
Hyperbolic tangent	$f(x) = \tanh(x)$	$(-1,1)$	$f'(x) = 1 - f(x)^2$
Identity	$f(x) = x$	$(-\infty, \infty)$	$f'(x) = 1$

The reason why easily differentiable activation functions are common is due to the fact that they allow the system to be trained with backpropagation. By calculating the output, of the FFNN, for a random set of input data $\mathbf{x}(n)$, the weight of the signal between node i in one layer and node j in the next layer can be updated:

$$w_{ji}^{new} = w_{ji}^{old} + \eta \delta_j x_i \tag{2.2}$$

Here η is the learning rate, x_i is the signal from the previous node and δ_j is the error signal for node j. This is done for all elements w_{ji} in $\mathbf{w}$. The error signal depends on the differential of the nodes activation function and the error between the systems predicted output and its expected output for input data $\mathbf{x}(n)$. By repeating this process for different $\mathbf{x}(n)$, a value for w_{ji} which minimizes the error in the prediction can be reached by a gradient descent method. This process is referred to as "training" the system. The errors in prediction that remains in a trained system can be caused by the limits of the system or by the gradient descent converging to a local error-minimum instead of a global error-minimum. The number of hidden layer and the number of neurons in them effects how the network behaves.

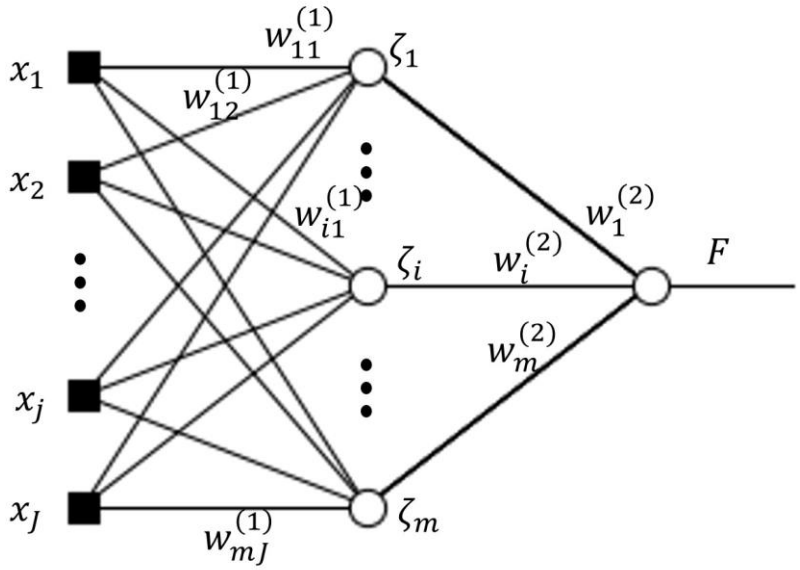


Figure 2.3 Illustration of a feedforward neural network with one hidden layer.

The output for a FFNN with one hidden layer and a liner activation function in the output layer can be expressed as the following function:

$$F(\mathbf{x}) = \sum_{i=1}^{m} w_i^{(2)} f\left(\sum_{j=1}^{J} w_{ij}^{(1)} x_j\right) \tag{2.3}$$

During the project, the activation function in the first layer f that was chosen as the logistic function due to the nature of the problem at hand.

2.2.2 Radial basis function network

Radial basis function networks (RBFN) are similar to a three layered FFNN with a difference in the process between the input layer and the hidden layer. The nodes in the hidden layer represents centers $\boldsymbol{\mu_i}$ in the same space as the input parameters. The signal between the input layer and the hidden layer depends on the Euclidian distance between the input-vector $\mathbf{x}$ and the vector for each of these centers $\boldsymbol{\mu_i}$, $\|\mathbf{x} - \boldsymbol{\mu_i}\|$. This means that there is no weight-vectors and the activation function is a radial basis function that depends on $\|\mathbf{x} - \boldsymbol{\mu_i}\|$.

A common radial basis function is the multivariate Gaussian function:

$$G_i = G(\boldsymbol{x}, \boldsymbol{\mu_i}) = \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x} - \boldsymbol{\mu_i}\|^2\right) \tag{2.4}$$

where $\boldsymbol{\mu_i}$ is the vector for the i^{th} center and σ is the standard derivation. The location of these centers can be chosen randomly or distributed more evenly with the aid of different algorithms.

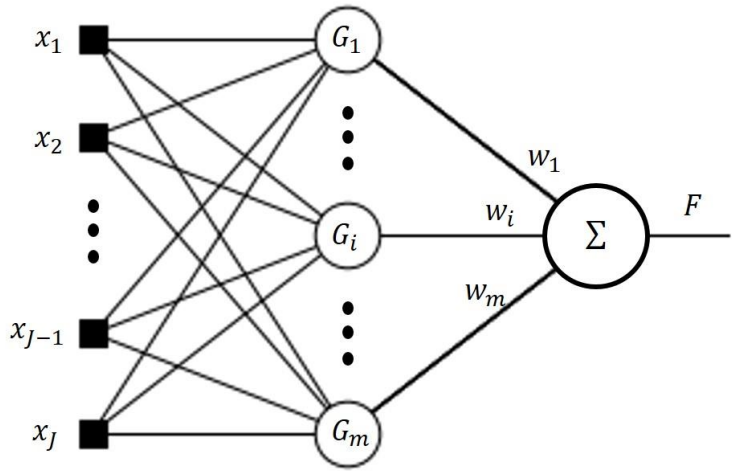


Figure 2.4. Illustration of a radial basis function network. G_i is the Gaussian function with respect to the i^{th}center, which has the coordinate $\boldsymbol{\mu_i}$.

A network where the RBF has m centers and has a linear activation function in the output layer has the following solution:

$$F(\mathbf{x}) = \sum_{i=1}^{m} \mathbf{w}_i G(\mathbf{x}, \boldsymbol{\mu}_i) \tag{2.5}$$

When working with RBFN, it is important to allow for a good coverage both for the input data and for the centers. Because the RBF depends on the Euclidian distance to the centers, these centers have to be well distributed and populated in relatively large numbers. In a system with a large number of input parameters, the same number of centers are often used.

2.2.3 Regularization of ANN

To prevent overfitting, regularization is an important mathematical operation[6]. It involves minimizing the regularized cost function $\mathcal{E}$, which is sum of the empiric cost function $\mathcal{E}_s$ and a regularization function Ω.

$$\mathcal{E}(F) = \mathcal{E}_s(F) + \Omega(F) \tag{2.6}$$

For an ANN problem, the empiric cost function will measure the error of the solver and the regularization function quantify how complex the solver is, by minimizing the sum of these a solution that balances accuracy and simplicity can be found. One advantage with radial basis function networks is that there is a very convenient way to analytically calculate a weight vector $\mathbf{w}$ to be optimized according to regularization, instead of approximate it with backpropagation.

Haykin [4] demonstrates the analytic solution for Eq. (2.6) with the solver F from Eq. (2.5) for the RBF network with linear activation function in the output layer.

$$\mathcal{E}(F) = \sum_{n}^{N} \left(y(n) - \sum_{j}^{m} w_j G(\mathbf{x}(\mathrm{n}), \boldsymbol{\mu}_\mathbf{i}) \right)^2 + \|\mathbf{D}F\|^2 = \tag{2.7}$$

$$\|\mathbf{y} - \mathbf{G}\mathbf{w}\|^2 + \|\mathbf{D}F\|^2$$

Here $\boldsymbol{G}$ is a $N \times m$- matrix with the element of the Gaussian function between the N input vectors and m centers with the elements $\mathbf{G}_{ni} = G(\mathbf{x}(n), \boldsymbol{\mu}_\mathbf{i})$, $\boldsymbol{D}$ is a chosen matrix stabilizer that quantifies the complexity of F. With the right choice of $\boldsymbol{D}$ can the regularization function can be rewritten as:

$$\|\mathbf{D}F\|^2 = <\mathbf{D}F, \mathbf{D}F> = \mathbf{w}^\mathbf{T}\mathbf{G_0}\mathbf{w} \tag{2.8}$$

where $\boldsymbol{G_0}$ is a symmetric $m \times m$- matrix with the Gaussian function between the different centres and has the elements $\mathbf{G}_{ki} = G(\boldsymbol{\mu_k}, \boldsymbol{\mu_i})$. The weight $\mathbf{w}$ which minimizes Eq. (2.7) can be calculated by deriving the function with respect to $\mathbf{w}$ and setting it to zero:

$$\mathbf{G^T G w} - \mathbf{G}^{\mathrm{T}}\mathbf{y} + \mathbf{G_0 w} = 0 \tag{2.9}$$

from this, the minimizing weight vector can be calculated as:

$$\mathbf{w} = (\mathbf{G^T G} + \mathbf{G_0})^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{y} \tag{2.10}$$

Instead of training, using regularization to compute the weight vector prevents a too complex solution that can lead to overfitting. Therefore for problems which require large number of iterations of the backpropagation, there is the possibility of saving computation time. Regularization can also be used in other ways in ANNs, when the complexity of the solution needs to be balanced with the accuracy. It is a matter of obtaining the proper regularization function Ω.

2.3 Support vector machines

Another way to perform supervised learning is the use of support vector machines (SVM). Unlike ANN, these machine learning models train during the predictions instead of training the system before making predictions. To determine the best predicted output, SVMs compares every test data $\mathbf{x}$ and training data $\mathbf{x}(n)$ with the aid of a kernel function $K(\mathbf{x}(n), \mathbf{x})$. Support vector machines were used for both regression and classification in the M.Sc. project.

For classification problems, the SVM looks at data in two classes with designation $y = 1$ and $y = -1$. The predicted class for an input data x is the sign of the multiplication of the kernel with the result y_n and constant α_n, summed over all training data:

$$f(x) = sgn\left(\sum_{n=1}^{N} y_n \alpha_n K(\mathbf{x}(n), \mathbf{x}) + b\right) \tag{2.11}$$

When there is more than two different classes the Python script will create multiple classifiers, so there is one classifier for every permutation of two classes [14], [15]. This is called support vector classification (SVC).

Support vector regression (SVR) calculates the continuous label for the data set with the function:

$$f(x) = \sum_{n=1}^{N} (\alpha_n - \alpha_n^*) K(\mathbf{x}(n), \mathbf{x}) + b \tag{2.12}$$

The value of the constants α_n and α_n^* and the bias b are optimized by the Python script in a way similar to regularization, minimizing the sum of the error and a quantifier for the complexity. The kernels which were used are displayed in Table 2.2

Table 2.2. List of SVM kernels available in Python toolkit Scikit-learn.

Name	Kernel function $K(\mathbf{x}(n), \mathbf{x})$
Linear kernel	$\mathbf{x}^{\mathrm{T}}\mathbf{x}(n)$
Polynomial kernel of the j^{th} degree	$(\gamma\mathbf{x}^{\mathrm{T}}\mathbf{x}(n) + r)^{j}$
Radial basis function kernel	$\exp(-\gamma\|\boldsymbol{x} - \boldsymbol{x}(\mathrm{n})\|^2)$

2.4 Clustering

To get a favourable distribution of the centers in the RBFN and to examine the unsupervised learning results, it is possible to implement a *k*-means clustering algorithm. *k*-means algorithm divides *N* sets of data into *k* different clusters, each with a center $\boldsymbol{\mu}$. A set of data $\mathbf{x}$ is assigned to the cluster of which it has the smallest Euclidian distance, $\|\mathbf{x} - \boldsymbol{\mu}\|$, to. The algorithm determines the *k* centers, which minimizes the total Euclidian distance for the *N* sets of data to its assigned cluster center. The centers in the radial basis function network can be distributed with this algorithm.

2.5 Thermodynamics -The need for cooling

The first law of Thermodynamics prescribes rules for an energy balance on any arbitrary control volume; i.e., energy cannot be created nor destructed but only can be transformed. The second law of Thermodynamics states that heat always flows from warmer objects to cooler objects if no external force is applied. These are the essential principles which govern the behaviour of an internal combustion engine (ICE). An ICE therefore can only work between two finite temperatures. This leads to a necessity to dissipate heat due to production of work. To make this process as efficient as possible, cooling systems are designed. For common automotive applications, a liquid coolant circulates the engine, absorbing heat until it arrives at the radiator where the heat is dissipated to the ambient air.
The above described process leads to different engine component temperatures, as well as high air temperatures downstream various heat exchangers. To quantify the systems and processes, experimental and computational studies are performed during vehicle thermodynamic development. Accordingly, this project is based on the data from wind tunnel measurements to balance and tune thermodynamic attributes of vehicles.

Although the data is obtained from thermodynamic development projects, the reported method is generic in nature and as stated in the literature, may be applied to other disciplines in a similar fashion.

3 Problem formulation

The powertrain for the vehicle is designed by the Vehicle Propulsion department and Thermodynamics group secures heat exposure prevention of the vehicle components when this powertrain is installed in a vehicle. In that sense, Thermodynamics group at the Volvo Car Corporation Environment and Fluid Dynamics Center owns the task of dissipating efficiently the excess heat energy from internal combustion engines. This thesis is set to determine the prospects of determining the functioning of the heat exposure prevention based on data from Thermodynamics group, using machine learning principles. This approach is expected to bring three advantages:

1- Once the ML tool is at place, the operator does not need to know the all the underlying physics of the system in question. Using data, a black-box model can be prescribed relatively easy. The specialized tools like CFD require years of training but a black-box model is substantially straightforward.
2- Specialized tools may need geometrical or non-geometrical data as inputs, which may not be available. Data-driven approach include effects from these inputs implicitly.
3- Compared to other methods, ML-based approaches may provide with concept decision support faster due to the simplicity of the framework.

Networks like those described in the previous chapter are used on data provided by Thermodynamics group from the Volvo Car Corporation Environment and Fluid Dynamics Center. To be able to work on without effecting the on-going projects, a sub-set of the huge database is created locally. The work process comprises choice of a Key Performance Index (e.g., a temperature or a power signal) and then training the ANN structures based on wind tunnel data. After the ANN have been tuned to predict the chosen KPI output with low error, other studies of sorting the features, clustering the data and model reduction can be performed.

3.1 Data generation

The data sets used were generated by Volvo Car Corporation's wind tunnel facility in the Torslanda plant. The facility features an 8.15 meter fan that can generate winds up to 250km/h [16]. All the data sets used where thermodynamics-and-cooling-performance tests performed during 2016. Five different car models where chosen due to their different properties and the availability of data.

VCC wind tunnel is a state of the art facility which includes boundary layer suction slits and a moving ground. For a routine test, full-size test vehicle is placed on a balance and effects of studied design changes are quantified by their consequences in drag coefficient magnitude. Tests aiming thermodynamic performance constitutes analyse of logs from several thermocouples positioned in the vehicle under hood. Beyond these, parameters are also extracted from the vehicle on-board diagnostics software and from instruments in the wind tunnel. The parameters are recorded every second. The five cars are referred according to Euro Car Segment [17] and the type of fuel it uses.

Table 3.1. The codes for the cars used with the types of car and volume of the engine compartment.

Name	Engine compartment volume	Type of car
E-Diesel	Low	Executive car
E-Gasoline	Low	Executive car
C-Gasoline	Low	Medium car
J-Diesel	High	SUV
J-Gasoline	High	SUV

The low car is used to describe vehicles with relatively smaller engine compartments. Sedans and kombi vehicles have smaller engine compartment volume than that of SUVs. The Medium car is the smallest car in the collection, the two executive cars are larger and both belong to the same series of cars but use different fuels and therefore different engines. The last two cars are sport utility vehicles (SUV) with higher engine compartment volumes. The gasoline-driven SUV is a smaller model than the diesel-driven SUV.

3.2 Key performance index

A key performance index (KPI) is used as objective to train ML algorithms. Initially temperature of the coolant fluid after the radiator was examined. Due to excessive fluctuations in data that the ANN could not keep up with, a new KPI was devised. The KPI chosen was power and was determined by the engine speed multiplied with the torque of the crankshaft, a common method of determining the power exerted by the engine.

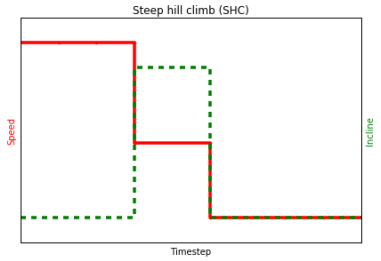
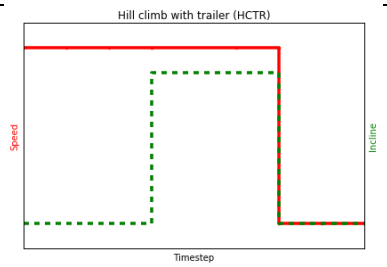
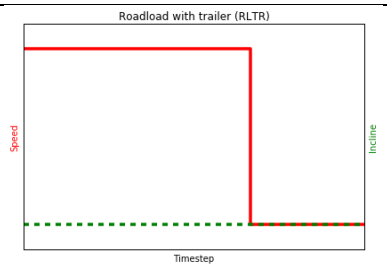
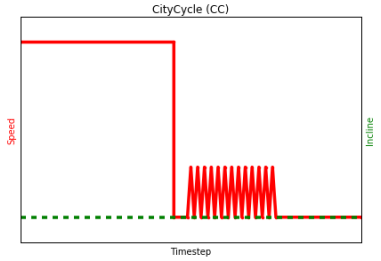
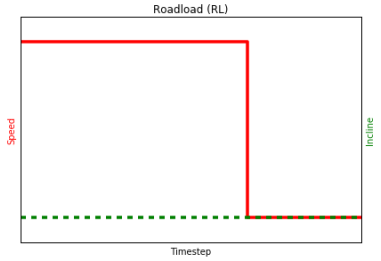
As the wind tunnel measurements are performed at different boundary conditions as speed, inclination and temperature were altered, KPI from the each boundary condition alternatives were used as the output of the artificial neural networks.

3.3 Wind tunnel tests

The wind tunnel measurements are performed in standardized test routines which simulate different driving scenarios. Several tests are performed in different ambient temperatures to simulate the markets distributed into several world regions. As high power needs of the engine causes higher dissipated heat powers, tests which involve towing trailers are often performed with several trailers of different weights. Being another critical condition, tests simulating vehicle behaviour on inclinations are done with loads on rollers carrying the test vehicle in the wind tunnel.

Some of the tests include a conditioning phase where the tunnel simulates running on a flat road surface at medium speed to heat up the engine and components. Afterwards, there is a soaking phase where the fan and the car engine is turned off. A list of different tests with illustrations of speed and inclination can be found in Table 3.2 and an enlarged image of the SHC test with phases can be found in Figure 3.1.

Table 3.2, Describes the different test types of test procedures used for thermodynamic development. Illustration also shows of how the speed and inclination varied during the course of a single test.

Name	Code	Conditioning	Note	Illustration
Steep hill climbing	SHC	Yes	Drive up to a hill for a period of time	
Hill climbing with trailer	HCTR	Yes	Drive up to a hill with a trailer for a period of time	
Roadload with trailer	RLTR	No	High speed with trailer from cold start. Continue until stabilized temperatures reached.	
City Cycle	CC	Yes	Cycle of starts and stops.	
Roadload	RL	No	High constant speed on flat road.	

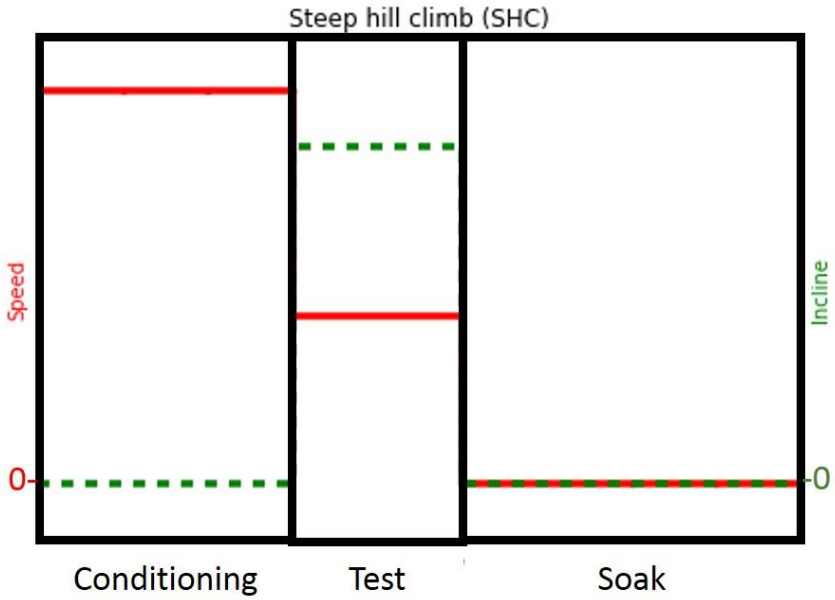


Figure 3.1 *A close up of the three phases of a steep hill climb test. First the engine is warmed up during the conditioning. Then the test starts, the speed is decreased and inclination is simulated. Finally the engine is turned off and the cooling of the car is recorded.*

4 Extraction of parameters from wind tunnel database

The data generated from the wind tunnel experiments are saved in Microsoft Excel worksheets in a certain way. The first sheet contains the parameters' name, a short description, name of the sensor that recorded the data and similar information, are registered in the file. Each of these Excel documents are around 1 GB in size.

A problem which was encountered early on in the project was the lack of standardisation of this information set. Data generated from the same car had the same name and descriptions, but between different cars this name convention was not kept. Most of the information is collected by sensors which have to be mounted in the car prior to the experiment. The work-intense process of installing these sensors caused that only the sensors the test designer asked for are installed This still means that most experiments comprise of data from between 350 and 400 parameters logged once every second for between 4000 and 5000 seconds.

4.1 Finding common parameters

With the assistance of scripted Python-modules, the Excel sheets from each wind tunnel test are read and a list of measured parameter-descriptions are created. This list is then compared with all the other lists in the created database and parameters which are not present in all of the databases are removed. Due to variation in naming of the parameters, this comparison was done on the descriptions of the parameters at the first page of every Excel sheet. Some important parameters which were measured in all vehicle test cases but had different description tags, were treated as special cases to account for this inconsistency. Being parameters to assure the proper functioning of the tunnel, some parameters were removed as they lacked any real information content for the purpose of this thesis.

4.2 Parameters

A total of 42 parameters was used for as features for the machine learning framework.

- **Temperature measurements**
  - Air after cooling fan
  - Air before and after CAC
  - Coolant before and after radiator
- **Wind tunnel control parameters**
  - Ambient temperature
  - Inclination
  - Wind speed
  - Solar simulation intensity
  - Various temperatures
- **Vehicle geometry**
  - Grill and spoiler area
  - Shutter area
  - Rolling drag
  - Rolling drag/velocity
  - Drag/velocity
  - Mass
- **Engine**
  - Crankshaft torque
  - Engine rpm

The only parameter which was not taken from the wind tunnel database was the geometrical information of the cars: the area of the grill, the spoiler and the shutters behind them. These parameters control the airflow to the engine.

4.3 Parameter ranking

Due to the high-dimensionality of the input data, a ranking procedure is prescribed to sort the data in their importance. Ranking the input parameters allows for estimation how much impact each of the parameters have on the performance of the car by means of imposed changes in KPI. The method of ranking are done by measuring the imposed change in the normalized root mean squared error (NRMSE) [11], [8] [1] when a ML case is run with and without the concerned parameter.

Mean squared error (MSE) is a method for estimating the error of a function by calculating the average difference between a set of expected data $\hat{y}(n)$ and a corresponding set of calculated data $y(n)$:

$$\mathrm{MSE} = \frac{1}{N}\sum_{n=1}^{N}(\hat{y}(n) - y(n))^2 \tag{4.1}$$

As the name suggest is NRMSE a normalised root of the MSE:

$$\mathrm{NRMSE} = \frac{\sqrt{\mathrm{MSE}}}{\hat{y}_{max} - \hat{y}_{min}} \tag{4.2}$$

With this normalization can different type of data be quantified equally and their error compared.

The Change of NRMSE (CON)-test is performed by construction and training a machine learning-method to make a prediction of KPI. The NRMSE between the prediction and the experimental data is then calculated. Afterwards, one of the input parameters of the ML tool is removed and the tool is redone and trained to work with one less parameter. The NRMSE between the new prediction and the experimental data is then calculated. The parameter is then returned and another parameter removed in its place. The difference in NRMSE for the network with all parameters and the network with one parameter missing is that parameter's CON-value.

Depending on the goal of the ranking, two alternative CON-value based method can be utilized [11]. Large changes in NRMSE mean that the parameter is important while small changes mean the modified parameter does not affect the network at all. The alternative method is to single out the parameter which gives the largest decreases in NRMSE. A large decrease in NRMSE could possibly be a sign that this network would work better without this parameter. Both these methods will be considered. Following the ranking procedure, the

[1] [11] demonstrated Change of MSE, [8] recommended that NRMSE was used to determine the accuracy of networks prediction.

overall dimension of the ML-case can be reduced by omitting unimportant parameters. This is a way to reduce the order of the model as well as increase the speed of the remaining ANN.

4.4 Creating neural network for prediction

Conforming to ML practices, before inserting the parameters from the wind tunnel into the different neural networks, each of them are normalized to between 0 and 1 [8]. Data from cars different but investigated incorporating the same type of test, are combined into N sets of training data, where N is the combined number of time-steps for every test. The input data is saved in a matrix, while the output data is saved as a vector.

In literature, input data is separated into different subsets as 70% training, 30% test set; alternatively 60% training set, 20% Cross-validation set and 20% test set [7].

5 Results

The trained networks are tested on a single vehicle test case at the wind tunnel. The results from a SHC test for the vehicle of class E with Diesel powertrain are plotted in all the subsections to allow for a comparison. SHC test procedure was chosen, because it was the most common vehicle test case in the studied database.

5.1 First ML attempt with temperature signal as KPI

Several wind tunnel parameters were investigated to be chosen as KPI and output of the various networks. Early tests were done using the temperature of the coolant at the radiator exit as KPI. The following graph was created with a FFNN with 15 neurons in the hidden layer.

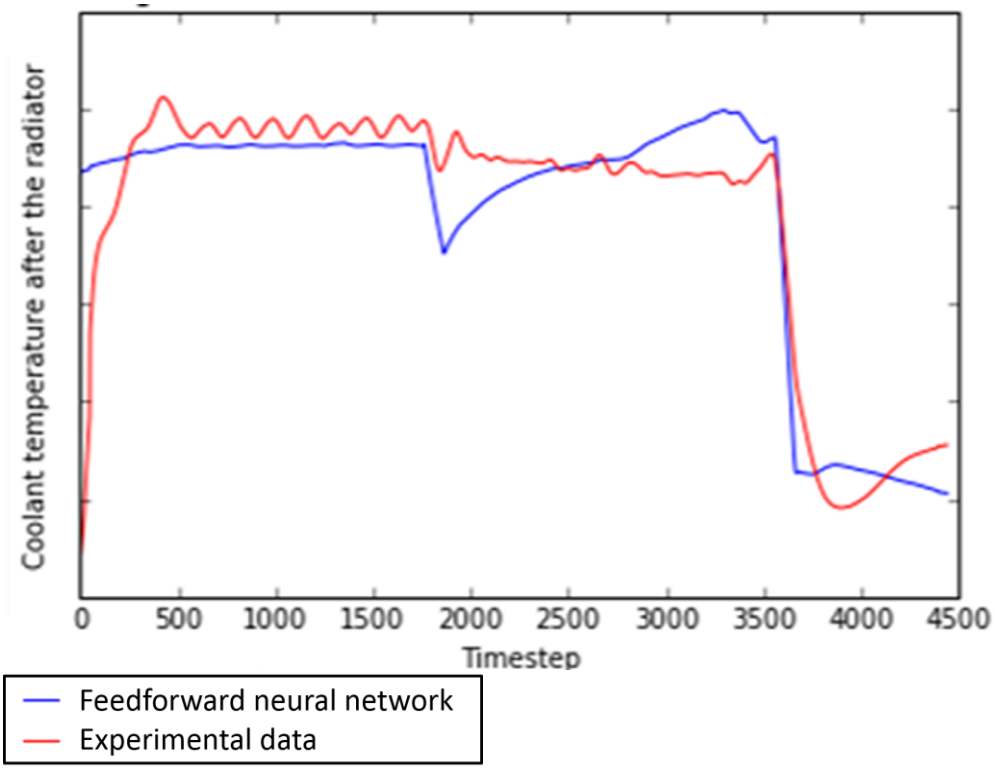


Figure 5.1 *Experimental and FFNN predicted data for the temperature of the coolant when it leaves the radiator in a steep hill climb test*

The graph in Figure 5.1 shows the problem encountered while using a temperature as KPI to train the ANN. The FFNN prediction is able to approximate the coolant temperature with an acceptable margin of error. The NRMSE for this plot was 0.1053, recommended acceptable error is 0.08 [8]. Here the error depend a lot on the initial conditioning phase of the engine and the vehicle components which is done in every wind tunnel vehicle test cases. Here it was observed that the network is not able to follow this initiation phase efficiently. Neither the designed FFN network was able to handle the fluctuations of the temperature signal. What it was able to do was predict the temperature accurately enough to determine what range the experimental temperature lies in which means the schematic in Figure 1.2 could be possible to achieve.

5.2 Updated KPI for ML

After several discussions another KPI was chosen. Being a more direct signal for the engine behaviour, a new KPI, the power of the engine, was decided. The power is calculated by the torque τ of the crankshaft multiplied with the rotational speed ω of the crankshaft and a constant to transform to the unit horsepower:

$$W = \tau \times \omega \times \frac{1}{5252}$$

ω is also known as the engine's rotations per minute (rpm). Because of spikes in the data, for this new KPI, a filter was also used one the data to smooth these out. With a Python routine, a median filter analyses the vector that is the output and changes the value of one element in it to the median of the surrounding elements.

The rest of the results focus on power signal as an output.

5.3 Prediction using feedforward neural network

The result of a FFNN with 20 neurons in hidden layer resulted in a prediction that could handle the step-like nature of the key performance index, but the accuracy was not satisfying. Changing the number of neurons in the hidden layer did not result in greater accuracy.

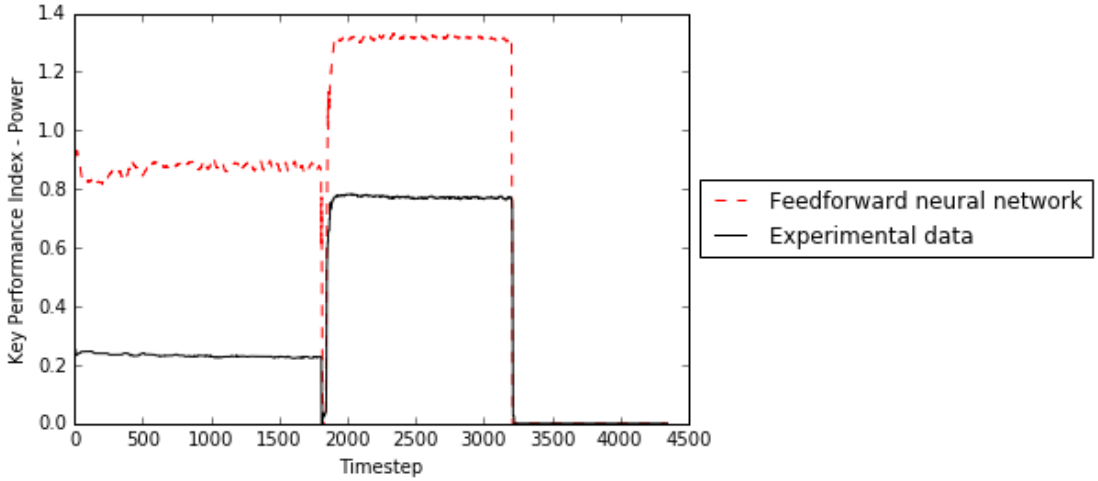


Figure 5.2. The experimental results for a Steep Hill Climb-run for the car from vehicle class E-Diesel. The predicted results from a feedforward neural network with one hidden layer is also illustrated. The FFNN used 20 neurons in the hidden layer.

5.4 Radial basis function network

With the radial basis function the center for the hidden layer were determined by a k-means clustering algorithm and the weight was calculated with Eq. (2.9). The entire set of the input data was feed to a *k*-means algorithm and *k* different clusters were formed. These centers of these clusters also functions as the centers for the RBFN. The number of centers and the dimension of the input vector are equal. The result are strongly different from the FFNN. Both the predictions for initial conditioning phase and the majority of the hill climb phase have achived a very close approximation. The spike the key performance index have at the start of the hill climb phase is the only part that the networks have problem while fiting. This

finding is conform with the results reported by Zhang [8] as the ANN method has known limitations for fast transient behavior.

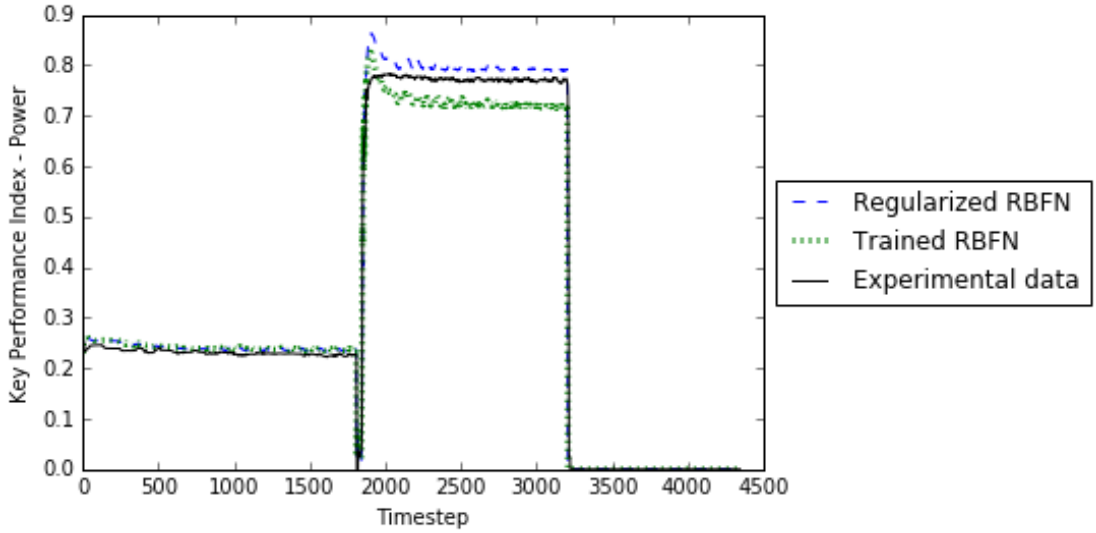


Figure 5.3. The experimental results for a Steep Hill Climb-run for the car E-Diesel. The predicted results from a radial basis function network is also illustrated.

5.5 Support vector regression

The SVR from the Scikit-learn module was performed with two different kernels governing the behaviour between the training data and test data. The result of the 3rd degree polynomial kernel is not able to follow the complexity of the data and only results in a constant value for SHC. The radial basis function kernel instead follows the experimental data accurately, as illustrated in Figure 5.4.

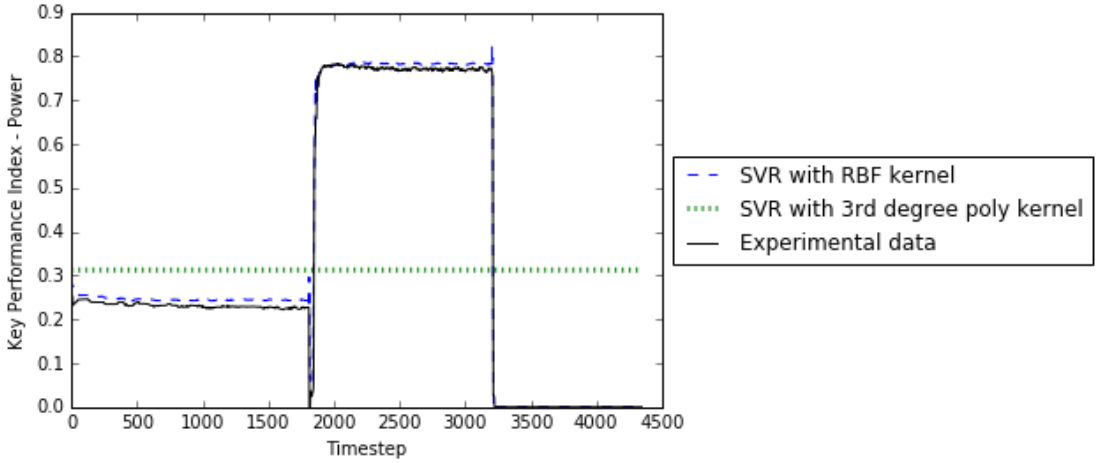


Figure 5.4. The experimental results for a Steep Hill Climb-run with the predicted results from a support vector regression with an RBF kernel and a 3rd degree polynomial kernel.

5.6 Accuracy and time

To compare the accuracy of the different ways of predictions, five different ML test runs for each type of vehicle tests were randomly chosen and the NRMSE between the predicted KPI and the experimental data was calculated and displayed in Table 5.1 and Figure 5.5.

Table 5.1. NRMSE for different prediction methods and different types of wind tunnel tests. The range of the error for each method is displayed at the bottom.

Test	Regularized RBFN	Trained RBFN	FFNN	SVR with RBF kernel	SVR with 3rd degree polynomial kernel
SHC	0.04	0.07	0.750	0.04	0.36
RL	0.02	0.06	0.175	0.05	0.31
RLTR	0.02	0.19	0.326	0.07	0.56
HCTR	0.04	0.05	0.187	0.04	0.34
CC	0.03	0.09	0.126	0.04	0.11
Range	0.02-0.04	0.05-0.19	0.12-0.75	0.04-0.07	0.11-0.56

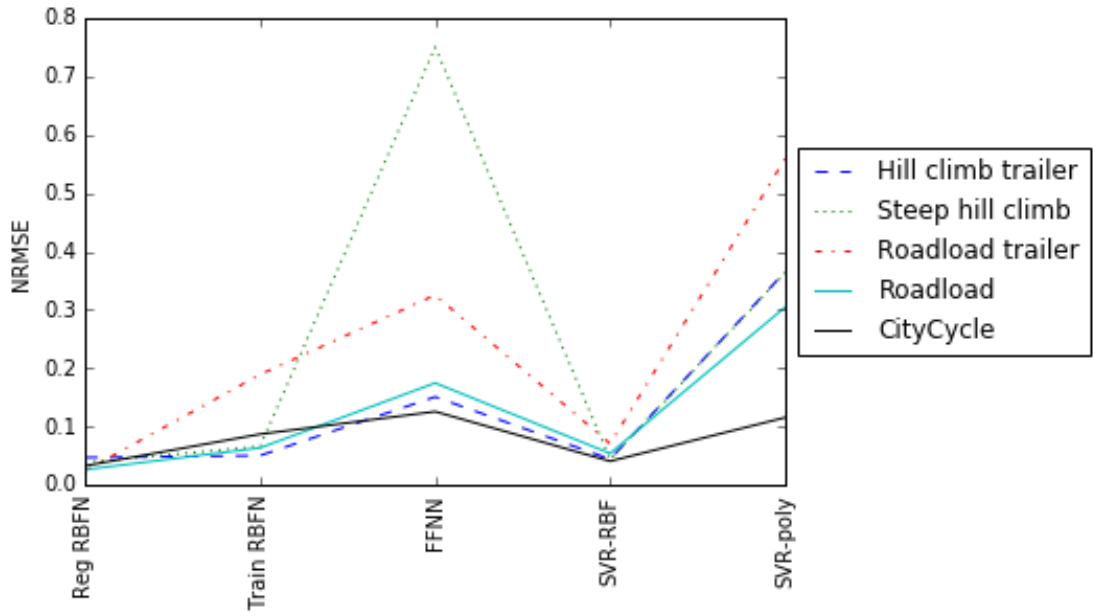


Figure 5.5. The error from the different types of test for the different prediction methods.

The results shows that for all the different types of vehicle tests, the greatest error is attained while using FFNN-predictions and the SVR with the 3rd degree polynomial kernel. The best predictions are done by the regularized RBFN and the SVR with RBF kernel. The different vehicle test cases reacted very differently with the trained RBFN, some were almost as good as the regularized version while others were worse. Only the regularized RBFN and SVR with RBF kernel accomplised an overall NRMSE of below 0.08 which was recommended [8].

The elapsed CPU-time for each of these ML predictions vary a lot depending on the metod used. In one case of *N* sets of data it takes less than 100 seconds to analyticaly calculate the

solution of the weight-vector, for the radial basis function network, with the help of regularization. After this, the network was trained and the prediction of one test took less than 1 second. To perform predictions with SVR for the same sets of data it took 800 seconds for polynomial kernel and 4000 seconds for the RBF-kernel. This substantial time difference between a compuational and analytical solution would be even more important when the number of training sets becomes even higher.

The general trend in the automobile industry is towards more time critical concept phase operations, thus fast & accurate concept studies are crucial.

5.7 A comparison for all studied vehicle classes

Figure 5.6 contains the SHC tests from all the different cars. The x-axis depicts the KPI for all test at the same point in the test cycle. The y-axis shows the temperature of the air after the engine fan; which is used here as an overall thermal representation of the engine compartment. The results from the experiment are shown together with the results from the RBFN and the SVR with RBF kernel.

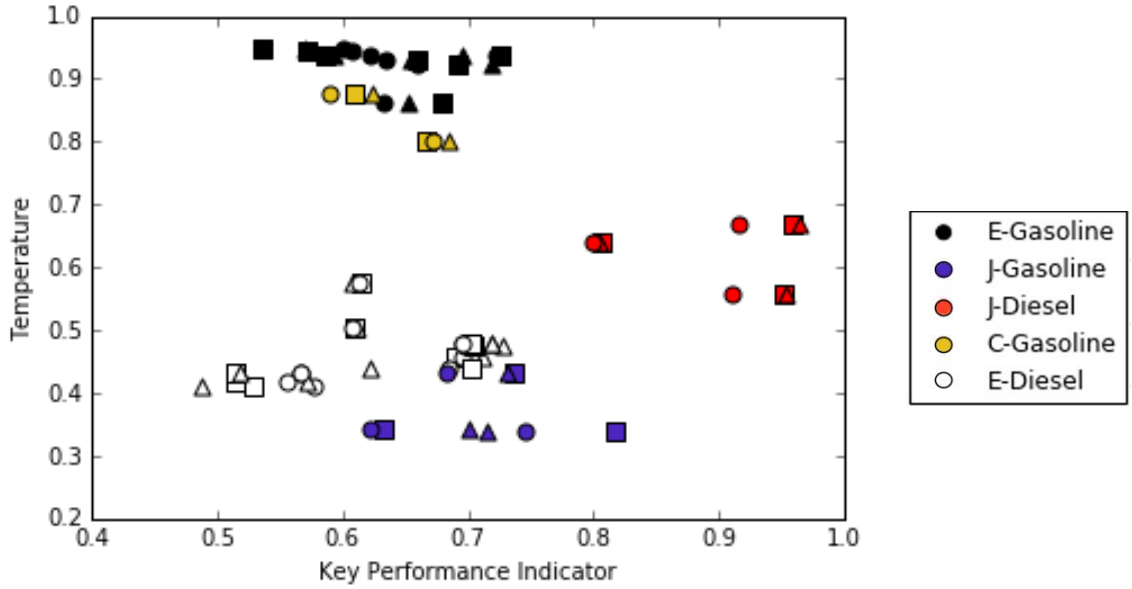


Figure 5.6 *Experimental results (●), RBFN predictions (■) and SVR with RBF kernel (▲) for different cars.*

The relationship between the performance of the car and temperature of the engine bay illuminates the differences between the different vehicle classes in a clear way. The two E-vehicle class cars are the sedan and kombi version of the same car and require about the same power for the same task but the gasoline engine generates more heat. The E-vehicle class Gasoline has more in common with C-vehicle class Gasoline, which also has a smaller engine compartment and uses the same fuel. J-vehicle class Diesel is the largest car, so that is would require the greatest amount of power to perform the same task. The reason for intermingling between the J-vehicle class Gasoline and the E-vehicle class Diesel is the author of this thesis not able to elucidate.

5.8 Classification

Because of the range of different car models VCC have on the market, it would be useful to group them into classes to easier see patterns. This may help to determine in which areas the company may profit investing research and development operations. There are of course already classification based on design commonalities but as will be seen, the model of engine and type of fuel will have great impact on the performance of the automobile. Alternatively, the method reported here can be used with other data related to vehicle performance and elucidate patterns not obvious to the experts.

The following graphs feature the relationship between the KPI and the temperature right after the automobile's cooling fan to represent an overall indicator for the temperature of the engine compartment. A snapshot of the relationship is taken during the test-phase of the wind tunnel test i.e. after conditioning and before the soak phases. The graphs shows the experimental data and the predictions from the RBFN and SVR with RBF-kernel.

The supervised classification problem was performed with SVC functions in the open-source Scikit-learn module of the Python. For the five cars described in Table 3.1, different methods of determining the classification problem are applied. On the following set of plots, the colour of the background indicates the cluster that the classifier classes that area as, while the colour of the data point indicates the known result.

The data which was classified was data taken from the steep hill climb tests during the hill climb phase of the test sequence. The graphs shown uses a normalized temperature of the position behind the cooling fan of the engine compartment on one axis and the normalized KPI on the other.

5.8.1 Vehicle classification using linear kernel

The simplest kernel presented in Table 2.2 is the linear kernel, which results in some rather accurate results.

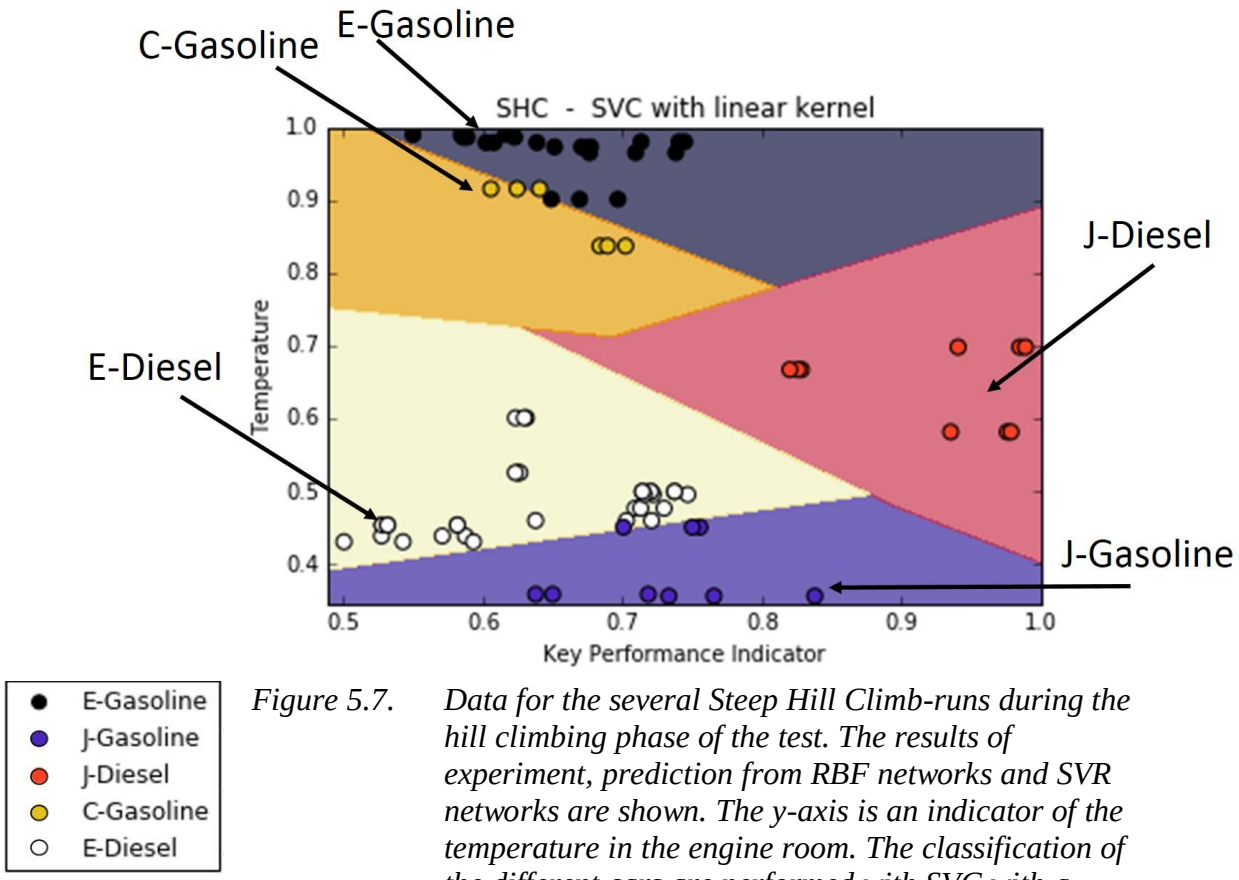


Figure 5.7. *Data for the several Steep Hill Climb-runs during the hill climbing phase of the test. The results of experiment, prediction from RBF networks and SVR networks are shown. The y-axis is an indicator of the temperature in the engine room. The classification of the different cars are performed with SVC with a linear kernel*

As the work of Bäck [9] reports, the approximations comprising linear kernels are providig reasonable results. This may be an advantage for concept studies as the linear algorithms are faster than algorithms including higher order polynomials.

5.8.2 Vehicle classification using polynomial kernel

Classification with a polynomial kernel of the 3rd degree gives less accurate results than the linear kernel because some of the C-Gasoline data points does not fall within the C-Gasoline classification zone.

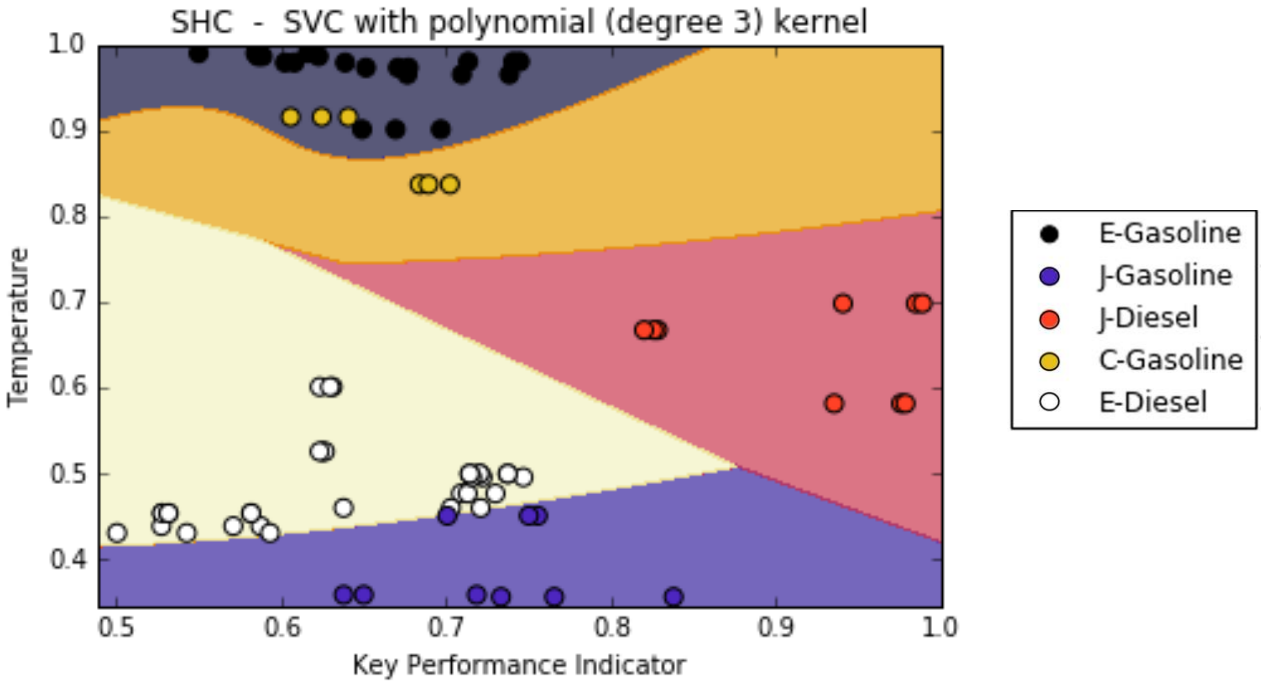


Figure 5.8. *Data for the several Steep Hill Climb-runs during the hill climbing phase of the test. The results of experiment, prediction from RBF-networks and SVR-networks are shown. The classification of the different cars are performed with SVC with 3rd degree polynomial kernel*

Additional ML tests pointed that the 5th degree polynomial kernel is able to place C-Gasoline in the right area, but the prediction with higher order polynomials gets a very complex form compared to the linear kernel.

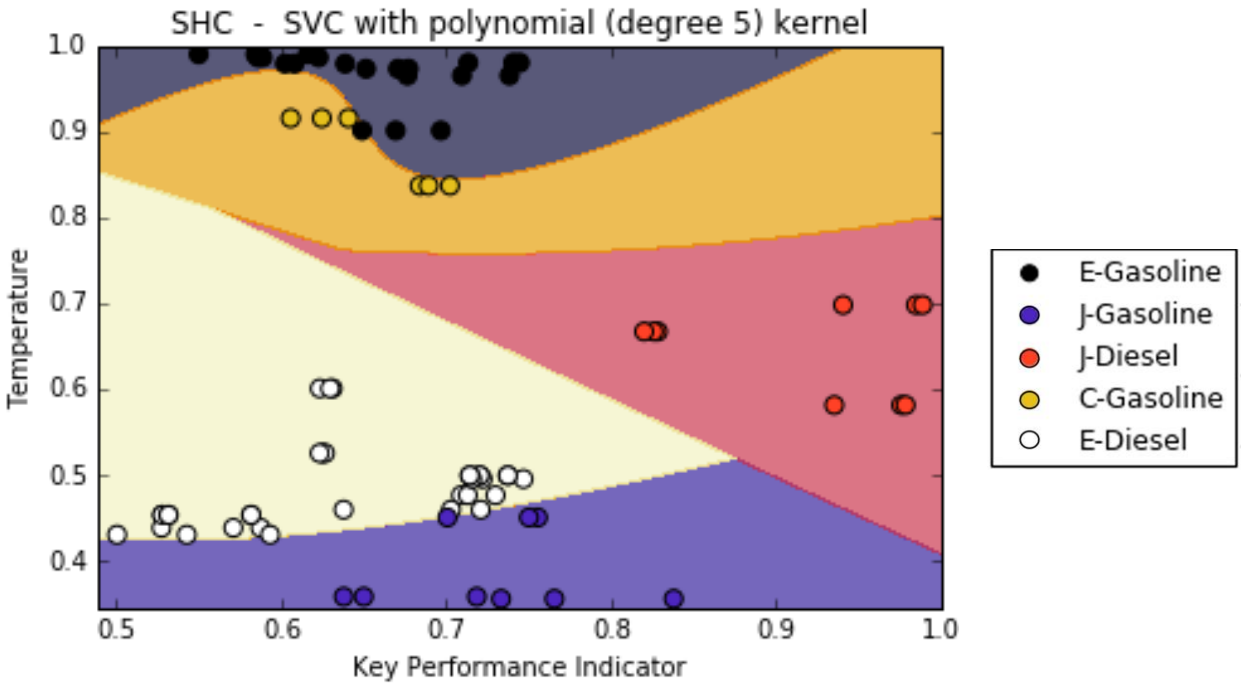


Figure 5.9. Data for the several Steep Hill Climb-runs during the hill climbing phase of the test. The results of experiment, prediction from RBF-networks and SVR-networks are shown. The y-axis is in indicator of the temperature in the engine room. The classification between the different cars are performed with SVC with 5rd degree polynomial kernel.

A comparison between the linear and polynomial shows a difference in the upper right corner. Both polynomial kernels identify this area as belonging to C-Gasoline while the linear kernel assigns this area to E-Gasoline. Because this area is "empty" there is nothing to indicate what class it should belong to. This may be another output of this M.Sc. thesis that the company would invest more experimental resources proactively in line with the clustering results above.

5.8.3 Vehicle classification using RBF kernel

The RBF kernel worked very well for regression but its limitation is very apparent when it comes to classification. When a radial basis function is used in RBFN, a great amount of design space coverage is needed to provide accuracy. SVC puts one class as the default class and the rest covers an area around its training points. The size of these areas are determined by the parameter γ from Table 2.2. The greater the constant γ, the smaller the standard derivation σ. In this sense we have problems in the areas where there is less amount of vehicle test data, thus VCC may try to distribute the experimental work accordingly.

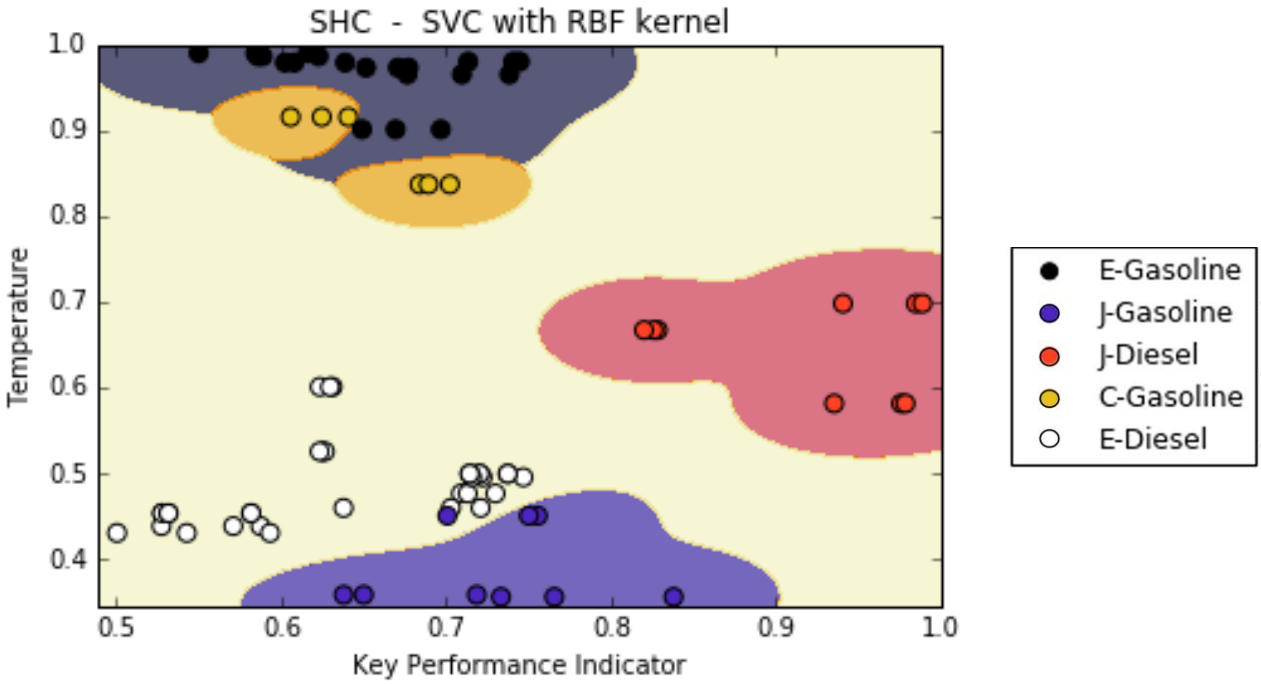


Figure 5.10. Data for the several Steep Hill Climb-runs during the hill climbing phase of the test. The results of experiment, prediction from RBF-networks and SVR-networks are shown. The y-axis is in indicator of the temperature in the engine room. The classification between the different cars are performed with SVC with radial basis function kernel

5.8.4 Vehicle classification using other vehicle tests than SHC

SHC test was chosen because it had the largest amount of data and because it had the best and most clear limits between the classes in the temperature-KPI space. But one other wind tunnel test also gave similar results. In Figure 5.11 is the results for the Hill Climb with Trailer (HCTR) test is plotted.

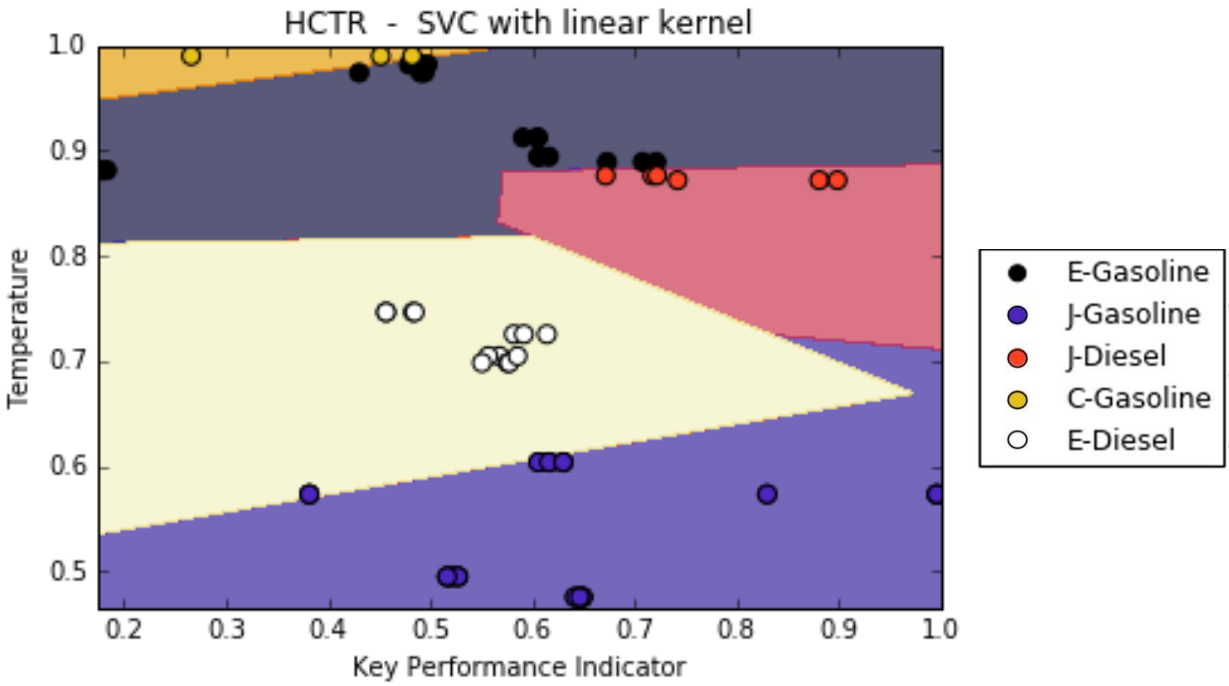


Figure 5.11. Data for the several Hill Climb with trailer-tests during the hill climbing phase of the test. The results of wind tunnel experiments, prediction from RBF-networks and SVR-networks are shown. The y-axis is in indicator of the temperature in the engine room. The classification between the different cars are performed with SVC with linear kernel

In this tests some pair of classes are closer to each other while others are positioned further apart. Due to the fact that wind tunnel tests for trailers of different weights are depicted here, there is a greater scatter between some of the classes than there was in the SHC test.

The City-Cycle test did not result in this kind clear results in this KPI-temperature space, because the constant starting and stopping made prescribing a good reference point for ML impossible. The Roadload and Roadload with Trailer-tests were also not possible to allow for good data in this KPI-temperature space, because these tests were made in a wide range of higher speeds. The Roadload at 180 km/h and at 120 km/h from the same car are too different to create these groups.

5.8.5 Classification for wind tunnel tests

Analysing and classifying different tests from the same car also has merits because it would help determine in what temperature-power intersection different tests are positioned in.

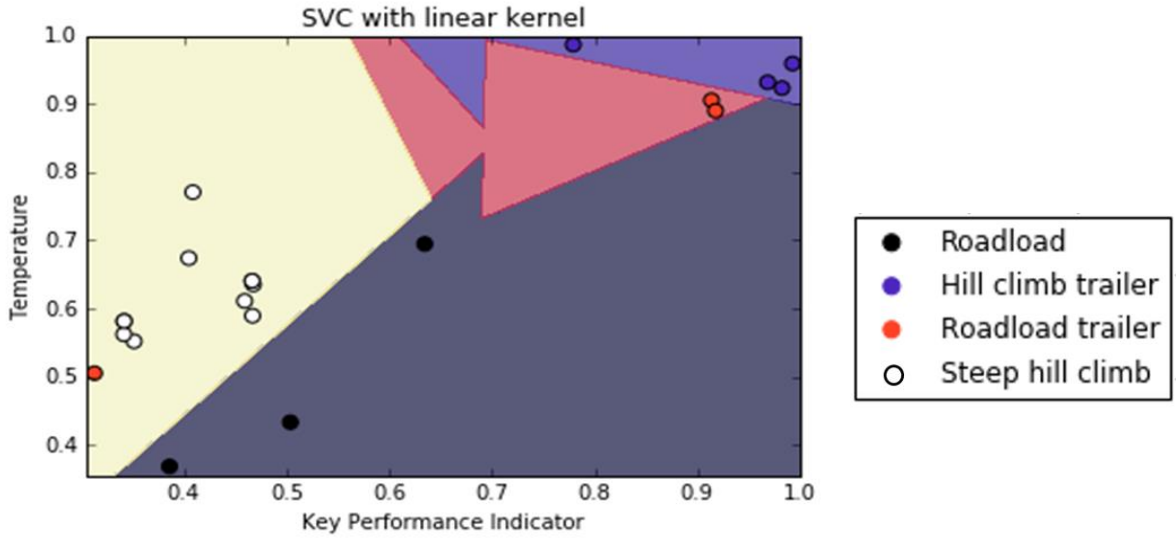


Figure 5.12. Data for the several different types of tests with the E-Diesel car. The results are from wind tunnel experiments. The y-axis is in indicator of the temperature in the engine room. The classification between the different cars are performed with SVC with linear kernel

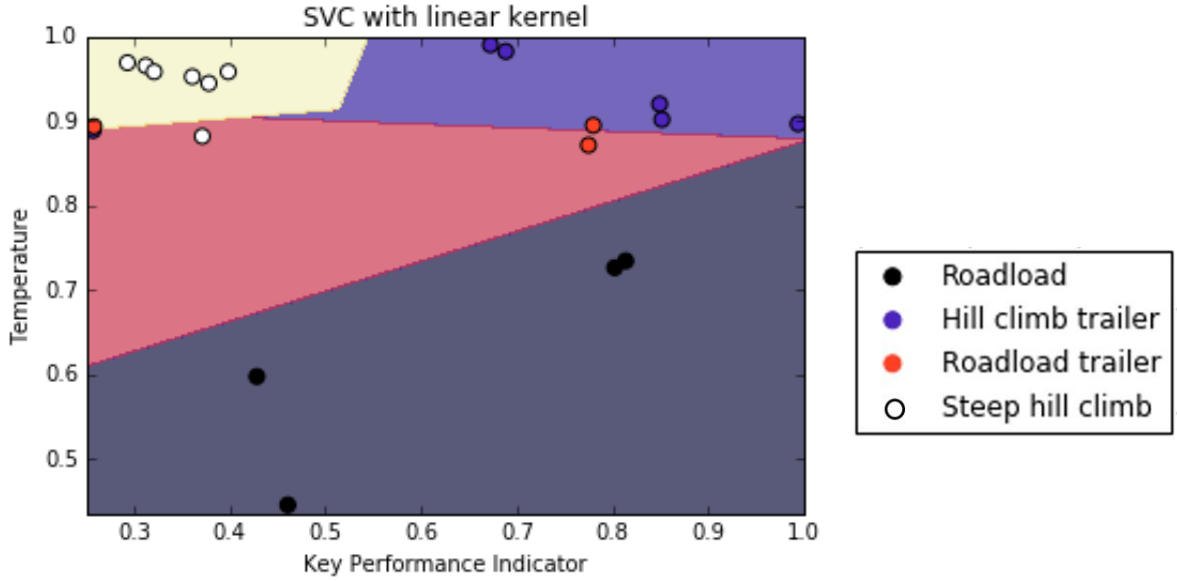


Figure 5.13. Data for the several different types of tests with the E-Gasoline car. The results are from wind tunnel experiments The y-axis is in indicator of the temperature in the engine room. The classification between the different cars are performed with SVC with linear kernel

Figure 5.12 and Figure 5.13 contain four of the wind tunnel tests from the car E-diesel and E-Gasoline respectively. The classification by linear kernel is not as accurate as in the case of only SHC test. This is because these tests vary much more in their nature and the sample size is smaller. But is is still possible to distinguish an indication of where each class of future wind tunnel tests will be placed in this space.

5.9 Clustering

Figure 5.14 shows that a cluster analysis, using the *k*-means method are not able to accurately discerning the 5 different cars used in the analysis. Alternatively, if the number of clusters are set to 3 an interesting phenomenon is seen in Figure 5.15.

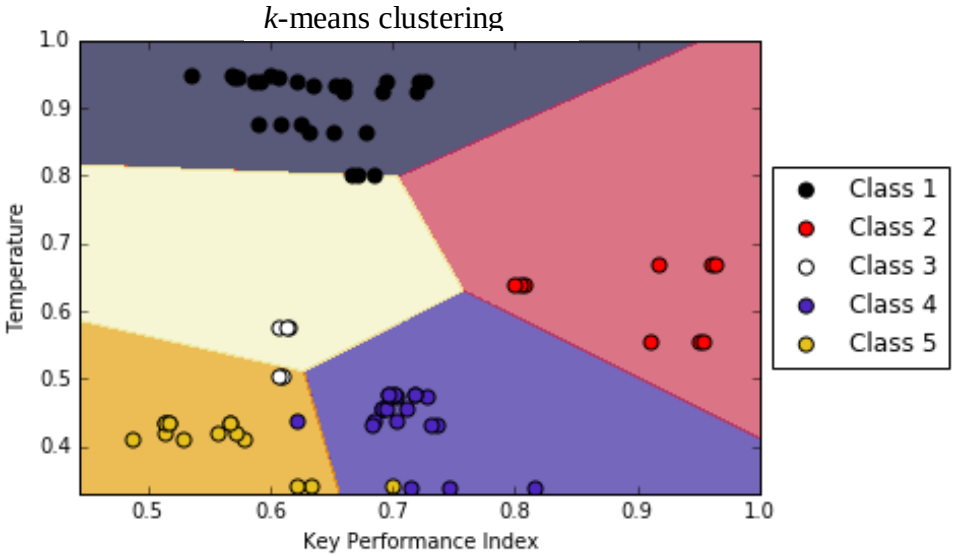


Figure 5.14 k-means clustering for steep hill climb test with 5 clusters.

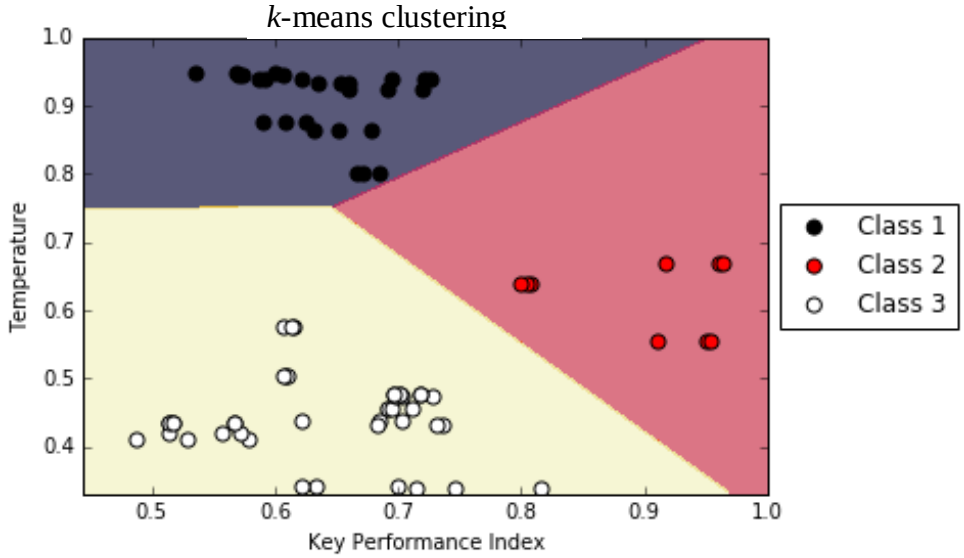


Figure 5.15 k-means clustering for steep hill climb test with 3 clusters.

In both Figure 5.14 and 5.15 the class designated as Class 1 consists of the gasoline fuel cars with low engine compartment volume, while Class 2 is the SUV designated J-Diesel with a large engine compartment volume. The remaining class or classes comprise of the cars E-Diesel and J-Gasoline.

The rest of the vehicle classes are clustered into Class 3, which includes the two cars with the coolest engine temperature, the gasoline SUV and the low engine compartment volume diesel car. This Class 3 is somewhat hard to comprehend, due to the difference in these car classes, but the existence of classes 1 and 2 in their current format indicates that gasoline fuelled cars with low engine compartment volume and diesel fuelled SUVs can be effortlessly separated in the feature space of engine temperature and KPI.

5.10 Sensitivity study

The results of the RBFN predictions above shows that the method can recreate data accurately. This approach may allow for sensitivity studies as well as predictions. If it is to be used for future predictions of overall thermal behaviour of concept vehicles, the technique has to be able to handle changes in the data in a reasonable way. In Figure 5.16 the same data has been feed to the network while one parameter has been changed. The area of the grill in the front of the car are changed in intervals of 5% and the results plotted.

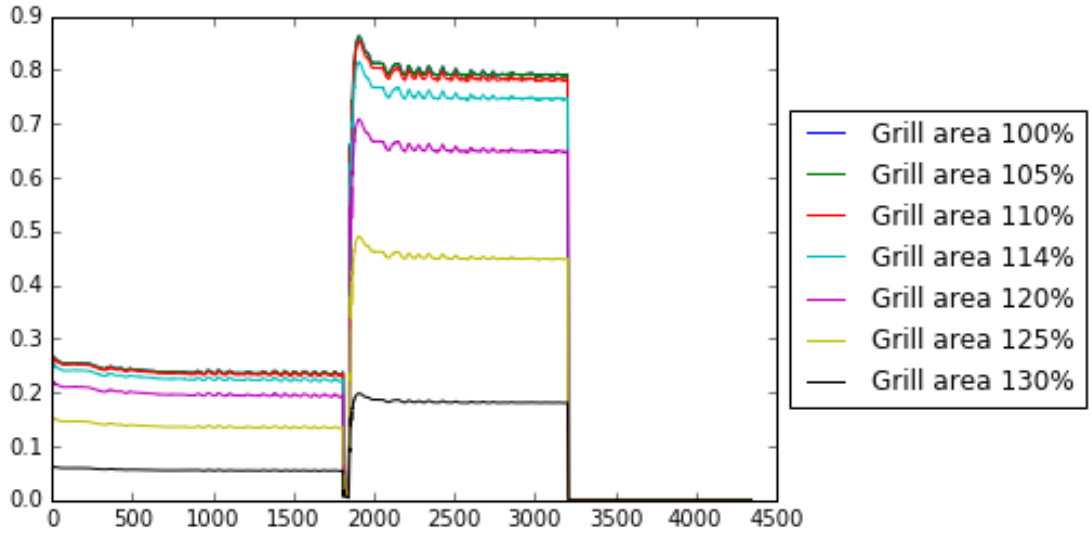


Figure 5.16 Predictions of KPI by RBFN when the area of the cars grill increases in small intervals

The plot shows that up until a change of 14%, the plot shows reasonable results but at 20% increase and above the results are becoming more and more unreliable due to the nature of extrapolation. One important factor to consider here is that when the grill area changes the airflow to the engine, some of the input parameters that are feed to the network would also change. Which means that the data in Figure 5.16 is reasonable but not necessarily accurate.

5.11 Ranking of parameters

In the concept phase, the automotive industry does not have endless means to handle every possible concept candidate. For that reason priorities are devised by means of educated guesses, system studies based on similar vehicles or concurrent analysis. This being a common practice in the industry, most of the time based on expert point of view [12].

When a ML framework is at place and working below strict error levels, this tool can be used further to rank the elements of the feature space in their importance based on changes they infer to the KPI. In this way, a Reduced Order Model (ROM) of the system at hand can be produced.

The CON-test was implemented on the RBFN and every parameter as tested but the results were inconclusive. Sung [11] presented two ways the importance of neither of them does reasonable fit into our results. The greatest changes in NRMSE are presented in Table 6.2 as percentage of the NRMSE for all the parameters.

Table 6.2

% of NRMSE	Parameter removed
77%	Tunnel dew-point temperature
86%	Coolant temperature before the radiator
100%	All parameters present
103%	Rolling drag
104%	Wind speed
106%	Dynamometer speed

Incorporating the first method of designating the largest changes as the parameters with the most impact on the output, the largest change came from the "Tunnel dew-point temperature", which is a reference temperature used to calculate the humidity in the tunnel. This does not fit in with what the most important parameters could be considered to be.

The alternative method to see the greatest occurring decreases in NRMSE as a precursor that the parameters were bad/unimportant for the data does not fit with the second parameter "Coolant temperature before the radiator"; seen by experts as a very important parameter. This is only a small subset of the problem with this test, but no reasonable results could be deduced.

6 Conclusions

Automotive industry has years of experience with recycling. However, the recycling of data is relatively new. This thesis aimed to give an insight on usage of tools from ML framework to achieve efficient reutilization of content in databases in Volvo Car Corporation. The focus of the present work was laid upon vehicle thermodynamic development databases.

With the use of open-source computer scripts, the relevant thermodynamic measurement data was extracted from VCC repositories and used to train ML tools, which could thereafter predict parameters involved in tuning the vehicle thermal attributes. In this training, different ANN network topologies where used. Two of the methods, RBFN and SVR with a RBF kernel, predicted the designated output parameter with a high degree of accuracy. By calculating an analytical solution for the weight-vector, the RBFN would complete a prediction at a fraction of the time it took the SVR to perform the same task. The SVR belonged to the Scikit-learn module of Python. Additionally, to be able to ameliorate predictions, SVM based approaches are tested while plugging in several types of linear/non-linear kernels. While a RBF kernel was required for the regression analysis, a much simpler linear kernel was sufficient for the classification of different cars and tests with SVC.

To help the concept leaders in choosing ML tools for development, elapsed CPU-time per ML method, additional to accuracy, was also reported in the M.Sc. thesis.

Using vehicle engine power as an output signal, an ANN based approach allowed for accurately predicting the behaviour of the system and permitted to test ML tools further, as these reached a max < 8% prediction error. Following reaching this literature-conform ability of deployed machine learning techniques to predict an arbitrary chosen key performance index, it was deemed that the effects of the internal combustion engine operation in the engine room could also be modelled likewise.

When the same type of wind tunnel tests from different vehicles are compared, a clear separation among the different cars could be seen in the 2 dimensional space of power and a temperature which serves as a reference for the temperature of the air in the engine compartment. This was also true for the same car with different types of test. Following the predictions using ANN, other tools for clustering both the experimental and ML based results are run. In this way, patterns depending on vehicle classes and wind tunnel test conditions are identified. As every step of the study amount of error introduced is constantly monitored, it is expected that these regions of vehicle clusters also constitute concept candidates which are not yet tested in the wind tunnel.

Additionally, methods for ranking the salient parameters affecting under hood thermal behaviour are presented. The CON-test was unable to find any reasonable relationship between a parameter and the NRMSE when that parameter was not taken into consideration. This indicates that this method is not yet suitable to be used to reduce the order of the model to shorten the time it needs to proceed. One reason for this might have been the coupling between the parameters, because both the humidity of the air and a reference temperature used to calculate the humidity was inserted as inputs to ANN so that the importance of humidity might have been overstated.

6.1 Future work

In a paper, Sung [11] compared three different methods which could be used to rank importance of input parameters of an artificial neural network. Of these ranking methods, the change of MSE appeared to be the most promising for the type and size of the problem Sung et al. investigated. This was not the case for this M.Sc. project and alternative methods should be explored.

The lack of accurate results from the feedforward neural network (FFNN) should not be viewed as a reason to dismiss the method, but as an indicator that more complex versions of the FFNN might be needed. Several hidden layers and a larger number of neurons in these layers might give more accurate results, but would require more time for the training phase than the radial basis function networks. Because the RBFN depends on the distance from centers in the space of the input parameter, RBFN could have greater problems when the distance from these points grow. If this makes it less useful for projections for new concepts needs to be examined.

All the input data used in this M.Sc. study came directly from the wind tunnel or was geometric information of the cars. The combination of crankshaft torque and engine rpm was only one of many physical calculations that could be done to reduce the order of the problem and possibly improve the accuracy of the prediction [18].

Classifications were performed in only 2 dimensional feature space. More extensive classification should be possible and may be used to determine relationships of more complex nature. Factor(s) common to the most popular automotive products could be identified and replicated in future concepts. Areas in the investigated feature space where VCC may profit introducing new products could be identified to fulfil customer demands.

Since it has been possible to predict vehicle thermodynamic attributes using ML based approaches, the method developed for this M.Sc. study is directly applicable to other engineering disciplines; for instance BEV battery thermal conditioning, aerodynamic analysis and model reduction, emission abatement and RDE, physics informed ML for turbulence modelling, NVH are areas that application of ML will have clear benefits.

7 References

[1] "Volvo Car Group Global Newsroom," Volvo Car Group, [Online]. Available: http://www.media.volvocars.com/global/en-gd. [Accessed June 2017].

[2] M. Törmänen, "Integrating Multi-Disciplinary Optimization into the Product Development Process," 05 12 2016. [Online]. Available: http://www.chalmers.se/SiteCollectionDocuments/Produkt-%20och%20produktionsutveckling/Nationell%20kompetensarena%20kring%20produktoptimering/MikaelTo%CC%88rma%CC%88nenIntegratingMulti-DisciplinaryOptimization.pdf.

[3] X. Ge and J. Jackson, "The Big Data Application Strategy for Cost Reduction in Automotive Industry," *SAE Int. J. Commer. Veh,* pp. 588-598, 2014.

[4] S. S. Haykin, Neural networks and learning machines (Vol. 3)., Upper Saddle River, NJ, USA::: Pearson., 2009.

[5] B. Mehlig, "Lecture notes," 18 10 2016. [Online]. Available: http://physics.gu.se~frtbm/joomla/index.ph.

[6] Fidler and S, "CSC 411: Lecture 01: Introduction," 11 January 2016. [Online]. Available: http://www.cs.utoronto.ca/~fidler/teaching/2015/slides/CSC411/01_intro.pdf. [Accessed 8 May 2017].

[7] A. Ng, "Coursera- Machine Learning," [Online]. Available: http://www.coursera.org/learn/machine-learning.

[8] Q. Zhang, A. Pennycott, R. Burke and S. Akehust, "Predicting the Nitrogen Oxides Emissions of a Diesel Engine using Neural Networks," *SAE TEchnoical Paper 2015-01-1626,* 2015.

[9] T. Bäck and C. Foussette, "Automatic Metamodelling of CAE Simulation Models," *ATX worldwide,* pp. 36-41, 2015.

[10] S. Aceves, D. Flowers, J.-Y. Chen and A. Babajimopoulos, "Fast Prediction of HCCI Combustion with an Artificial Neural Network Linked to a Fluid Mechanics Code," *SAE,* 2006-01-3298.

[11] A. H. Sung, "Ranking input importance in neural network modeling of engineering problems," *Neural Networks Proceedings,* pp. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on (Vol. 1, pp. 316-321). IEEE., 1998.

[12] M. Khaled, M. Ramadan, H. El-Hage, A. Elmarakbi, F. Harambat and H. Peerhossaini, "Review of underhood aerothermal management: Towards vehicle simplified models," *Applied Thermal Engineering,* vol. 73, no. 1, pp. 842-858, 2014.

[13] X. Zhu and A. B. Goldberg, Introduction to Semi-supervised learning, Morgan and Claypool Publisher, 2009.

[14] "scikit-learn: Machine Learning in Python," [Online]. Available: http://scikit-learn.org. [Accessed 2017].

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research,* pp. 2825-2830, 2011.

[16] VolvoCars, “Ny vindtunnel i världsklass ger Volvoköparen lägre CO2-utsläpp,” 25 September 2008. [Online]. Available: https://www.media.volvocars.com/se/sv-se/media/pressreleases/17008.

[17] C. Thiel, J. Schmidt, A. Van Zyl and E. Schmid, “Cost and well-to-wheel implications of the vehicle fleet CO 2 emission regulation in the European Union,” *Transportation Research Part A: policy and practice 63,* pp. 25-42, 2014.

[18] L. Graening and T. Ramsay, “Flow Field Data Mining Basesd on a Compact Streamline Representaion,” *SEA Technincal Paper,* 2015-01-1550.