



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

---



# **Control of Hydraulic Hybrid Wheel Loader Using Machine Learning**

A research project at Volvo CE

Master's thesis in Electrical Engineering

Manjunath Rajashekhar Batli  
Suraj Sureshabu Danaraddi

---

DEPARTMENT OF ELECTRICAL ENGINEERING  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Gothenburg, Sweden 2020



MASTER'S THESIS 2020

# Control of Hydraulic Hybrid Wheel Loader using Machine Learning

Manjunath Rajashekhar Batli  
Suraj Sureshababu Danaraddi



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering  
*Division of Systems and Control*  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Gothenburg, Sweden 2020

Control of Hydraulic Hybrid Wheel Loader using Machine Learning  
Master's Thesis in Electrical Engineering

**Manjunath Rajashekhar Batli**  
**Suraj Sureshababu Danaraddi**

© Manjunath Rajashekhar Batli, Suraj Sureshababu Danaraddi 2020.

Supervisor: Karl Uebel, Volvo Construction Equipment  
Examiner: Anders Grauers, Department of Electrical Engineering

Master's Thesis 2020  
Department of Electrical Engineering  
Division of Systems and Control  
Chalmers University of Technology  
SE-412 96 Gothenburg  
Telephone +46 31 772 1000

Cover: Image of Volvo Wheel Loader-170. Volvo Construction Equipment.

Chalmers Reproservice  
Gothenburg, Sweden 2020

Control of Hydraulic Hybrid Wheel Loader using Machine Learning  
Master's Thesis in Electrical Engineering  
Manjunath Rajashekhar Batli  
Suraj Sureshbabu Danaraddi  
Department of Electrical Engineering  
Division of Systems and Control  
Chalmers University of Technology

## **Abstract**

Control of Hydraulic Hybrid for wheel loader using Machine Learning is one of the possible ways to control Energy Management Strategies. In this thesis, The control of Supplementary Hydraulic Hybrid System was studied and project divided analysis of this control into two Phases. In Phase 1, Existing Rule Based Energy Management Strategy was replaced with Neural Networks. Investigations were carried out whether neural networks is able to replicate the Rule Based Energy Management Strategy. From the training accuracy of a neural network and simulation results, it was found that a Neural Network was able to predict States similar to Existing Rule Based Energy Management Strategy for most of the cases. In the Phase 2, Dynamic programming was used to simulate a Short Loading Cycles to obtain optimized results of hybrid torque request and pump pressure. The results of hybrid torque request and pump pressure were then trained using Neural Network Regression and implemented online to predict hybrid torque request and pump pressure. The neural network regression models was successful in predicting the optimised hybrid torque and pump pressure. The results of Phase 1 demonstrated that a neural network can replace the rule based controller when deciding which hybrid state the hybrid drive line should operate on and in Phase 2, it was shown that a neural network can determine the torque request and pump pressure accurately and comparing with results from dynamic programming, which find the optimal control, it was possible to show that the neural network was very close to the optimal control. This approach showed the potential of improving the fuel efficiency.

Keywords: Hydraulic Hybrid, Machine Learning, Neural Networks, Dynamic Programming, Rule Based Energy Management System, Wheel Loader, Short Loading Cycles, Control System.



## Acknowledgements

This thesis represents a final part of our master's program at Chalmers University of Technology. We wish to express our greatest gratitude to Volvo Construction Equipment for giving us opportunity to carry out research project.

We whole-heartedly thank our Volvo Construction Equipment supervisor Karl Uebel for your invaluable assistance that you provided during our study. We would also like to thank Henrique Raduenz for your continuous encouragement and willingness to assist in any possible way throughout the project.

Further, We would like to thank our University supervisor Prof. Anders Grauers for your consistent guidance and support path during our thesis.

Finally, We would like to pay special regards to our parents and friends for their unconditional support throughout our master's program.

*Gothenburg, September 2020*

Manjunath Rajashekhar Batli

Suraj Sureshababu Danaraddi



# Contents

<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xiii</b>
<b>Abbreviations</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Problem Definition . . . . .	1
1.3 Scope . . . . .	2
1.4 Method . . . . .	2
1.5 Thesis Outline . . . . .	3
<b>2 Theory</b>	<b>5</b>
2.1 Hydraulic Hybrid Vehicles . . . . .	5
2.1.1 Series Hybrid . . . . .	5
2.1.2 Parallel Hybrids . . . . .	6
2.1.3 Combined Hybrids . . . . .	7
2.2 Hydraulic hybrid wheel loader Concept . . . . .	8
2.3 Driving Cycles . . . . .	10
2.3.1 Short Loading Cycle . . . . .	10
2.3.2 Load and Carry Cycle . . . . .	11
2.4 Machine Learning . . . . .	12
2.4.1 Supervised learning . . . . .	12
2.4.1.1 Linear regression . . . . .	12
2.4.1.2 Logistic regression . . . . .	13
2.4.1.3 Artificial neural network . . . . .	15
2.4.1.4 Support Vector Machines . . . . .	17
2.4.2 Unsupervised learning . . . . .	19
2.5 Neural Network Regression and Deep Learning Regression . . . . .	20
2.6 Dynamic Programming Method . . . . .	22
2.7 Overview of the system model . . . . .	22
2.7.1 Backward-facing simulation model . . . . .	23
2.7.2 Wheel loader simulation model overview . . . . .	23
2.7.3 Control System . . . . .	24
2.7.4 Energy Management Strategies . . . . .	25
2.7.5 Torque control system . . . . .	26

<b>3</b>	<b>Energy Management System</b>	<b>27</b>
3.1	Phase 1: Replicate a RB-EMS with a Neural Network . . . . .	27
3.1.1	Neural Network Training . . . . .	27
3.1.2	Implementation of Neural Network in Simulink . . . . .	30
3.2	Phase 2: Optimization of hybrid torque and Machine 2 pressure. . . . .	30
3.2.1	Optimization using Dynamic Programming . . . . .	30
3.2.2	Training of Hybrid torque request . . . . .	31
3.2.2.1	Obtaining Data . . . . .	31
3.2.2.2	Linear regression . . . . .	31
3.2.2.3	Neural network linear regression . . . . .	32
3.2.3	Training of the Machine 2 pressure . . . . .	35
3.2.3.1	Obtaining training data . . . . .	35
3.2.3.2	Linear regression . . . . .	35
3.2.3.3	Neural network regression . . . . .	35
<b>4</b>	<b>Results and Discussion</b>	<b>37</b>
4.1	Phase 1 . . . . .	37
4.2	Phase 2 . . . . .	41
4.2.1	Dynamic Programming results . . . . .	41
4.2.2	Comparison of accuracy of Linear Regression and Neural Network Regression . . . . .	42
4.2.2.1	Training on prediction of hybrid torque request . . . . .	42
4.2.2.2	Testing on prediction of hybrid torque request . . . . .	44
4.2.2.3	Training on prediction of Machine 2 pressure . . . . .	45
4.2.2.4	Testing on prediction of Machine 2 pressure . . . . .	47
4.2.3	Results of neural network regression implemented on Simulink model . . . . .	49
<b>5</b>	<b>Conclusion</b>	<b>53</b>
	<b>Bibliography</b>	<b>55</b>

# List of Figures

2.1	Schematic illustration of the SHHV. Picture source - [7]	6
2.2	Schematic illustration of the PHHV. Picture source - [7]	6
2.3	Schematic illustration of the PSHHV. Picture source - [7]	7
2.4	Schematic illustration of the conventional wheel loader driveline, Picture source- [4]	8
2.5	Hydraulic wheel loader driveline, Picture source- [4].	9
2.6	SLC, Picture inspired by [10].	10
2.7	Velocity profile of SLC, Picture source- [8].	11
2.8	Velocity profile of LCC, Picture source-[8].	11
2.9	Uni-variate linear regression	13
2.10	Sigmoid function	14
2.11	One vs all Multi-class classification. Picture Source [18]	15
2.12	Neuron, Picture Source:[20]	16
2.13	Artificial Neural Network	16
2.14	SVM Stage 1, Picture Source:[22]	18
2.15	SVM Stage 2, Picture Source:[22]	18
2.16	Flow chart of K-means algorithm, Picture Source:[24]	20
2.17	Neural network forward propagation	21
2.18	Wheel loader simulation model. Picture source Wheel loader: [29]	23
2.19	Backward-facing simulation model of the wheel loader modelled in Simulink	24
2.20	Method of calculation of blocks modeled in Simulink. Picture source Wheel loader: [4]	24
2.21	RB-EMS state flow architecture. Picture source: [4]	26
3.1	Schematic Representation of Implementation	27
3.2	ML implementation on simulink	30
3.3	Neural network regression implementation in Simulink	33
3.4	Neural network regression Simulink block	34
3.5	Forward propagation inside Layer 1	34
3.6	Forward propagation inside Layer 2	34
3.7	Machine 2 pressure neural network implementation in simulink	36
4.1	Confusion matrix of ML Training Accuracy	38
4.2	Confusion matrix of ML in Simulink for training data	39
4.3	Confusion matrix of ML in Simulink for cross validation data	39
4.4	Confusion matrix of ML in Simulink for test data	40

4.5	Comparison of ML Predicted states with RB-EMS predicted states . . . . .	40
4.6	Comparison of Hybrid torque, SOC, Velocity obtained from optimized DP . . . . .	41
4.7	Comparison of Machine 2 pressure (pump pressure), SOC, Velocity obtained from optimized DP . . . . .	42
4.8	Training accuracy of the predicted hybrid torque values using linear regression model . . . . .	43
4.9	Training accuracy of the predicted hybrid torque values using neural network regression model . . . . .	43
4.10	Test accuracy of the predicted hybrid torque values using linear regression model . . . . .	44
4.11	Test accuracy of the predicted hybrid torque values using neural network regression model . . . . .	45
4.12	Training accuracy of the predicted Machine 2 pressure (pump pressure) values using linear regression model . . . . .	46
4.13	Training accuracy of the predicted Machine 2 pressure (pump pressure) values using neural network regression model . . . . .	46
4.14	Training accuracy of the predicted Machine 2 pressure (pump pressure) values using linear regression model . . . . .	47
4.15	Training accuracy of the predicted Machine 2 pressure (pump pressure) values using NN regression model . . . . .	47
4.16	Comparison of Hybrid torque, SOC, Velocity obtained by integrating ML model in Simulink model . . . . .	49
4.17	Comparison of Machine 2 pressure (pump pressure), SOC, Velocity obtained by integrating ML model in Simulink model . . . . .	50
4.18	Comparison of normalised fuel consumption between simulation model with RB-EMS and simulation model with online ML. . . . .	50

# List of Tables

2.1	Strategy states. Source [4]	25
-----	-----------------------------	----



# Abbreviations

ANN	Artificial Neural Network.
DL	Deep Learning.
DP	Dynamic Programming.
EMS	Energy Management System.
ICE	Internal Combustion Engine.
LCC	Load and Carry Cycle.
ML	Machine Learning.
NN	Neural Network.
PHHV	Parallel Hydraulic Hybrid Vehicle.
PSHHV	Power-Split Hydraulic Hybrid Vehicle.
PTO	Power Take-Off.
RB-EMS	Rule-Based Energy Management Strategy.
SHHS	Supplementary Hydraulic Hybrid System.
SHHV	Series Hydraulic Hybrid Vehicle.
SLC	Short Loading Cycle.
SOC	State of Charge.
SVM	Support Vector Machines.
TC	Torque Converter.



# 1

## Introduction

### 1.1 Background

In recent years, sustainable and fuel-efficient transportation led to the evolution of alternative fuels and energy efficient vehicles. The carbon free and renewable potential of electricity has led to electrification and electric hybridization of cars. In the field of construction machineries, hybrid technology is considered an important candidate to address the issues of high emissions and improve fuel efficiency [1]. The construction machineries often operate in short repetitive cycles which involves a lot of starts and stops. Hydraulic hybrid technology, which can deliver high specific power output can be favorable to use in construction machinery [2] [3]. A previous study has shown the potential of combined hydraulic hybrid concept in a wheel loader to improve the fuel efficiency [4].

Most of the construction machineries use automatic transmission to transfer engine power to the wheels. Torque Converter (TC) is used in wheel loader for automatic transmission. Usually, TC have low transmission efficiency at higher torques. If the dependency on the TC is reduced during its low-efficiency phase then, a significant amount of fuel can be saved by using a hydraulic hybrid. Further, the fuel efficiency can be improved by effective utilization of the hybrid system. An energy management strategy is required to efficiently utilize the energy from the Internal Combustion Engine (ICE) and the hydraulic system. In the previous work [4] a rule based strategy was developed and led the foundation for potential energy savings.

### 1.2 Problem Definition

A rule-based strategy proposed for the system in [4] is quite complex and is not able to deliver the expected fuel efficiency for different driving cycle. Based on the previously carried out preliminary results [5], it has been found that Dynamic Programming (DP) and online machine learning (ML) strategies can produce better fuel efficiency for different driving cycle, while maintaining an acceptable operability in the wheel loader applications.

### 1.3 Scope

The project is divided into two phases. The first phase of the project aims to replicate the existing Rule Based Energy Management Strategy (RB-EMS) [4] using a control strategy based on ML. The ML strategy should predict the same states as the RB-EMS, therefore learning the transition rules and further leading to similar fuel consumption as that of the previously used RB-EMS. In the second phase of the project, the focus is on improving the fuel efficiency by optimizing the hybrid system control with DP and implementing with ML technique. The following questions will be investigated and answered:

- **Q1:** Will the ML strategy be able to learn and replace the functionality of RB-EMS ?
- **Q2:** Which ML strategies are efficient and are used to replace different control systems?
- **Q3:** Does implementation of ML technique, using optimized results of DP, have the potential of improving fuel efficiency?
- **Q4:** What are the challenges involved in the implementation of online ML in the existing energy management system?

### 1.4 Method

The methods used in carrying out the thesis are listed below.

- Literature survey and pre-study.
- Training of different ML strategies.
- Modeling.
- Simulation.

To fulfill the previously discussed scope and to generate the necessary background, the project will begin with a literature study on the analysis of previously carried out thesis work and conceptual analysis of the Simulink model provided. Different ML strategies is studied and the suitable ML models is trained and implemented in phase 1 and phase 2 using Short Loading Driving Cycles (SLC). DP is used to obtain the optimal solutions and ML model is used to train the control sequence of DP in phase 2. The trained ML models are implemented on the Simulink model in Phase 1 and Phase 2. Finally, the results obtained by simulating the model with new control strategy was evaluated and the performance of the hybrid system was analyzed.

## 1.5 Thesis Outline

The remaining of this thesis report consists of following chapters.

**Chapter 2:** Relevant theory about different types of Hydraulic hybrid vehicles are presented. It also explains about hydraulic hybrid wheel loader concept, DP, different driving cycles and ML techniques. Finally, Overview of system model are presented in this chapter.

**Chapter 3:** In this chapter, two phases of implementation in EMS are presented. Phase 1 explains about implementation of neural network (NN) to replicate existing rule based strategy states. Phase 2 explains about implementation of ML technique to improve efficiency with the help of DP.

**Chapter 4:** In this chapter, Results of phase 1 and phase 2 are presented which includes training accuracy, ML techniques implemented in RB-EMS of a Simulink model.

**Chapter 5:** This chapter includes conclusion from the thesis work together with recommendation about the future work.



# 2

## Theory

This section explains about relevant theory of wheel loaders and ML techniques.

### 2.1 Hydraulic Hybrid Vehicles

The hybrid vehicles have a secondary energy carrier attached to the conventional driveline system, which often consists of fossil fuel as primary source of energy and ICE to generate energy. The primary energy carrier often have high energy density and larger energy storage capacity. However, secondary energy carrier has short amount of energy storage capacity and are usually electricity or hydraulic energy. This secondary energy carriers recuperate and stores the energy during situations such as regenerative braking and hence are very effective to improve fuel efficiency of the vehicle [6]. Different configurations of hydraulic hybrid vehicles have their own significance and are explained in the subsections.

#### 2.1.1 Series Hybrid

In series hybrid vehicle, the secondary energy generator is often connected to the engine shaft. Similarly, in Series Hydraulic Hybrid Vehicle (SHHV) a hydraulic pump is connected to the engine shaft and a clutch is placed between the engine and the pump in order to enable decoupling of the engine. An accumulator is connected to the pump to store the hydraulic energy [7]. Series hybrid is a traditional categorization used for transportation vehicles, whereas there are other configurations for working machines that are more difficult to put in these categories. Illustration of the SHHV configuration can be seen in Figure 2.1.

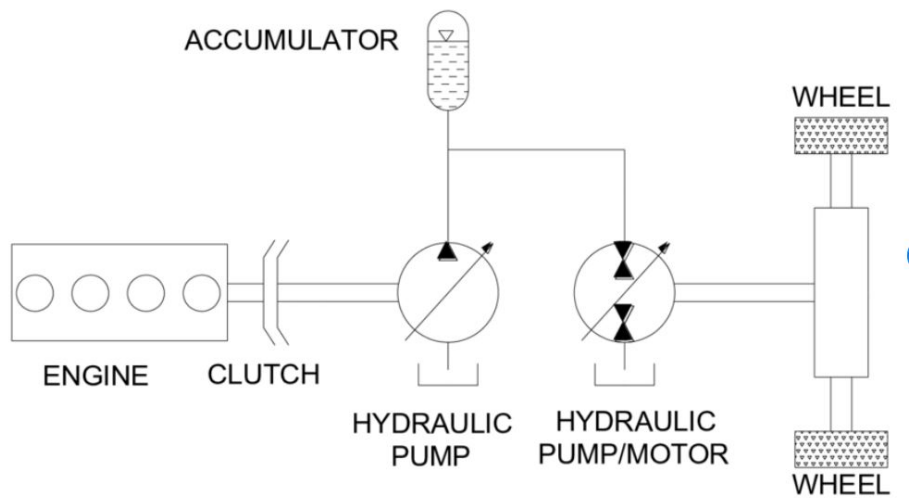


Figure 2.1: Schematic illustration of the SHHV. Picture source - [7]

### 2.1.2 Parallel Hybrids

Figure 2.2 illustrates the Parallel Hydraulic Hybrid Vehicle (PHHV). PHHV has a conventional drive train, which consists of an engine directly connected to the transmission. A hydraulic machine is connected to the drive shaft in-between the engine and the transmission. The hydraulic machine is connected to an accumulator to store the hydraulic energy. There is a clutch placed in-between the engine and the hydraulic machine, which enable complete decoupling of the engine from the road load. The power is supplied completely by the hydraulic system during the decoupling phase and whenever the pressure/charge in the accumulator is very low, the engine is coupled again to recharge the accumulator [7].

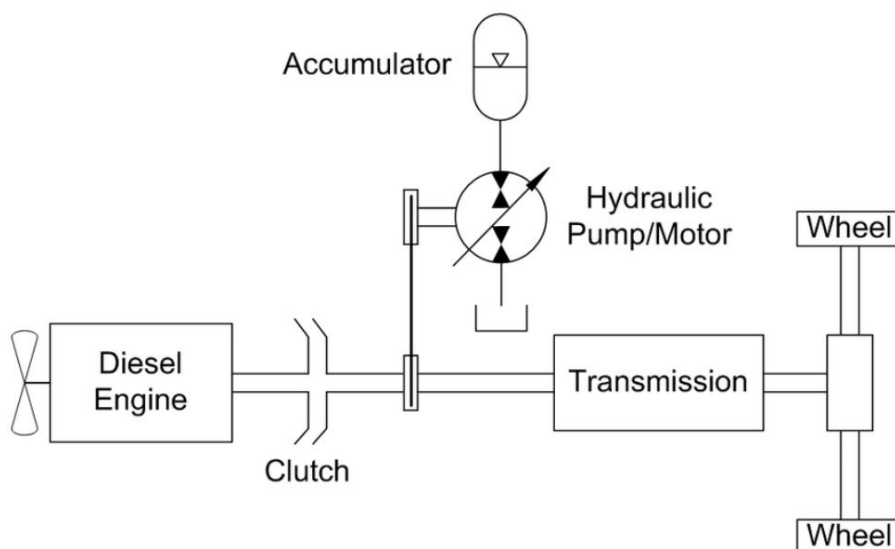


Figure 2.2: Schematic illustration of the PHHV. Picture source - [7]

### 2.1.3 Combined Hybrids

Figure 2.3 illustrates the Power-Split Hydraulic Hybrid Vehicle (PSHHV). It has the architecture of both series and parallel hybrid configuration. The hydraulic machine is connected to the drive shaft as in SHHV and the drive train is conventional as in PHHV. PSHHV have the advantages of both series and parallel configurations. Series configuration always allows the engine to run at optimal speed in addition, helps to reduce the fuel consumption. Parallel configuration allows the vehicle to consume energy parallel from both engine and hydraulic system whenever the torque demand is high, while keeping engine at optimal speed and eventually improving the fuel efficiency. Hence, the power split hybrids have advantage of both series hybrid and parallel hybrid configurations [7].

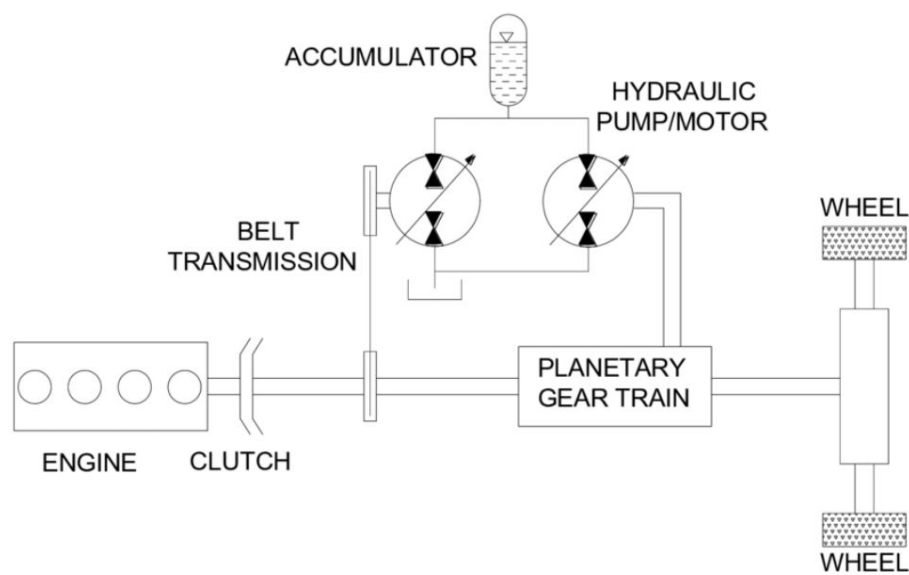


Figure 2.3: Schematic illustration of the PSHHV. Picture source - [7]

## 2.2 Hydraulic hybrid wheel loader Concept

A wheel loader is a construction machinery used to perform different operations such as load loose materials with the bucket mounted at front and to move the gravel at the construction site. Often the size of the wheel loader is determined by the load it carries and the type of operations it performs. Figure 2.4 represents the driveline system of a conventional wheel loader. The main components such as ICE, transmission system, TC, Pump/motor, Power Take-Off (PTO), cylinders, work hydraulic control valves and Final Drive (FD) are also illustrated [4] [8].

Wheel loaders uses a TC instead of a clutch. It is an important component in the driveline system for main reasons, torque amplification and mechanical decoupling of the ICE from the wheels [4]. This is how a TC replaces the work done by clutch, in a typical wheel loader [9].

The PTO transfers power from the engine to the main hydraulic pump. The hydraulic pump is mainly used to supply hydraulic power to the cylinders that control the functions such as boom lifting/lowering, bucket tilting and steering of the machine. These functions are controlled by the main control valves and these valves are called as Work Hydraulics Control Valve in Figure 2.4 [6] [4] [8].

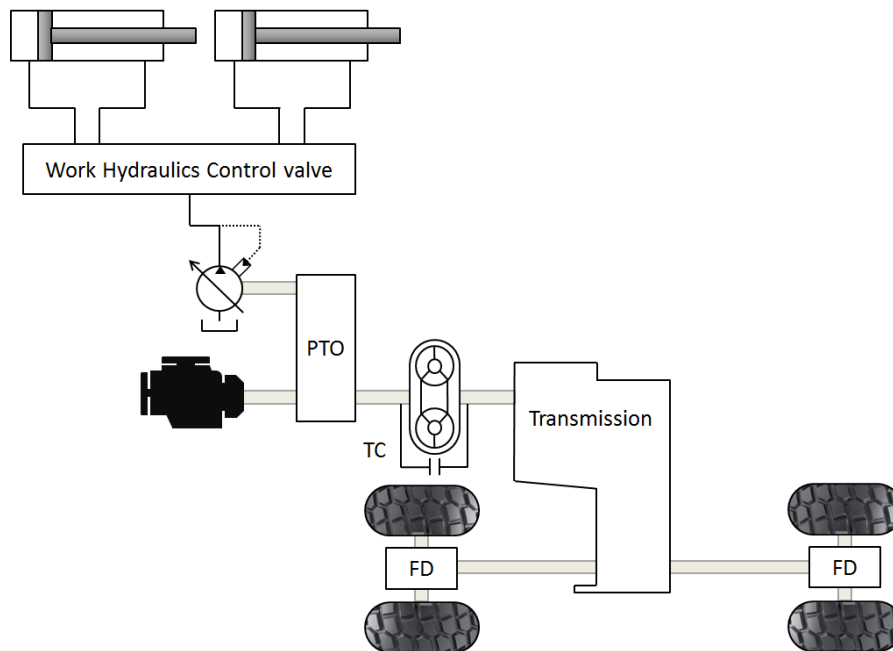


Figure 2.4: Schematic illustration of the conventional wheel loader driveline, Picture source- [4]

The hydraulic hybrid system investigated during this thesis work is as shown in Figure 2.5. A Supplementary Hydraulic Hybrid System (SHHS) is integrated into a conventional wheel loader shown in Figure 2.4. It can be seen that the hydraulic machine is connected to the driveline at two locations, one to the PTO at the ICE

and one to the transmission. These two connections enable both series and parallel hybrid operations. Rule-based strategy is used to control the operations of the valves, and this effect has reduced a significant amount of fuel consumption [4] [9].

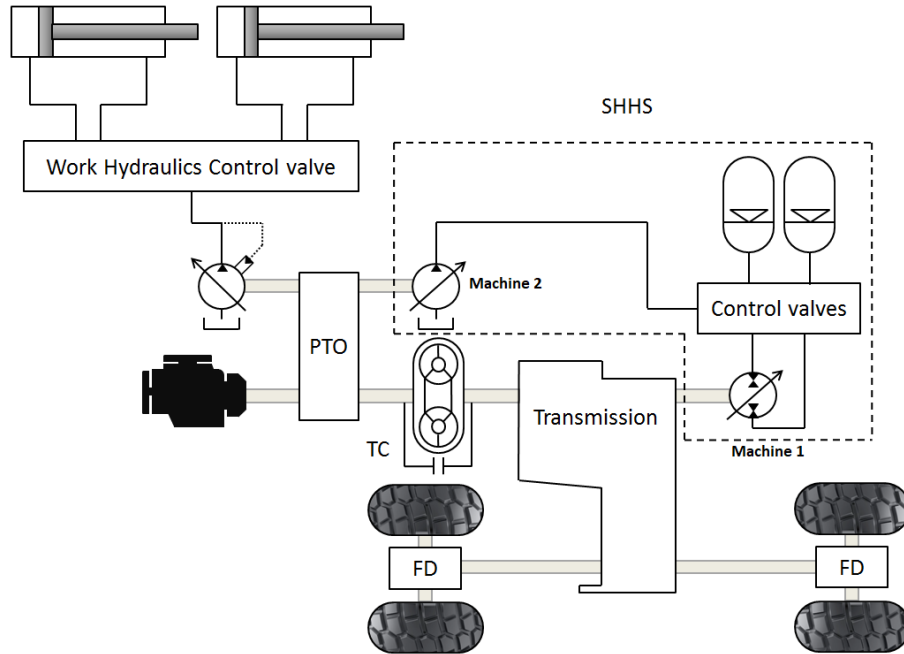


Figure 2.5: Hydraulic wheel loader driveline, Picture source- [4].

## 2.3 Driving Cycles

Wheel loader often operate in a short and repetitive cycle. The most often performed driving characteristics of a wheel loader can be generalized and represented in the form of SLC and Load and Carry Cycle (LCC). These SLC and LCC will be studied and used to train the ML model.

### 2.3.1 Short Loading Cycle

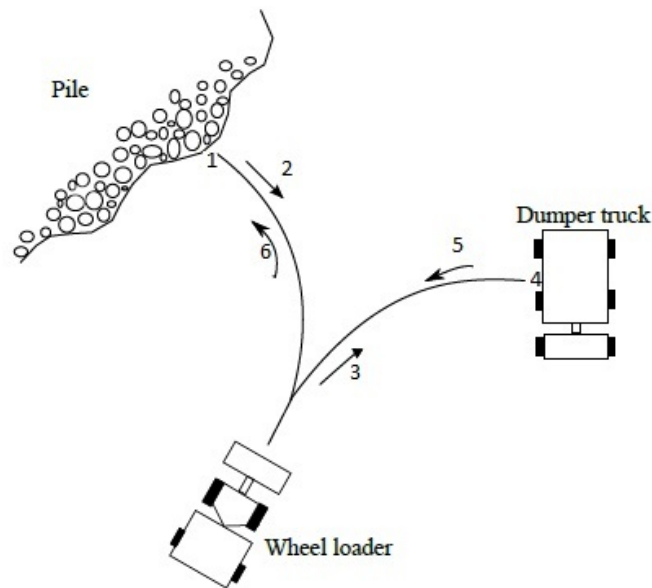


Figure 2.6: SLC, Picture inspired by [10].

Picture description of a typical SLC can be seen in Figure 2.6 and the velocity profile of a SLC performed by wheel loader can be seen in Figure 2.7. The driving pattern of SLC is explained as follows:

- **bucket-fill (step 1):** The operator moves from the starting point and fills the bucket.
- **Reversing from gravel pile (step 2):** The operator shifts to reverse and moves away from the gravel pile.
- **Approaching hauler (step 3):** Once the vehicle comes in the direction of hauler, the operator applies brake and shifts to move forward.
- **Unloading the bucket (step 4):** While in the process of approaching the hauler, the operator lifts the bucket and gets ready to unload.
- **Reversing from the hauler (step 5):** The operator shifts to reverse and moves away from the hauler.
- **Approaching gravel pile (step 6):** The operator now brakes and shifts forward. Now the cycle gets complete and the operator gets ready to repeat the cycle.

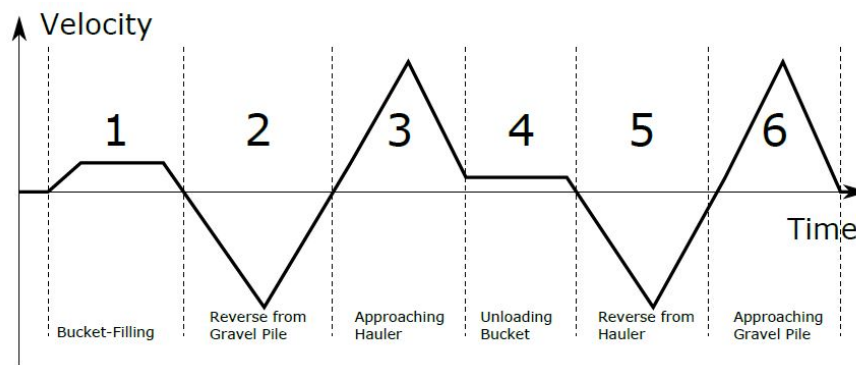


Figure 2.7: Velocity profile of SLC, Picture source- [8].

### 2.3.2 Load and Carry Cycle

The LCC is similar to SLC but the transportation phase is long which involves carrying the load in the bucket from the gravel pile to the hauler and returning back from the hauler and approaching the gravel pile. The typical velocity profile for the LCC is represented in Figure 2.8.

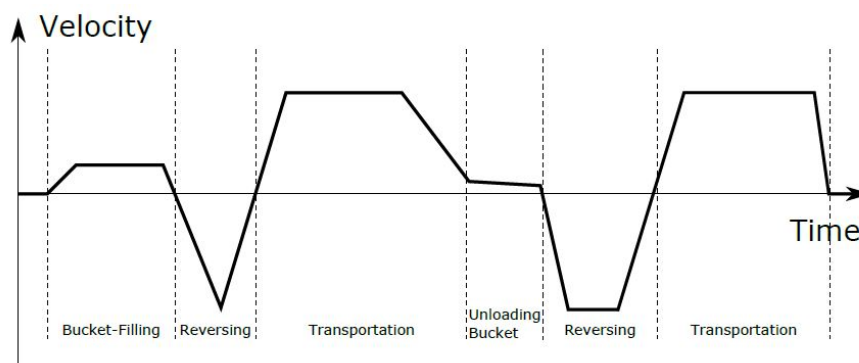


Figure 2.8: Velocity profile of LCC, Picture source-[8].

## 2.4 Machine Learning

ML is a branch of artificial intelligence that allows computer systems to learn directly from examples, data, and experience [11]. Through enabling computers to perform specific tasks intelligently, ML systems can carry out complex processes by learning from data, rather than following pre-programmed rules [11].

There are many ML algorithms that are classified under the following approaches.

1. Supervised learning
2. Unsupervised learning
3. Reinforcement learning
4. Recommender systems

### 2.4.1 Supervised learning

Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events [12]. The system is able to provide targets for any new input after sufficient training [12]. The algorithm can also minimize the error by reducing the differences between actual and intended output. The data are divided into training, cross validation and test set. Training data are used to train the model and learn the output pattern. Cross validation data are used to check if model is biased, variance or rightly fitted. There is always a trade off between biased and variance. A good learning algorithm would discover various heuristics and rules that apply to task at hand, such a system is said to be unbiased [13]. However, achieving low bias results in high variance. To overcome this, model are built in sufficiently versatile manner to reasonably approximate a broad range of input/output mappings is necessarily sensitive to the idiosyncrasies of particular data used for its training and therefore require large training set [13]. Best performance is achieved when model has minimal bias and variance term.

Some of commonly used types of supervised learning are as follows.

1. Linear regression
2. Logistic regression
3. Artificial neural network
4. Support vector machines

#### 2.4.1.1 Linear regression

Linear regression using one single independent variable is called uni-variate regression while analysis using more than one independent variable is called multivariate regression analysis [14]. To explain further about linear regression, consider a small example of uni-variate regression.

In the Figure 2.9 X is the input with number of experience in years and y is the output with salary of a person in SEK. With regression technique, the linear relationship between salary of a person and work experience is found out. The regression line shows the best fit of the model.

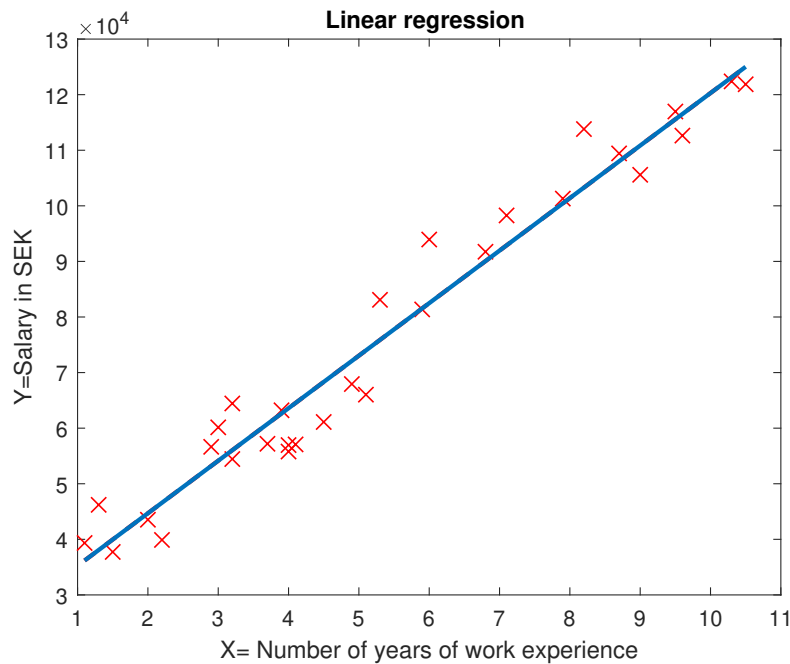


Figure 2.9: Uni-variate linear regression

The regression line is found by using the following hypothesis equation.

$$h_{\theta}(x) = \theta_0 + \theta_1(x) \quad (2.1)$$

Here,

x=training data (number of experience in years)

y=output data (supervised learning)

The hypothesis  $h_{\theta}(x)$  maps from input  $x$  to estimate the value of  $y$  by finding best values of  $\theta_0$ ,  $\theta_1$ . Ideally  $\theta_0$  and  $\theta_1$  is selected so that  $h_{\theta}(x)$  is close to  $y$  for the given training examples( $x$ ,  $y$ ). Theta's are calculated using following cost function equation.

$$J(\theta_0, \theta_1) = \frac{1}{2m} * \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 \quad (2.2)$$

The cost function equation (2.2) is minimized to find best possible  $\theta_0$ ,  $\theta_1$  using gradient descent or normal equation. Once best  $\theta_0$ ,  $\theta_1$  are found out, they are inserted in equation (2.1) to get best fit regression line for the model.

Similarly, linear regression for multiple variable/inputs can be solved using the following hypothesis equation.

$$h_{\theta}(x) = (\theta^T * x) = \theta_0 * x_0 + \theta_1 * x_1 + \theta_2 * x_2 + \theta_3 * x_3 + \dots + \theta_n * x_n \quad (2.3)$$

#### 2.4.1.2 Logistic regression

Logistic regression is popular probabilistic based statistical model used to solve the classification problems [15]. Generally logistic regression is classified as negative class (0) and positive class (1) that is the output  $y$  is either predicted 1 or 0. Logistic regression estimates the probabilities using a logistic function, which is also referred

## 2. Theory

---

to as sigmoid function [15].

To predict negative class and positive class, the following hypothesis equation is used .

$$h_{\theta}(x) = g * (\theta^T * x) = g(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{-\theta^T x}} \quad (2.4)$$

where,

$g = \frac{1}{1 + e^{-z}}$  is the sigmoid function.  
 $z = \theta^T x$ .

A **sigmoid function** is a mathematical function having a characteristic "S"-shaped curve or sigmoid curve [16] and is shown in Figure 2.10

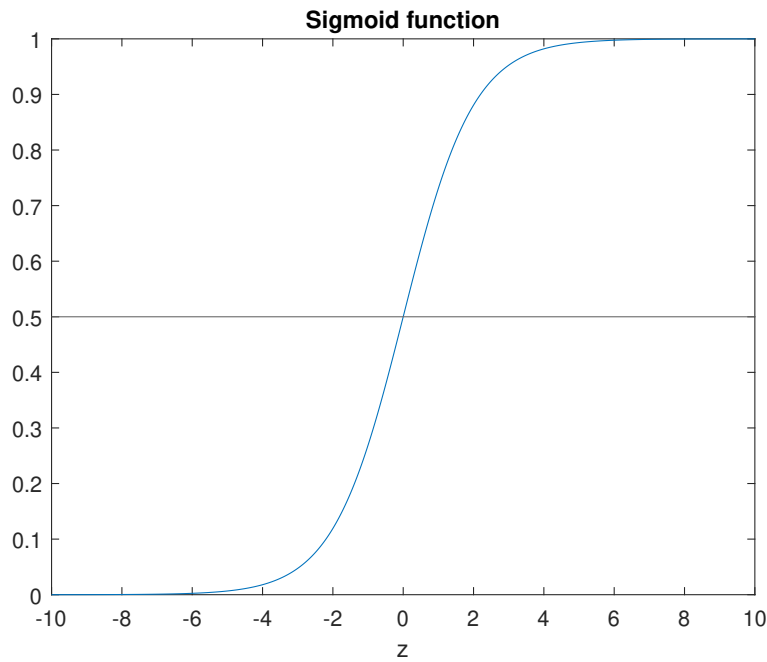


Figure 2.10: Sigmoid function

There is threshold classifier for hypothesis equation at 0.5.

If  $h_{\theta}(x) \geq 0.5$ , predict  $y=1$  which reflects to  $g(z) \geq 0.5$  when  $z \geq 0$ .

If  $h_{\theta}(x) < 0.5$ , predict  $y=0$  which reflects to  $g(z) < 0.5$  when  $z < 0$ .

The cost function of logistic function is different from linear regression. Logistic function produces wavy output, causing many local optima. In other words, it will not be a convex function [17]. Therefore, the following cost function will be used to obtain theta.

$$J(\theta) = -\frac{1}{m} \left[ \sum_{i=1}^m y^{(i)} \log h_{\theta}(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)})) \right] \quad (2.5)$$

Note that,  $y$  is always either 0 or 1.

Therefore,

If  $y=1$ ,  $J(\theta) = -\log(h_{\theta}(x))$

If  $y=0$ ,  $J(\theta) = -\log(1 - h_{\theta}(x))$

Cost function is minimized using optimizing techniques like gradient descent, conjugate gradient, BFGS, L-BFGS.

**Multi-class classification logistic regression:** It is the type of classification method where there are more than two possible outcomes. One of the common example is to predict weather to be either sunny/cloudy/rain/snow. In case of binary logistic regression, the output  $y$  is always either 0 or 1 whereas in Multi-class classification logistic regression the output  $y$  for weather prediction example would be 1,2,3,4 respectively for each class.

**One vs all Multi-class classification:** It is one of widely used multi-class classification. The Figure 2.11 shows the example of one vs all multi-class classification with 3 classes.

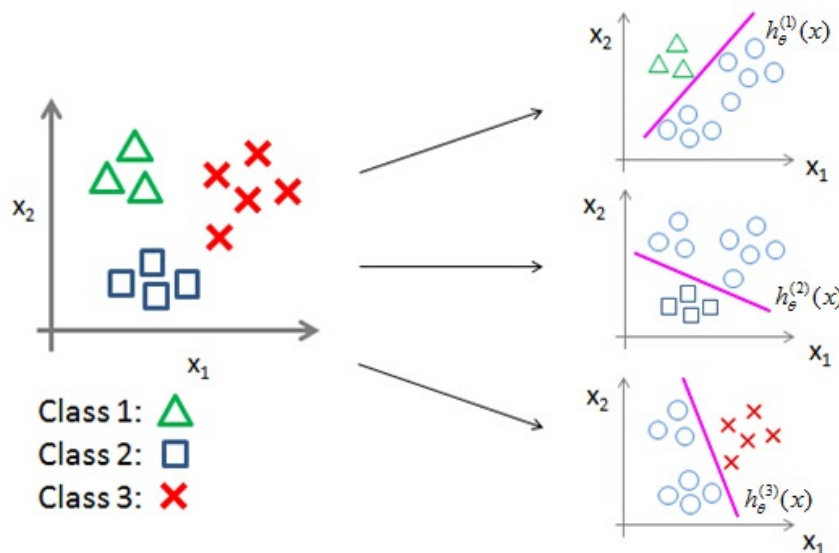


Figure 2.11: One vs all Multi-class classification. Picture Source [18]

Basically, Figure 2.11 shows that it selects one class and then lumps the other two class into a single second class. This is done repeatedly by applying binary logistic regression to each case, and then use the hypothesis that returned the highest value as prediction [17].

### 2.4.1.3 Artificial neural network

A NN is an interconnected assembly of simple processing elements, units or nodes, whose functionality is loosely based on the animal neuron [19]. The processing ability of the network is stored in the interunit connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns [19].

Artificial neural network (ANN) works on the same principle as that of human brain. To understand it better, Refer to neuron of human brain in Figure 2.12.

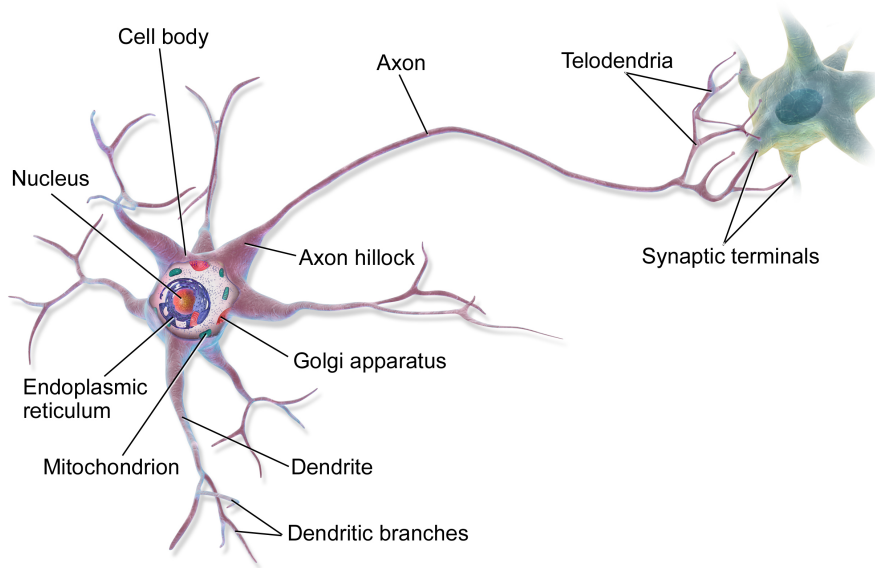


Figure 2.12: Neuron, Picture Source:[20]

In Figure 2.12, neurons are basically computational units that take inputs (dendrites) as electrical inputs (called "spikes") that are channeled to outputs (axons) [17]. Similarly, In ANN  $x_1, x_2, x_3, \dots, x_n$  are input features act as dendrites and output of the hypothesis act as axon.

In ANN, there is a input node  $x_0$  called as bias unit that always produces constant value 1 or other constant. In ANN, activation functions are preferred to improve convergence speed . Theta's are considered as weights. Input features are known as first layer, output of hypothesis are known as output layer and intermediate layers between first and output layers are known as hidden layers. The hidden layers are represented by  $a_o^2, \dots, a_n^2$  called as activation units.

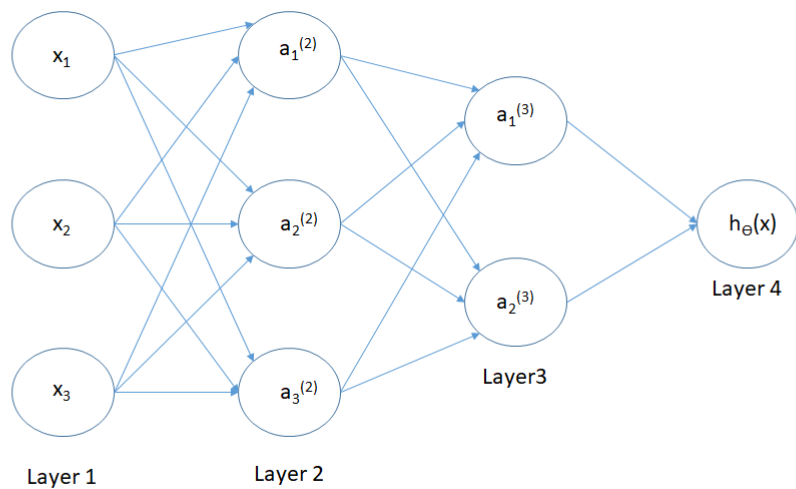


Figure 2.13: Artificial Neural Network

Figure 2.13 shows an example of ANN. It consists of 4 layers, Layer 1 and layer 4 are known as Input and output layer respectively where as layer 2, layer 3 are hidden layers of the network. The NN is calculated as follows.

$$a_1^{(2)} = g(\theta_{10}^{(1)}x_0 + \theta_{11}^{(1)}x_1 + \theta_{12}^{(1)}x_2 + \theta_{13}^{(1)}x_3) \quad (2.6)$$

$$a_2^{(2)} = g(\theta_{20}^{(1)}x_0 + \theta_{21}^{(1)}x_1 + \theta_{22}^{(1)}x_2 + \theta_{23}^{(1)}x_3) \quad (2.7)$$

$$a_3^{(2)} = g(\theta_{30}^{(1)}x_0 + \theta_{31}^{(1)}x_1 + \theta_{32}^{(1)}x_2 + \theta_{33}^{(1)}x_3) \quad (2.8)$$

The equations 2.6, 2.7, 2.8 are used to calculate the activation nodes for layer 2. Similarly, activation nodes are calculated for layer 3 and output of NN is calculated as follows:

$$h_{\theta}(x) = a_1^{(3)} + a_2^{(3)} \quad (2.9)$$

In general,

$a_i^{(j)}$ ="activation" of unit i in layer j [17]

$\theta^{(j)}$ =matrix of weights controlling function mapping from layer j to layer j+1[17]

#### 2.4.1.4 Support Vector Machines

Support Vector Machines (SVM) is a alternative approach to logistic regression which is based on statistical learning theory by Vapnik et al. SVM is mainly used for classification type of problems.

It is based on the principle of structural risk minimization, rather than the traditional empirical risk minimization principle [21]. Figure 2.14, 2.15 shows a simple example of SVM. There are two different classes in Figure 2.14 segregated in three different ways. Finding the best hyperplane is very important in SVM. In this example, hyperplane B is performed excellently in segregating the two classes. For further verification, SVM uses margin method which is shown in Figure 2.15. This method is used to find maximised distance between nearest point and hyperplane.

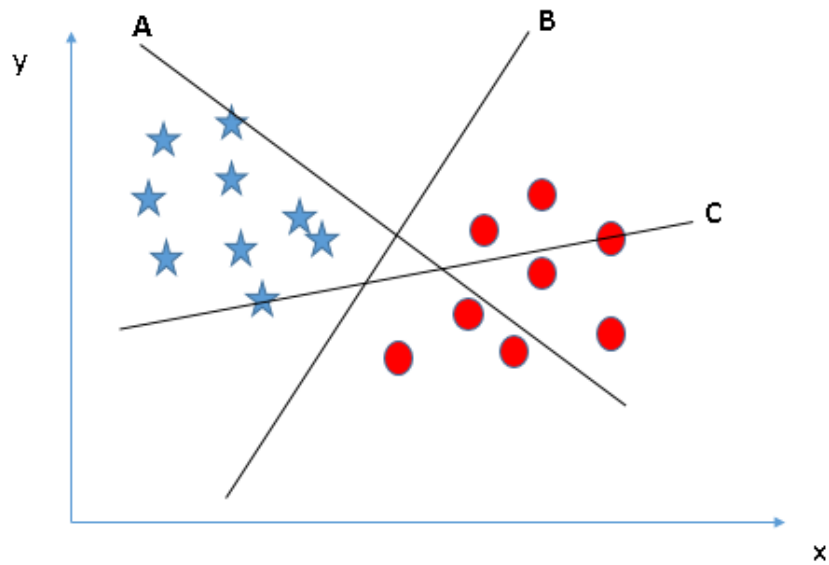


Figure 2.14: SVM Stage 1, Picture Source:[22]

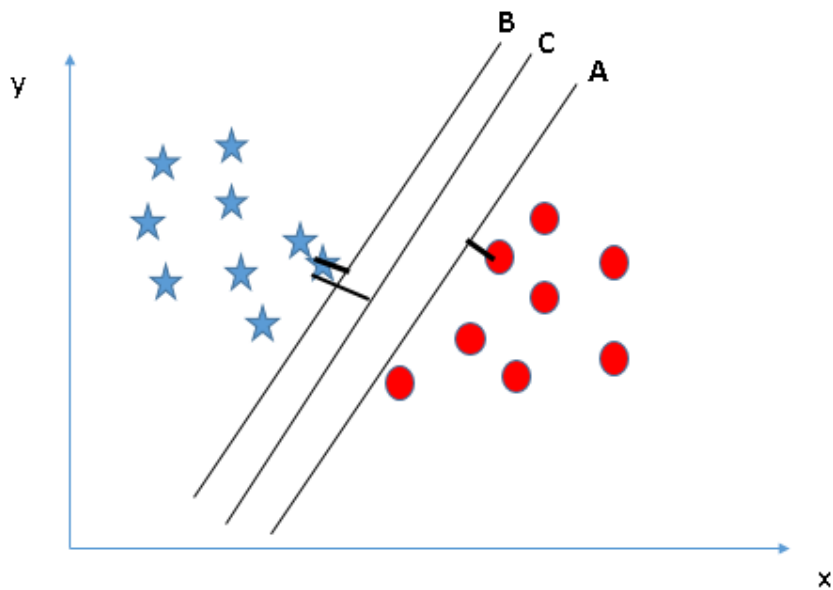


Figure 2.15: SVM Stage 2, Picture Source:[22]

It is also quite effective when it comes to non linear boundary decisions, it is calculated using Gaussian Kernel and similarity function.

Nowadays, due to technological advancement SVM packages like liblinear, libsvm are used to solve for parameter  $\theta$ . The parameter which needs to specify for these packages are choice of parameter C and choice of kernel(similarity function).

## 2.4.2 Unsupervised learning

Supervised learning is one of the most widely used and successful procedure for multilayer feedforward network using backpropagation algorithm [23]. However, it has been limited by poor scaling behaviour: the learning becomes unacceptably slow as the size of the network or problem increases [23]. This problem arises for non linear network where to achieve high accuracy many hidden layers are used. One solution to these kinds of problems is to adopt unsupervised learning procedures.

In unsupervised learning, rather than providing output to be learned by the network, it forms a cluster of its own by randomly initialized cluster centroid. This procedures is repeated several times until it has lowest cost function. There can be two or more cluster centroid, This algorithm is known as K-means algorithm. Some of the applications of clustering are market segmentation, social network analysis.

**Market segmentation:** It consists of large number of customer database that are grouped in different market segments using clustering method which helps in serving market segment in a better way [17].

**Spam emails:** To avoid spam emails in mailbox, unsupervised learning works in effective way. It basically divides an email into different segments like header, sender, content and then they are grouped together. This helps in detecting the spam email accurately.

The following Figure 2.16 shows flow chart of K-means algorithm.

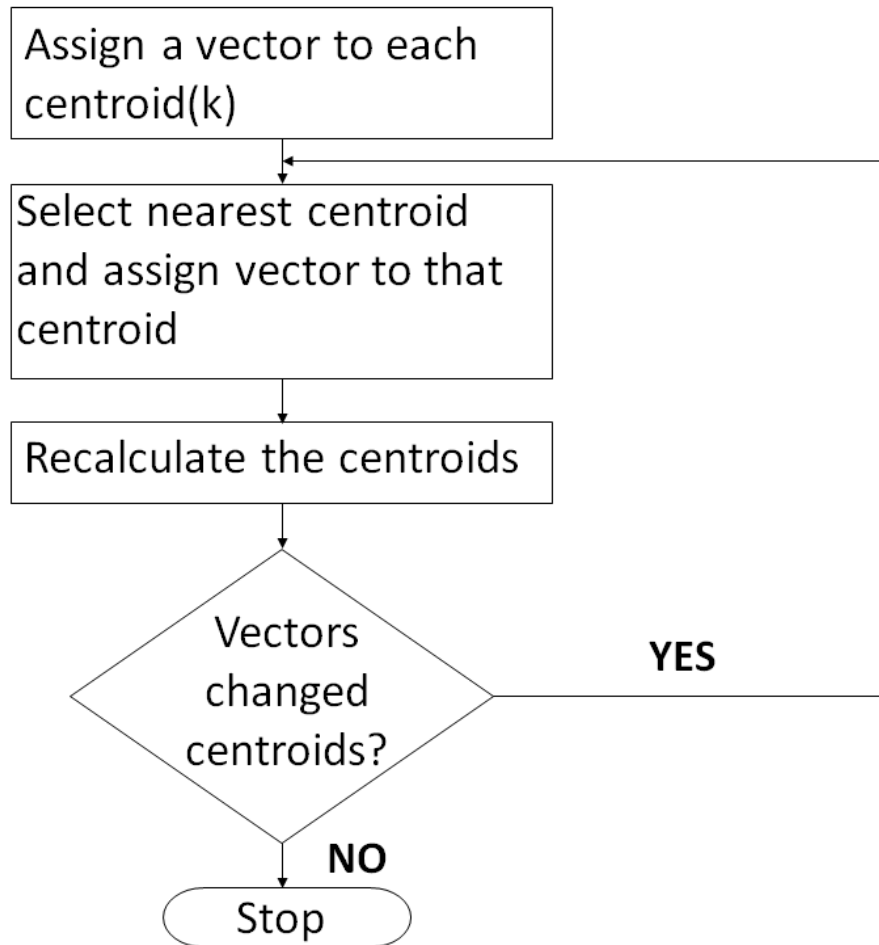


Figure 2.16: Flow chart of K-means algorithm, Picture Source:[24]

Where  $k$  is number of clusters.

Apart from K-means algorithm, few other unsupervised learning algorithms are hebbian learning, encoder learning paradigm, maximise information transmission, spatial coherence as an objective function [23].

## 2.5 Neural Network Regression and Deep Learning Regression

The regression models are mainly used to predict the continuous output values [25]. The NN regression models are mainly used in predicting and forecasting by finding the relationship between the input variables and the output variables. The idea of NN regression and Deep Learning (DL) regression is the combination of linear regression and NN. The only difference between NN regression and DL regression is that the DL model has many hidden layers, where as the simple NN regression model has one hidden layer [26].

The regression can be implemented using following steps:

Step 1: Implementation of activation function.

$$h_W(x) = h_W(x) = W_0(x) + W_1(x) \dots W_n(x) = W^T * X \quad (2.10)$$

Here, X = Input training data

The hypothesis  $h_W(x)$  maps from input x to estimate the value of y by finding best values of  $W_0$ ,  $W_1$  up to  $W_n$ . Ideally the values of W are selected so that  $h_W(x)$  is close to y for the given training examples(x,y). In order to get the values of W the following steps are used.

Step 2: Forward Propagation.

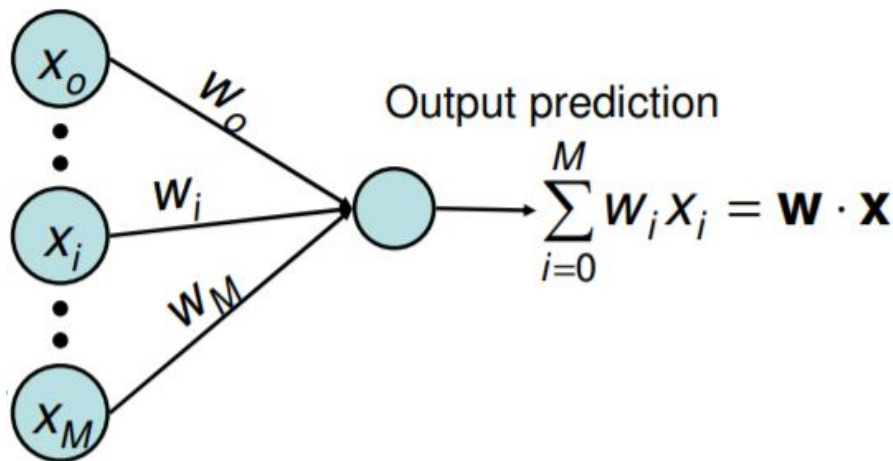


Figure 2.17: Neural network forward propagation

The forward propagation in the NN is represented in the Figure 2.17. A simple linear activation of neuron involves the multiplication of weights  $W_0$  up to  $W_n$  with the input feature matrix X. However other non-linear activation functions like tanh, Rectified Linear Unit (ReLU) and Leaky ReLU can also be used in the hidden layers in order to improve the learning process fast.

Step 3: Implementing Cost function.

$$J(W_n) = \frac{1}{2m} * \sum_{i=1}^m (h_W(x^{(i)}) - y^{(i)})^2 \quad (2.11)$$

The cost function equation (2.11) is minimized to find best possible  $W_n$  values using gradient descent or normal equation. By minimizing the cost function the best  $W_n$  are found out. they are inserted in equation (2.10) to get best fit regression line for the model.

Step 4: Implementing gradient descent to minimise the error.

In order to minimise the cost function we need to find the derivative of the cost function  $\mathbf{J}$  with respect to each of the parameters  $\mathbf{W}_n$ , these derivatives can be obtained by back propagation. The cost function will be minimum only when the derivative of  $\mathbf{J}$  w.r.t  $\mathbf{W}_n$  is zero.

Step 5: Updating the gradient descent.

$$W_n := W_n - \alpha * \frac{\partial J}{\partial W_n} \quad (2.12)$$

By using the partial derivatives of  $\mathbf{J}$  w.r.t  $\mathbf{W}_n$ , gradient descent is computed using the equation 2.8. Here  $\alpha$  is the learning rate. It also defines the rate at which the gradient descent propagates. If  $\alpha$  is too small, then gradient descent may converge very slowly and it might require large training examples. If  $\alpha$  is very large, it may change the value of  $W$  very quickly and might not converge to the global optimal. Hence it is very important to choose the correct learning rate  $\alpha$ .

The steps from 2 to 5 are repeated for gradient descent to reach a global optimal. The values of  $W$  obtained at the end are used to predict new values using the equation 2.10.

## 2.6 Dynamic Programming Method

DP deals with sequential decision processes, which are models of dynamic systems under the control of a decision maker. At each stage or in each period of some planning horizon, the decision maker chooses an action from a set of available alternatives, which generally depends on the current state of the system [27]. The objective of DP is to find optimal performance of the system.

To illustrate further, DP is used to find the optimal values of hybrid torque demand and reference Machine 2 pressure which can be seen in Figure 2.5. DP performs forward simulations and backward simulations in order to achieve optimal control system for above mentioned output.

It is not possible to implement DP online because it performs forward and backward simulations whereas, In real time control, it is not possible to perform backward simulation. Therefore results obtained from DP are trained using ML techniques which are implemented on Simulink model.

## 2.7 Overview of the system model

In the present thesis, a backward-facing simulation model is used to simulate and study the wheel loader system. The same model is used to evaluate all the developed EMS using ML.

### 2.7.1 Backward-facing simulation model

In the backward-facing simulation model, the direction of computation goes backwards, from wheels through all the components of drive-line system, finally the required control inputs are obtained. One of the main advantage of backward-facing simulation is that it does not require a feedback control system. Whereas most of the forward facing simulations require a feedback system. The developed backward-facing simulation model can be used to simulate and study for different drive cycles, component sizes and vehicle size, without redesigning the whole control system. [28]

### 2.7.2 Wheel loader simulation model overview

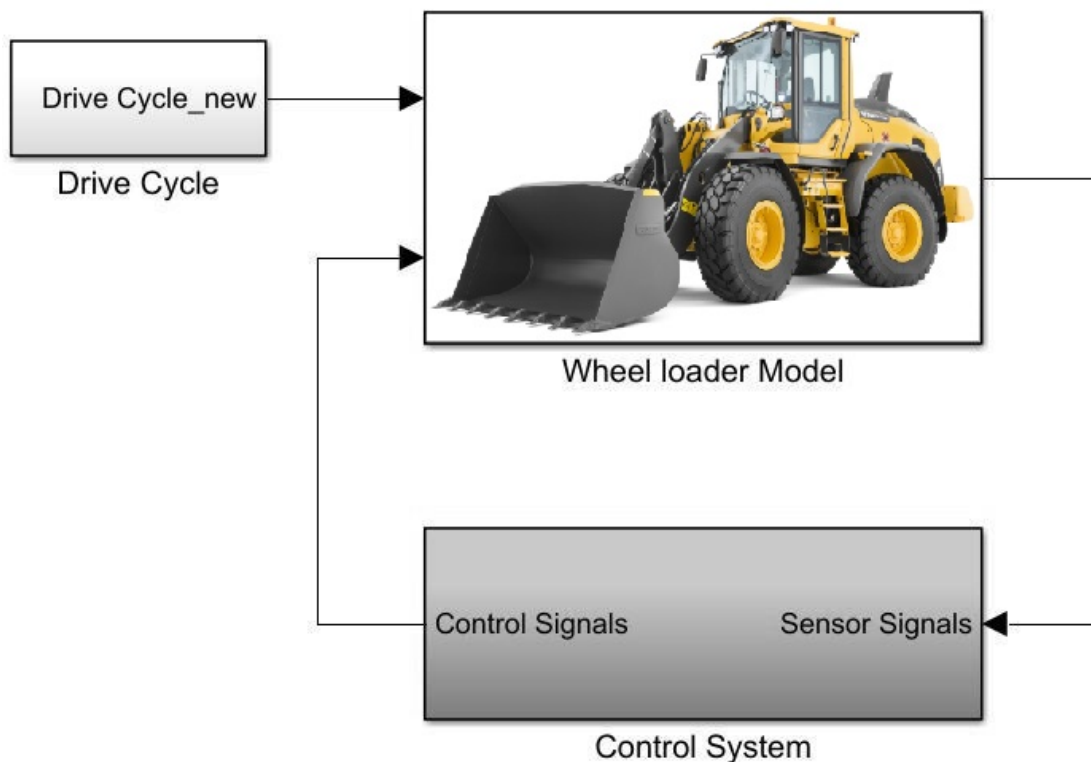


Figure 2.18: Wheel loader simulation model. Picture source Wheel loader: [29]

The wheel loader model used in the present work is shown in the Figure 2.18. The simulation model has been adapted from the previous work [4]. The drive cycle model inputs the wheel loader with signals such as velocity, gear, traction force. The control system takes input from the sensors signals of the wheel loader and provides the appropriate signals back to the wheel loader model.

## 2. Theory

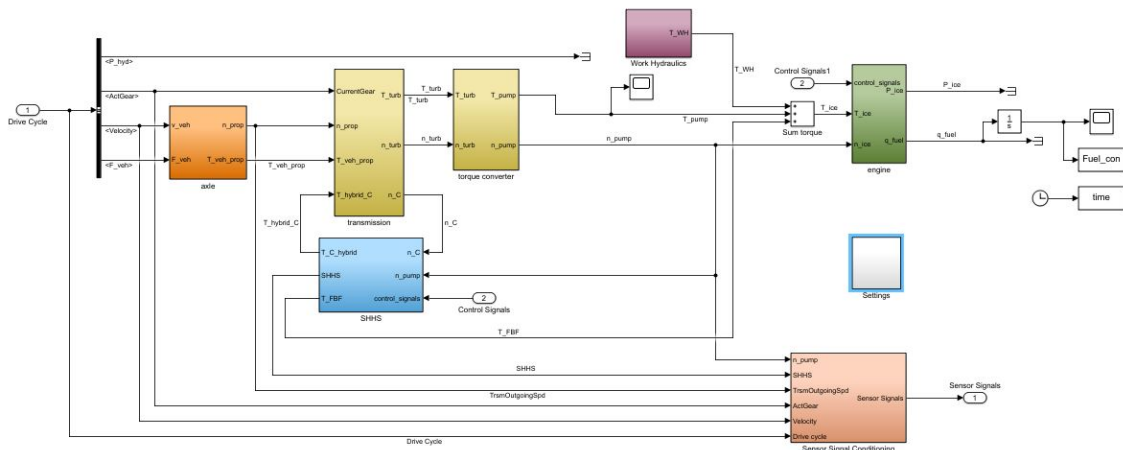


Figure 2.19: Backward-facing simulation model of the wheel loader modelled in Simulink

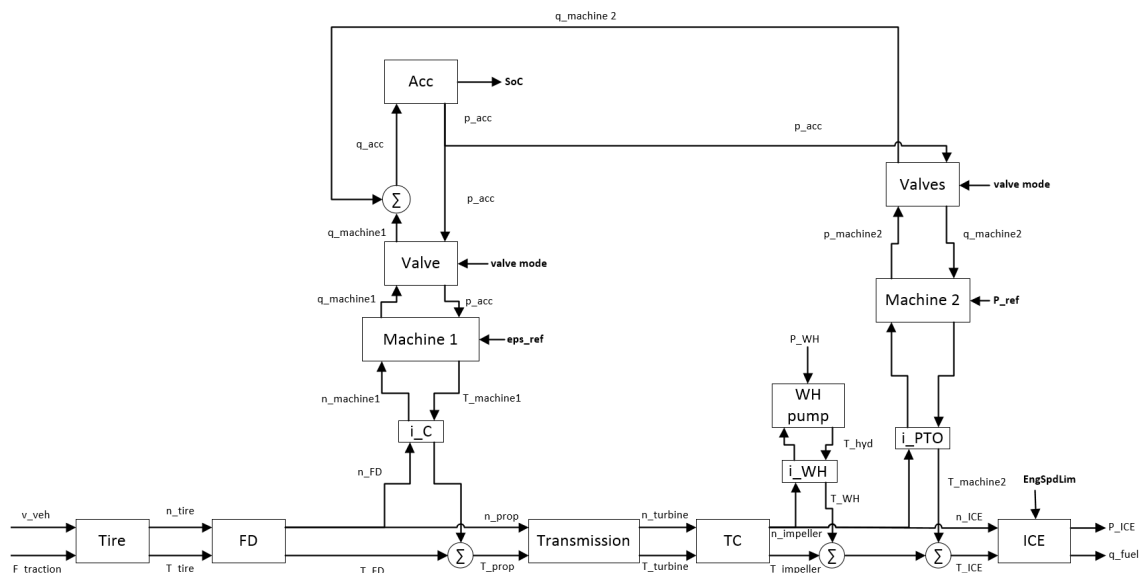


Figure 2.20: Method of calculation of blocks modeled in Simulink. Picture source Wheel loader: [4]

Figure 2.20 shows backward facing simulation model of the wheel loader. The direction of computation is backwards as explained before. The drive cycle provides the inputs which are used in further simulations. The same work-flow principle was adapted and a Simulink model was built. The Simulink model of the wheel loader can be seen in Figure 2.19. In the paper the wheel loader model is briefly explained. The complete description about the components and the model can be seen in [4] [9].

### 2.7.3 Control System

The sensor signals from the different components of the wheel loader are fed to the control system. The main function of the control system is to control the speed

of ICE and the SHHS. The sensor signals provide vital information such as engine speed, vehicle velocity, gear and accumulator pressure. These signals are further processed in control system and the processed signals help in selecting appropriate EMS states. The suitable utilization of SHHS is calculated for these EMS states. The outputs of the control system are then fed to the relevant components for execution.

### 2.7.4 Energy Management Strategies

EMS subsystem previously contained a rule based system to make decisions of energy management system [4]. The RB-EMS was executed using stateflow in Simulink. This system was more complex as it had many sub-sections and loops. EMS has a total of nine states. These states were used for utilising SHHS. Based on the sensor signals and the SHHS diagnosis signals, the most suitable EMS was fed further in the control system execution. In the present work, the rule based system is replaced by NN to classify these nine states.

The states with short description can be seen in Table 2.1. The states named parallel operate as parallel hybrid configuration and the states named series operate as series hybrid configuration. Both these configurations utilise the power from SHHS instead of completely utilising the power from ICE. During the idle state the SHHS system is shut down. The wheel loader enters the charge state at the start of the vehicle and when the accumulator pressure is below pre-charge pressure [4] [9].

Table 2.1: Strategy states. Source [4]

States name	States Description
Idle state	Hybrid system is shut off
Parallel Direction Change	Machine 1 recuperates energy by performing pump operation
Parallel Drive	Machine 1 performs motor operation and assists in propelling the vehicle
Parallel Bucket-fill	Machine 1 performs motor operation and assists in propelling the vehicle
Parallel Accumulator Charge	Machine 1 performs pump operation and pre-charge the accumulators
Series Direction Change	Energy is recuperated by Machine 1 performing pump operation and extra energy stored by Machine 2 performing pump operation
Series Drive	Machine 1 performs motor operation and Machine 2 performs pump operation, assisting vehicle propulsion
Series Bucket-fill	Machine 1 performs motor operation and Machine 2 performs pump operation, assisting vehicle propulsion
Series Accumulator Charge	Machine 2 performs pump operation and Pre-charge the accumulators

Figure 2.21 shows the architecture of the rule based energy management system. It has got state specific rules which determine the utilization of the states. All the states share a common diagnostic rules such as parking brake check and engine check. The architecture is complex in composition. All these state specific rules, transitions rules and diagnostic rules are executed along and a suitable state is determined at every unit time in the simulation, making the system more complex to execute and computationally expensive.

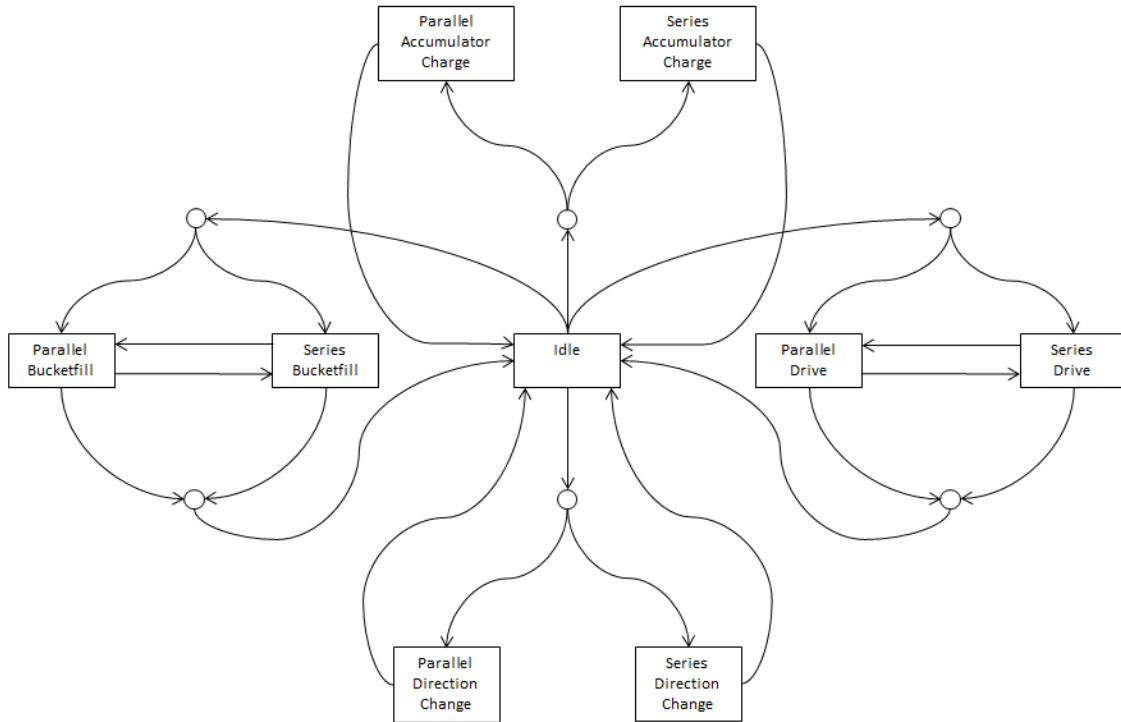


Figure 2.21: RB-EMS state flow architecture. Picture source: [4]

### 2.7.5 Torque control system

The main task of torque control system is to control hydraulic hybrid torque distribution and also determine from where the torque should be retrieved either from ICE or SHHS [4]. This decision is made from hydraulics manager subsystem. Available hydraulic hybrid torque received from hydraulics manager subsystem will act as input to torque control system. Along with torque, brake pedal demand, gear, acceleration pedal demand and several other parameters serve as input to torque control system. In the torque control system, total torque demand, state specific torque demand and brake hybrid torque demand is calculated. It also calculates a suitable limitation in ICE rotational speed to keep the driveline power output close to a conventional wheel loader [4].

# 3

## Energy Management System

The project was mainly divided into two phases as shown below.

### 3.1 Phase 1: Replicate a RB-EMS with a Neural Network

In this phase, RB-EMS was replaced with NN. The same states are maintained and transition rules are learned by the network.

To illustrate further, Figure 3.1 shows schematic representation of implementation.

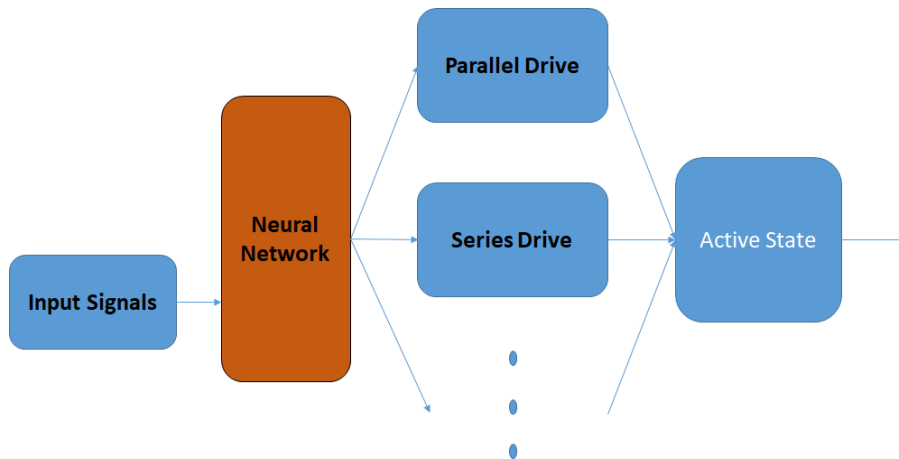


Figure 3.1: Schematic Representation of Implementation

There are many ML techniques which were discussed in the theory section and initially NN and support vector machine were found to be reliable for this kind of implementation because it belongs to one vs all multi-class classification category where only one State is predicted at given instant of time. SVM require high computational cost and not possible to implement on online ML. Therefore NN was selected as it was easy to implement in Simulink as well as machine.

#### 3.1.1 Neural Network Training

NN training was carried out in MATLAB script. Before training the network lot of parameters had to be decided, the key parameters are shown below.

1. Number of layers.

2. Number of input features.
3. Training examples
4. Number of iterations
5. Number of outputs

**Number of layers:** There are no particular rules when selecting layers for the network. Various architectures were tried and tested starting from 1 hidden layer up to 4 hidden layers. Along with number of layers, layer size was also decided by testing different sizes with a training accuracy. Finally, 3 hidden layers with size of 10, 20, 30 for first, second and third hidden layer respectively was chosen as it produced acceptable accuracy. Further increase in hidden layers or size resulted in same accuracy. Sigmoid activation function is used for training

**Number of input features:** Selection of input features plays an important role in the training accuracy as selection of irrelevant may lead to inaccuracy/predicting wrong state. Optimum number of features needs to be selected as increase/decrease in number of features results in high variance/bias. Also, it is reasonable to choose these features since we have certain sensors on our machines today. Before selecting the features for NN, conditions and input features used for RB-EMS was observed. Since the task involved was to replicate RB-EMS so, the following 7 features were selected based on their key decisions/conditions involved in selecting States for RB-EMS.

1. Acceleration Pedal Demand
2. Gear
3. Brake Pedal Demand
4. State of Charge
5. Engine Speed
6. Machine Velocity
7. Torque Converter Output Speed

**Training examples:** The training examples were taken from real world measurements for SLC of a wheel loader. In total there were 54 SLC. These data were divided into three categories as follows.

1. Training Data: This data is used to train the NN. The data was reformatted/restructured according to the requirement of the script. For training data, 33 SLC data are used for training data.
2. Cross Validation Data: Cross-validation is a data resampling method to assess the generalization ability of predictive models and to prevent overfitting [30]. This data also helps to analyze if there is a requirement of additional data / input feature for training data. 9 SLC data are used for cross validation data.
3. Test Data: This data gives information about the adaptation to the training data. This estimate quantifies the generalization ability of the model [30]. Testing data also gives an idea whether trained network follows same pattern when a new data is provided to the network. 9 SLC data are used for Test data.

**Number of iterations:** Several number of iterations are performed to find global optimal value in order to achieve least possible cost function. At certain number of iterations, it reaches global optimal value and further increase in number of iteration does not change the results significantly.

**Number of outputs:** From Table 2.1, Five output states is used for training. Idle state was left out of training because RB-EMS is built in a manner that every time when a system needs to change from one state to another (this does not include switching from parallel mode to series mode or vice versa), it has to pass through Idle state for few milliseconds. This is quite difficult for NN to predict because the training data are based on ruled based strategy and there are instances when input features have same value for Idle state and other states. These overlapping data results in wrong prediction of state by NN. Idle state was predicted in Simulink MATLAB function by using few operating conditions.

After deciding on key parameters, next step was to extract good accuracy from network. The accuracy was measured using confusion matrix. A confusion matrix is a tabular way of visualizing the performance of your prediction model. Each entry in a confusion matrix denotes the number of predictions that were made by the model where it classified the classes correctly or incorrectly [31]. In this phase, performance is evaluated by comparing predicted results by NN with RB-EMS states. As the goal of this phase was to replicate rule based energy management strategy with NN. Confusion matrix is created for testing of neural network. NN results is considered as predicted value and results of ruled based strategy is considered as actual target value.

### 3.1.2 Implementation of Neural Network in Simulink

The optimal trained NN in the MATLAB script was implemented in Simulink using MATLAB function block.

The following Figure 3.2 shows the working principle of MATLAB function block implemented in Simulink.

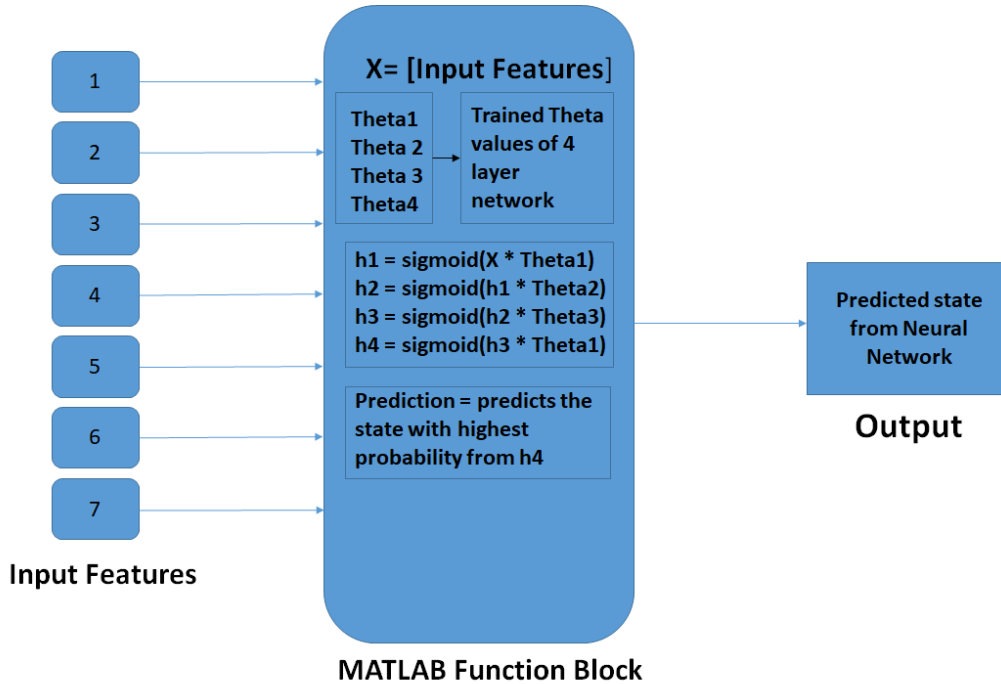


Figure 3.2: ML implementation on simulink

Figure 3.2 shows there are 7 input features connected to MATLAB function block. These features are mentioned in 3.1.1. The theta/weights are obtained from NN trained in MATLAB script. As shown in Figure 3.2, sigmoid function is applied to the product of Theta 1 and input features. similar operations are carried out for 2, 3 and 4<sup>th</sup> layer respectively. Finally, the state with highest probability predicted by NN (h4) is sent to the output. Furthermore, few conditions are added to MATLAB function block to increase the accuracy.

## 3.2 Phase 2: Optimization of hybrid torque and Machine 2 pressure.

### 3.2.1 Optimization using Dynamic Programming

As discussed in the previous chapter, DP cannot be implemented online because of high computational cost and require prior knowledge of power demand. One way of achieving a close-to-optimal online EMS is by using results from DP optimization as inputs for supervised learning algorithms [5]. In this thesis, NN one of the supervised learning algorithm is used to learn the decision making process by DP and implement

it online as an EMS [5]. DP is used to obtain optimised values for Hybrid torque request and Machine 2 pressure. With suitable DP model, SLC are simulated using DP. The optimised results of Hybrid torque request and Machine pressure 2 obtained from DP are expected to be learnt by NN.

### **3.2.2 Training of Hybrid torque request**

#### **3.2.2.1 Obtaining Data**

The optimized hydraulic hybrid torque request that was obtained from DP tool was used as a target value to be trained using ML. A total of 15 test cycles that gave a feasible solution in DP tool were used to obtain the data required for ML. Feature data to train the ML model were also obtained from the optimised driving cycles of DP and some of the feature data were obtained from the respective test cycle data.

**The training feature data that were obtained from optimised cycles of DP:**

1. Engine speed
2. SOC of the accumulator
3. Gear

**The training feature data that were obtained from test cycle data:**

4. Traction force
5. Vehicle velocity

Two regression models were used to train the hybrid torque request. These models were compared with the training accuracy, complexity and the duration of training. Since the NN regression model had the best accuracy, it was selected for the implementation in the Simulink model.

**Regression models trained:**

1. Linear regression
2. NN regression.

#### **3.2.2.2 Linear regression**

##### **Feature normalization**

The feature data that were used to train had a different range, The learning process of the model was very slow and there were high chances of the gradient descent reaching a local optimal. Hence it was necessary to normalise the training features to a common scale. This was done by dividing the maximum value of a feature with all the data points of that feature, by doing this all the features were within the scale of -1 to 1. An example of feature normalization is shown in Equation 3.1

$$Velocity_{normalized} = \frac{Velocity_{(1-m)}}{Velocity_{max}} \quad (3.1)$$

After normalizing all the features, the feature data was arranged into a matrix X of dimension (m,n) where m is the number of data points or examples, and n was the number of features. The target feature, hybrid torque request which was to be trained was also arranged into a matrix Y of dimension (m,1).

#### Gradient descent

The values of  $\theta$  were initialized to zero. The cost function was found out by using the Equation 2.2. The gradient descent was obtained by Equation 3.2 and performing the gradient update of  $\theta$  using Equation 3.3. The gradient update was dependent on learning rate  $\alpha$ .  $\alpha$  was the hyper parameter, which was set by trial and error method to obtain a good gradient descent. These steps were repeated for for number of iterations, which was again a hyper-parameter. Number of iterations was also tuned to get a good results. The values of  $\theta$  at the end of gradient descent was saved.

$$\delta = \frac{1}{m} * (X * ((X * \theta) - y)) \quad (3.2)$$

$$\theta = \theta - \alpha * \delta \quad (3.3)$$

#### Predicting new values by the trained regression model

The new value with its feature data was multiplied with the saved values of  $\theta$  to get the predicted value. In the Equation 3.4,  $\hat{Y}$  represents the predicted value.

$$\hat{Y} = X * \theta \quad (3.4)$$

#### 3.2.2.3 Neural network linear regression

##### Feature normalization

The same features were used to train the NN regression model. The feature normalization was also carried out using the Equation 3.5.

$$y = \frac{((y_{max} - y_{min}) * (x - x_{min}))}{(x_{max} - x_{min}))} + y_{min} \quad (3.5)$$

[32]

Where,

x: is an element of the vector (input or output) to normalize.

$x_{max}$ : is the value of the greatest element of the vector to normalize.

$x_{min}$ : is the value of the smallest element of the vector to normalize.

y: is the normalized value of x.

$y_{max}$ : is the maximum value (1).

$y_{min}$ : is the minimum value (-1).

### Forward Propagation and Backward Propagation

The NN was a two layered network with one hidden layer and one output layer. Two activation functions namely, tansig and purelin were used during the training and to predict new values. Tansig was used as activation function of neurons in the hidden layer and purelin was used as activation function of the output layer. The equation governing tansig is shown in Equation 3.6. The mathematical functionality of tansig function is explained in [33]. Purelin activation function was the linear combination of inputs from the prior node. Equation 3.7 explains the purelin activation [34] [35]. The derivatives of the cost function with respect to input features are calculated during back propagation and the gradients were updated according to Equation 2.12. Hyper-parameter, number of neurons in the hidden layers was set to 50 and number of iterations was set to 1000, these parameters were set by trail and error method to get the faster learning and accuracy.

$$a = \frac{2}{(1 + \exp(-2 * h_w(x)))} - 1 \tag{3.6}$$

Where,  $h_w(x)$  was found by equation 2.1

$$a = \sum_{i=1}^l W^T * X + b \tag{3.7}$$

Where, i: is the number of neurons in the previous layer.  
b: is the bias term.

### Prediction of new values

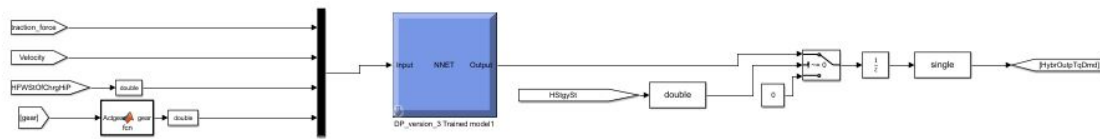


Figure 3.3: Neural network regression implementation in Simulink

The data coming from the sensor signals were arranged into a matrix using mux as show in Figure 3.3, a total of 4 features were arranged into a matrix of size (1,4). The forward propagation calculation block was fed with this input matrix. Figure 2.17 and Figure 3.5 shows the forward propagation in predicting new values, the features/inputs gets multiplied with the corresponding weights and tansig activation function is used in the neurons of Layer 1 to calculate the value of **a** using Equation 3.6.

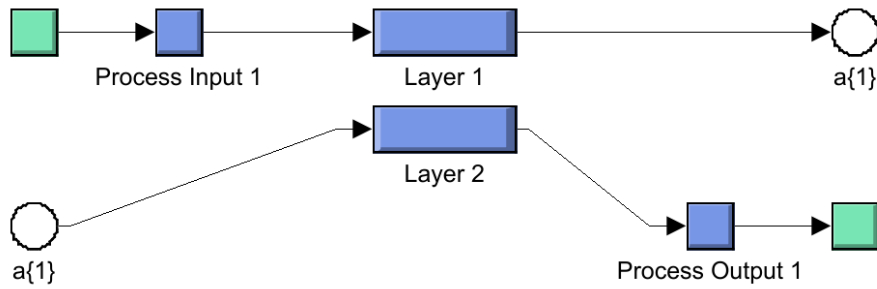


Figure 3.4: Neural network regression Simulink block

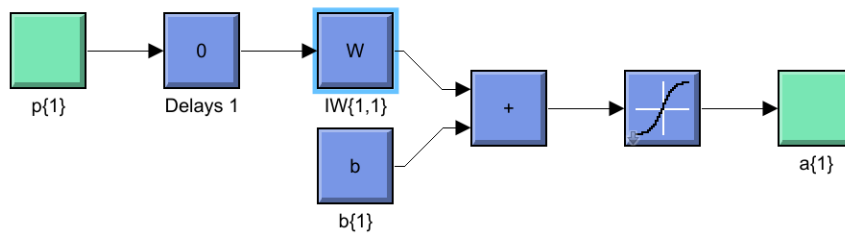


Figure 3.5: Forward propagation inside Layer 1

The values of  $\mathbf{a}$  obtained in neurons of Layer 1 are then multiplied with the corresponding weights similarly as shown in Figure 3.6, purelin activation function is used in the Layer 2 to get the output using Equation 3.7, the output of the network is pre-processed using Equation 3.5 for the torque request to have its original value instead of the normalized value.

During Idle conditions the Torque Request should be zero, therefore a switch was implemented to ensure that the torque request was zero during idle conditions. The filter operation using a switch can be seen in Figure 3.3.

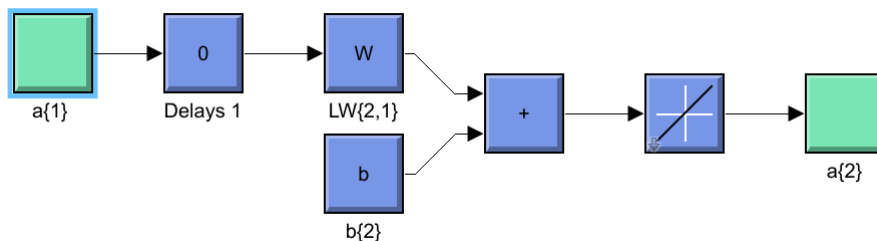


Figure 3.6: Forward propagation inside Layer 2

The output obtained from NN regression was then divided with the shaft gear ratio to get the actual hybrid output torque before loss. In reference to Figure 2.21,

the complex rule based state flow system is replaced by simple regression model to predict the hybrid torque request which is shown in Figure 3.4.

### **3.2.3 Training of the Machine 2 pressure**

#### **3.2.3.1 Obtaining training data**

The optimized Machine 2 pressure that was obtained from DP was used as the target value to be trained using ML. Feature data to train the model was obtained from the optimized driving cycles (output) of DP. Some extra feature data was created by performing algebraic operations on the feature data obtained from the DP tool. Only three features were available for the training because some of the features were unavailable in the optimized drive cycle data of the DP tool, and some features had a problem of data mismatch.

**The training feature data that were obtained from the optimized cycles of DP:**

1. Hybrid torque request
2. SOC of the accumulator.

**The training features that were obtained by algebraic operations:**

3. Hybrid torque request \* SOC of the accumulator

The algebraic operation 'hybrid torque request \* SOC' gives the neural network a better dimension of data and hence improves the training accuracy.

Similar to the training of hybrid torque, Machine 2 pressure was also trained using two models and since NN regression model had better accuracy over linear regression model, it was selected for the implementation in the Simulink.

#### **3.2.3.2 Linear regression**

The modelling and implementation of linear regression model and NN regression model are same as explained before. There is only difference in the values of hyper-parameters that were used to tune the model for better accuracy.

#### **3.2.3.3 Neural network regression**

The NN model is a two layered NN, with 3 inputs, 30 neurons in the second Layer and one neuron or unit in the output layer. The number of iterations were set to 1000.

#### Prediction of new values

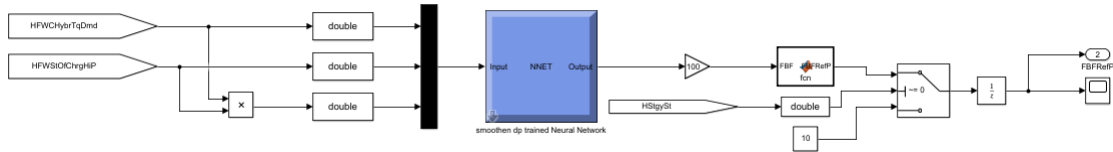


Figure 3.7: Machine 2 pressure neural network implementation in simulink

The data coming from the sensor signals at every millisecond and its algebraically operated signals were arranged into a matrix in the math function block as shown in Figure 3.7. A total of 3 features were arranged into a matrix of size (1,3). The forward propagation calculation block was fed with this matrix. The prediction of new values for Machine 2 pressure using forward propagation block is similar to that explained in NN regression for hybrid torque calculation, see Figure 3.5, 3.6.

The math function block was used to set the upper limit of the Machine 2 pressure request as 350 bars and during the idle conditions the pressure request was set to be 0 bars. During Idle conditions the Machine 2 pressure should be zero, therefore a switch was implemented to ensure that the pressure was zero during idle conditions. The filter operation using a switch can be seen in Figure 3.7.

# 4

## Results and Discussion

The ML technique developed is evaluated by simulating hydraulic hybrid wheel loader model with implementation of ML in Simulink. Firstly, in the Phase 1 training accuracy of ML replicating rule based strategy is shown. Then the results of implementation in Simulink with training, cross validation and test data are shown using confusion matrix. The comparison of ML predicted states and Rule based predicted states using same data are displayed. In the Phase 2, Optimized DP results are analyzed by plotting Hybrid torque, State of Charge (SOC) and Velocity. Similarly, Machine 2 pressure, SOC and Velocity are analyzed. Followed by comparison of accuracy between linear regression and NN regression. Finally, results of NN regression implemented in Simulink is shown where it is analyzed whether the network produces reasonable results.

### 4.1 Phase 1

Figure 4.1 shows training accuracy of states predicted by ML. Confusion matrix shows comparison of predicted value with actual target value. There are total of 5 states predicted using ML excluding idle state which was discussed in Section 3.1.1. State 1, State 2, State 4, State 6 and State 9 represents following States.

- **State 1** represents **Parallel Direction Change**
- **State 2** represents **Parallel Drive**
- **State 4** represents **Parallel Bucket-fill**
- **State 6** represents **Series Bucket-fill**
- **State 9** represents **Series Drive**

#### 4. Results and Discussion

The states description are explained in Table 2.1. From the Figure 4.1, It can be observed that state 1 has accuracy of 99.2% and state 2 and 9 has accuracy of 89.9% and 80.1% respectively. State 2 and 9 perform same task with state 2 acting as parallel hybrid and state 9 acting as series hybrid. So in this way state 2 predicting state 9 or vice versa would not cause any problem. Similarly, state 4 and 6 perform the same way as state 2 and 9. So it can be said that accuracy of state 4 and 6 are good as it is either predicting state 4 or 6. Finally, looking at Figure 4.1 it can be observed that ML is successful in replicating ruled based predicted states and these training results are extracted from MATLAB and implemented in Simulink model.

		Predicted Value						
		1	2	4	6	9		
Actual Target Value	1	15442	6		3	114	99.2%	0.8%
	2		13399	1		1497	89.9%	11.1%
	4			38264	3590		91.4%	8.6%
	6			6315	14221		69.2%	30.8%
	9		2590			10403	80.1%	19.9%

Figure 4.1: Confusion matrix of ML Training Accuracy

Figure 4.2, 4.3, 4.4 shows results of Phase 1 implementation of ML in Simulink model compared with ruled based predicted states using confusion matrix for training, cross validation and test data respectively.

Figure 4.2 shows results of ML implemented using Training data. From the results, it can be seen that ML has good accuracy while predicting state 1, 4 and 6 but accuracy of state 2 and 9 are slightly on lower side because of conditions put up for Idle state 0. As stated in previous chapter, Idle condition was not considered for training instead certain operating conditions were implemented in MATLAB function block such that system always goes to idle state for few milliseconds when switching from one state to another. Due to this, few conditions are overlapping with state 2 and 9 which is main reason for having low accuracy.

0	578345	38168	944	11467	1217	74	91.7%	8.3%
1	3110	12455					80.0%	20.0%
2	6063		8446			388	56.7%	43.3%
4	26			33188	8640		79.3%	20.7%
6	2			3830	16704		81.3%	18.7%
9	6135		1184			5674	43.7%	56.3%
	0	1	2	4	6	9		

Predicted Value

Figure 4.2: Confusion matrix of ML in Simulink for training data

Figure 4.3, 4.4 shows result of ML implementation using cross validation and test data. Results indicate that ML is able to produce good accuracy for new data which has not been trained. This signifies that there was no issue regarding bias/variances and network is able to adapt to new data while maintaining good accuracy.

0	199750	14977	84	2250	596	7	91.5%	8.5%
1	33	3991					99.2%	0.8%
2	2153		2800			177	54.6%	45.4%
4	3			5987	3164		65.4%	34.6%
6				303	3881		92.8%	7.2%
9	1975		724			1815	40.2%	59.8%
	0	1	2	4	6	9		

Predicted Value

Figure 4.3: Confusion matrix of ML in Simulink for cross validation data

0	181155	13175	142	2392	351	12	91.6%	8.4%
1	38	4157					99.1%	0.9%
2	1714		2302			207	54.5%	45.5%
4	2			7159	2821		71.7%	28.3%
6	2			478	3673		88.4%	11.6%
9	1772		73			1836	49.9%	50.1%
	0	1	2	4	6	9		

Predicted Value

Figure 4.4: Confusion matrix of ML in Simulink for test data

Figure 4.5 shows results of comparison of ML implementation in Simulink against ruled based predicted states. Results indicate that ML is quite successful in replicating strategy states predicted by ruled based energy management system. From Figure 4.5, it can be seen that ML using NN is able to follow same pattern as that of RB-EMS. Apart from few inaccurate results in predicting state 1 and Idle state due to overlapping of few conditions which is explained in previous paragraph, it enters and exit bucket-fill phase(state 4 and 6) accurately which is an crucial state for wheel loader.

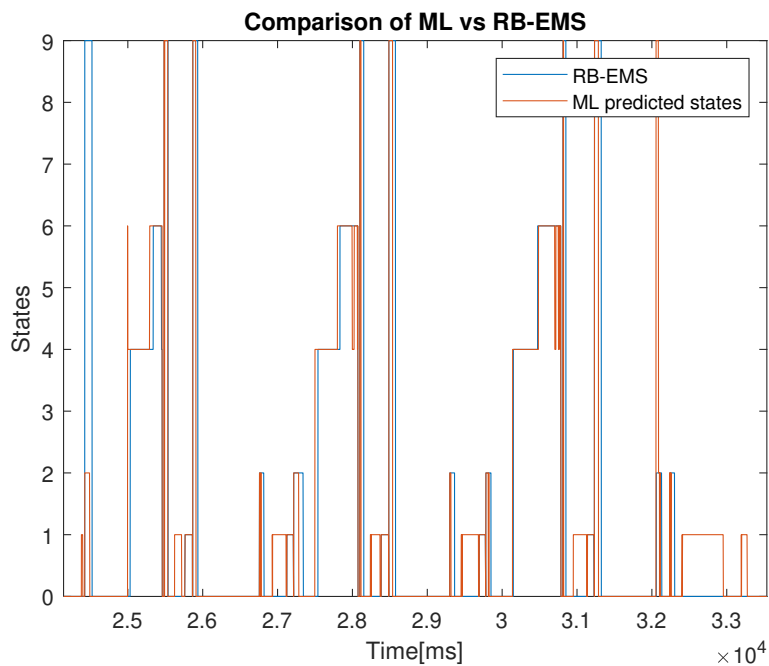


Figure 4.5: Comparison of ML Predicted states with RB-EMS predicted states

## 4.2 Phase 2

### 4.2.1 Dynamic Programming results

The DP search for ways to improve the fuel efficiency by utilizing the hybrid system at the right time and hence minimizing the dependency on the torque converter during its low efficiency phases. Bucket-fill phase is one of the phases during which the torque converter has a very low efficiency. Utilising hybrid system during bucket-fill phase would reduce the fuel consumption. DP optimises the power request from ICE and SHHS.

Figure 4.6 shows the graph of vehicle velocity, SOC and hybrid torque request for two SLC's. The yellow box shown in the Figure 4.6 corresponds to the bucket-fill phase. A large amount of torque was delivered from the hybrid system. Hence the SOC of the accumulators was decreasing during bucket-fill phase. It can also be seen that just before the bucket-fill phase the accumulators were charged and SOC was slightly increased, as the SOC was about to be consumed during the bucket-fill phase. The green box shown in the Figure 4.6 corresponds to the braking phase. During braking, energy was recuperated and hence we can see that there was slight increase in SOC.

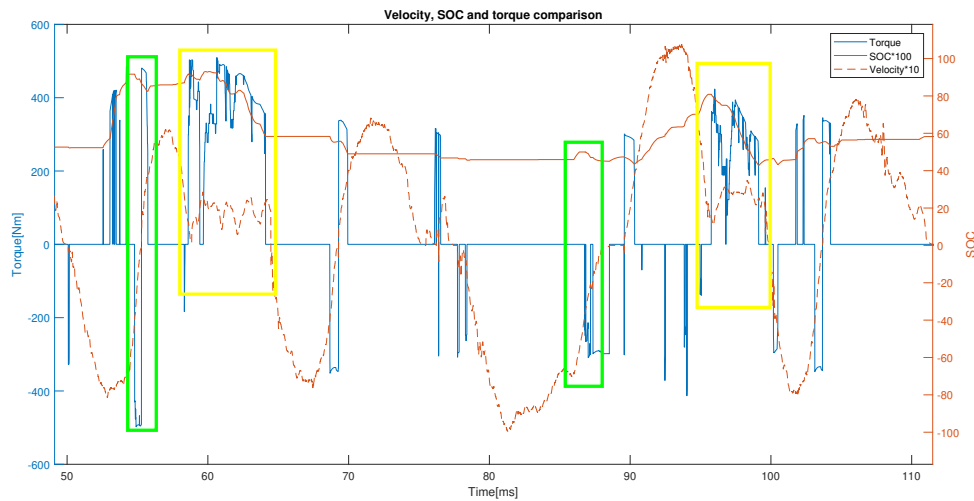


Figure 4.6: Comparison of Hybrid torque, SOC, Velocity obtained from optimized DP

Figure 4.7 shows the graph of Vehicle velocity, SOC and the Machine 2 pressure for the two SLC's. The bucket-fill phase is highlighted by a yellow box. It can be observed that the SOC of the accumulators was decreasing and the SOC of the accumulators was used to perform the bucket-fill operation. Machine 2 also assisted the bucket-fill operation and hence the pressure on Machine 2 also increased during this phase. It can also be observed in the green box that during braking there was a slight increase in SOC of the accumulators. During braking, the energy was recuperated by Machine 1 and increased the SOC of the accumulators. Since the

pressure in the accumulators was increasing, the pressure on Machine 2 should also increase in order to further assist HFW, and hence it can be observed that the Machine 2 pressure was also increased during braking. There was a lot of noise in the Machine 2 pressure data. A filter was used in the further application to smoothen the data.

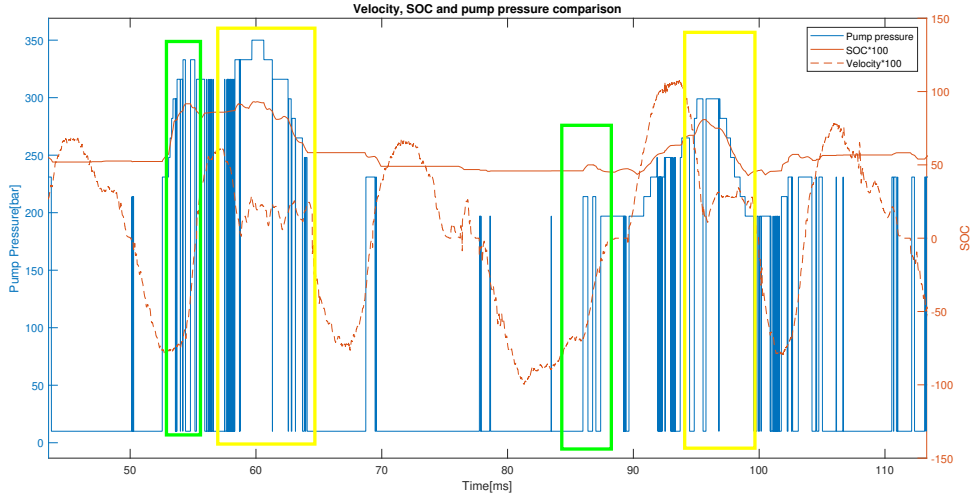


Figure 4.7: Comparison of Machine 2 pressure (pump pressure), SOC, Velocity obtained from optimized DP

## 4.2.2 Comparison of accuracy of Linear Regression and Neural Network Regression

The optimized hybrid torque request from the DP was trained using linear regression and NN regression. Both the regression models were trained using the same data points. The training data was divided into training set and test set. The training set was used to train and tune the hyper parameters to get good accuracy. Test set was used to predict the hybrid torque request values for the new untrained data points.

### 4.2.2.1 Training on prediction of hybrid torque request

Figure 4.8 shows the graph of target values of hybrid torque (obtained from DP) versus predicted values of hybrid torque from the linear regression model on training set. The learning rate ( $\alpha$ ) and number of iterations were tuned to get the best accuracy. The final values of learning rate and number of iterations used were 0.01 and 30000 respectively. The blue line in the graph is the target hybrid torque and the red line is the predicted values from the linear regression model. The mean squared error was found out to be 3760.27. Since, there were only 4 features available to train the model, the accuracy of the simple linear regression model was less. It can also be observed that the predicted values followed the trend of the target values, but there was a lot of deviations and spikes in the predicted data and also lot of

disturbance in the predicted data can be observed whenever the target data was 0 Nm.

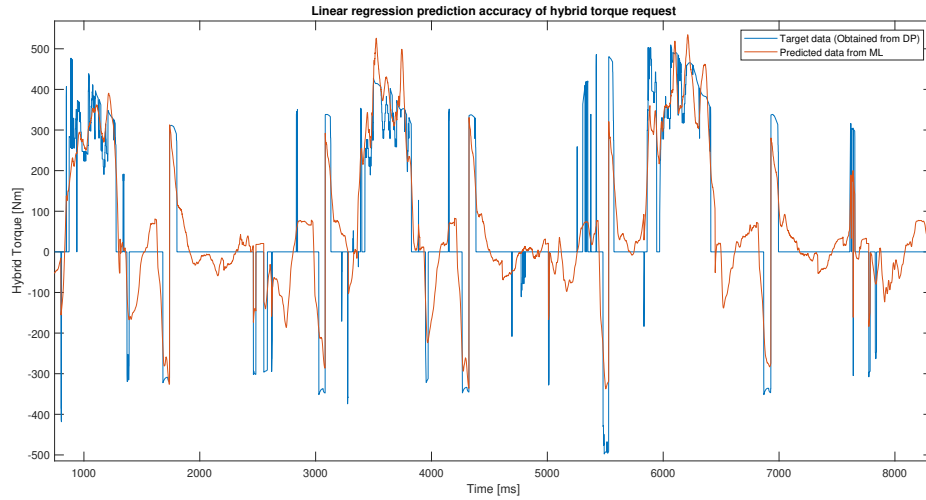


Figure 4.8: Training accuracy of the predicted hybrid torque values using linear regression model

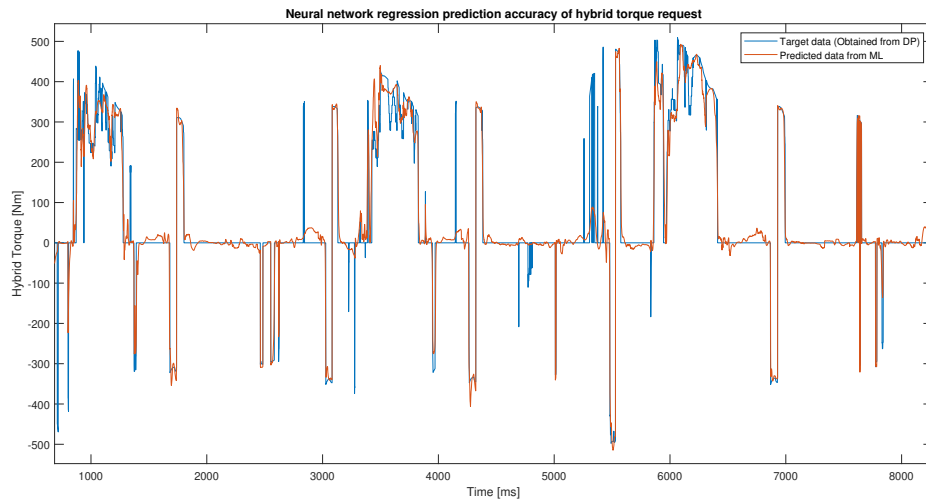


Figure 4.9: Training accuracy of the predicted hybrid torque values using neural network regression model

Figure 4.9 shows the graph of the target values of hybrid torque (obtained from DP) versus predicted values of the hybrid torque from the NN regression model on the training set. The mean squared error between the target and the predicted data was found out to be 2627.91, which is less than the mean squared error of linear regression model. A clear follow-up trend of the predicted data can be observed, even at higher torque request unnecessary spikes were not present. The predicted data was a lot more smoother in predicting the neutral values (0 Nm) than the

predicted data of the linear regression model. The NN regression predicted more accurately on the training set than the linear regression model, it can be observed from Figures 4.8 and 4.9.

### 4.2.2.2 Testing on prediction of hybrid torque request

The trained linear regression model was used to predict the hybrid torque on test data set. Figure 4.10 shows the graph of target data versus the predicted hybrid torque request. The mean squared error over the test set was 3850.42. Similar to the predictions on trained data, the predictions on the test data was also not accurate and a lot of spikes and deviation from the target data can be observed. The fewer features and training data were the reasons for the model being less accurate.

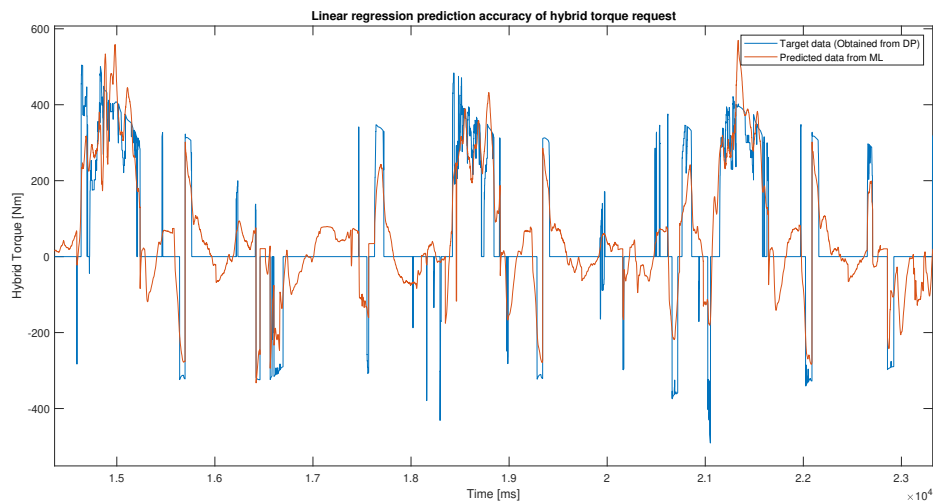


Figure 4.10: Test accuracy of the predicted hybrid torque values using linear regression model

The trained NN regression model was used to predict hybrid torque on the test data. Figure 4.10 shows the graph of target data versus the predicted hybrid torque request. The mean squared error over the test set was 2513.26, which is lesser than the mean squared error of the linear regression model. Although there was some disturbance in predicting the neutral value (0 Nm), the overall accuracy was much better than the linear regression model.

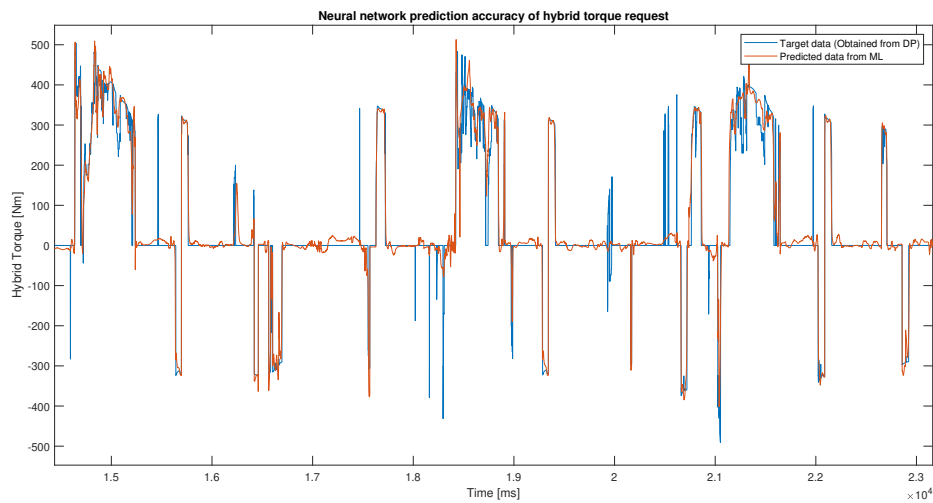


Figure 4.11: Test accuracy of the predicted hybrid torque values using neural network regression model

Since the accuracy of the NN model was better than the linear regression model in predicting hybrid torque, it was considered for further implementation. The problem in wrongly predicting neutral values was solved by using a filter in further implementation.

#### 4.2.2.3 Training on prediction of Machine 2 pressure

Figure 4.12 shows the graph of target values of Machine 2 pressure (obtained from DP) versus predicted values of Machine 2 pressure from the linear regression model on training set. The learning rate ( $\alpha$ ) and number of iterations were tuned to get the best accuracy. The final values of learning rate and number of iterations used were 0.15 and 10000 respectively. The blue line in the graph is the target hybrid torque and the red line is the predicted values from the linear regression model. The mean squared error was found out to be 6162.08. It can be observed from the Figure 4.12 that the predicted data was failing to follow the target data at many regions. The model was not accurate in predicting the neutral value (0 bars). Since the actual mapping that the training model had to learn was a complex nonlinear function, the linear regression model could not accurately learn the non linear relation between the features and the target values resulting in bad accuracy. Lack of training data also contributed for the model to be less accurate.

## 4. Results and Discussion

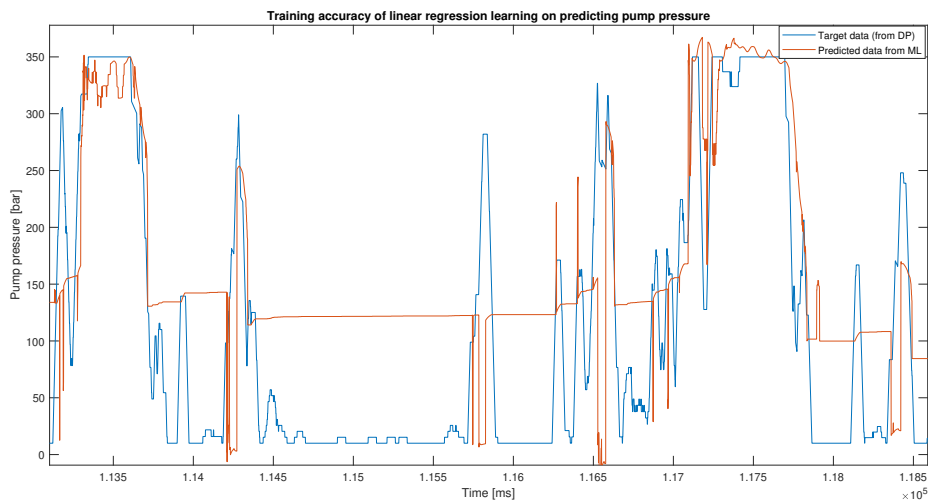


Figure 4.12: Training accuracy of the predicted Machine 2 pressure (pump pressure) values using linear regression model

Figure 4.13 shows the graph of the target values of Machine 2 pressure (obtained from DP) versus predicted values of the Machine 2 pressure from the NN regression model on the training set. The mean squared error between the target and the predicted data was found out to be 2976.28, which is less than the linear regression model. The NN regression predicted more accurately on the training set than the linear regression model, it can be observed from Figures 4.12 and 4.13. The predicted data followed the target data more accurately than the linear regression model, but still the NN predictions were not accurate in predicting the neutral value (0 bars). The main reason was the lack of features and data to train the model.

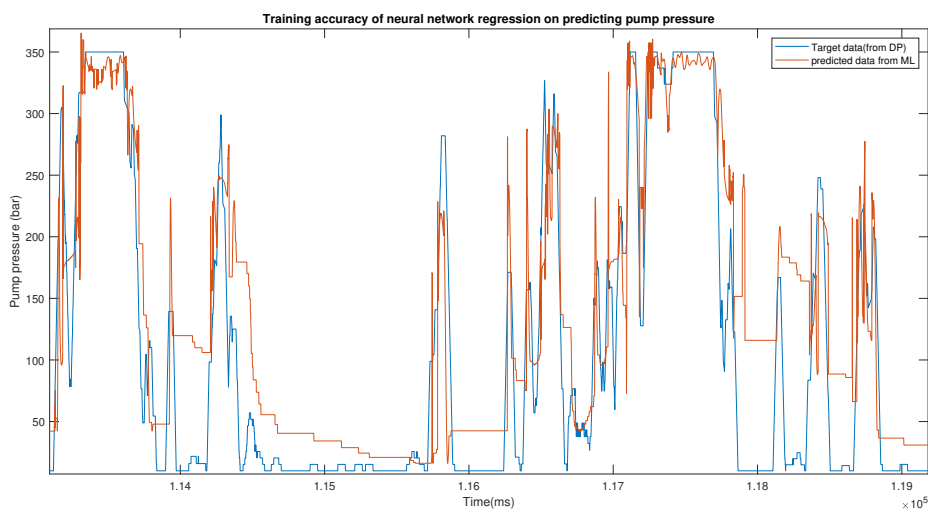


Figure 4.13: Training accuracy of the predicted Machine 2 pressure (pump pressure) values using neural network regression model

#### 4.2.2.4 Testing on prediction of Machine 2 pressure

The trained linear regression model was used to predict the Machine 2 pressure on test data set. Figure 4.14 shows the graph of target data versus the predicted Machine 2 pressure. The mean squared error the test set was 6228.76. The predicted data followed the trend of target data with very poor accuracy. The model also failed in predicting the neutral values similar to the predictions made on training data set.

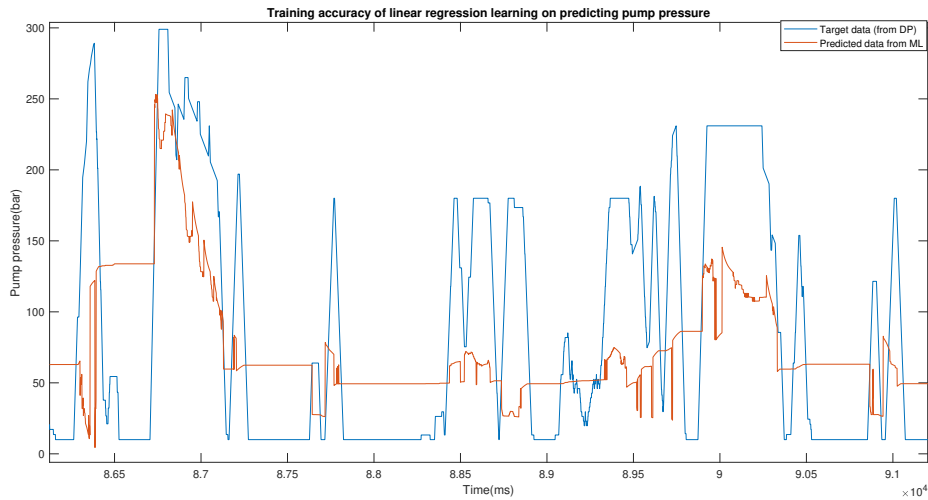


Figure 4.14: Training accuracy of the predicted Machine 2 pressure (pump pressure) values using linear regression model

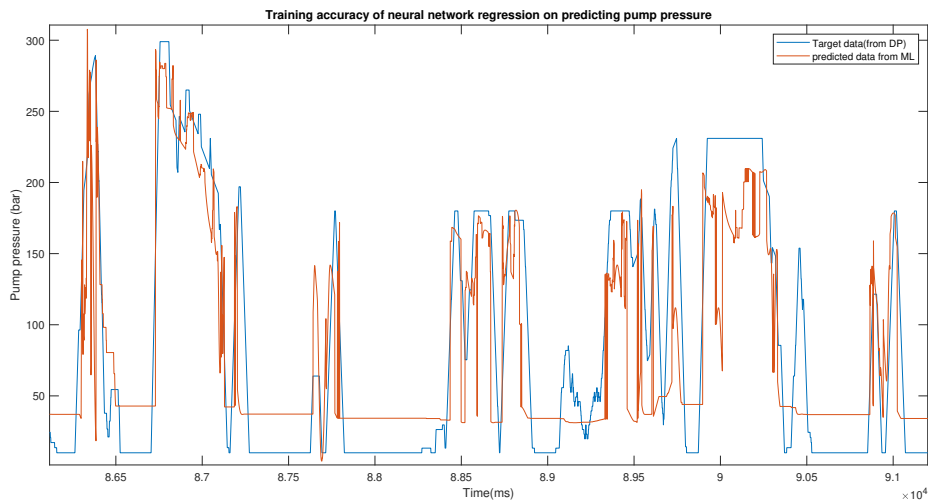


Figure 4.15: Training accuracy of the predicted Machine 2 pressure (pump pressure) values using NN regression model

The trained NN regression model was used to predict the Machine 2 pressure on test data set. Figure 4.15 shows the graph of target data versus the predicted Machine 2 pressure. The mean squared error over the test set was 2983.35. The predicted

#### 4. Results and Discussion

---

data followed the trend of target data with more accuracy, but model suffered the problem of wrongly predicting the neutral values. When compared with the predictions of linear regression model on test set, the NN model was more accurate.

Since the accuracy of the NN model was better than the linear regression model in predicting Machine 2 pressure, it was considered for further implementation. The problem in wrongly predicting the neutral values was solved by using a filter in further implementation.

### 4.2.3 Results of neural network regression implemented on Simulink model

The two NN regression models are used to predict hybrid torque and Machine 2 pressure were implemented to replace the rule-based systems for calculating hybrid torque request and Machine 2 pressure.

Figure 4.16 shows the graph of vehicle velocity, SOC and hybrid torque request for three SLC's, the hybrid torque was predicted by the NN regression model. It was observed that the NN regression model has learned from the optimized cycles of DP. The predictions made by NN regression was similar to the optimized cycles of DP. The yellow box in Figure 4.16 shows the bucket-fill phase and a large amount of torque was delivered from the hybrid system during the bucket-fill phase. Hence the SOC of the accumulators was decreasing during bucket-fill phase, similar to the optimized results by DP cycle. Braking is represented by green rectangular box. During braking, energy was recuperated by the hybrid system, similar to the optimized cycles of DP, a negative torque can be observed and there was a slight increase in the SOC due to the recuperation of energy. Also, in comparison with the results of NN training in Figure 4.9 and 4.11, the hybrid torque request was always 0 Nm during idle condition.

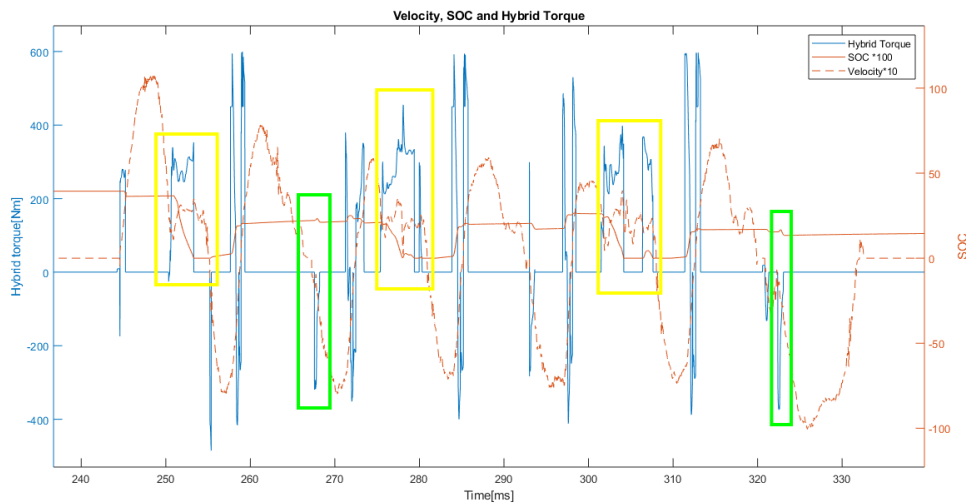


Figure 4.16: Comparison of Hybrid torque, SOC, Velocity obtained by integrating ML model in Simulink model

Figure 4.17 shows the graph of Vehicle velocity SOC and the Machine 2 pressure request for the three SLC's, the Machine 2 pressure was predicted by the NN regression model. The bucket-fill phase is highlighted by a yellow box. It can be observed that the SOC of the accumulators was decreasing during the bucket-fill phase and the SOC of the accumulators was utilized for the bucket-fill operation, similar to the optimized DP tool results 4.7. It can also be observed in the green box that during braking there was a slight increase in the SOC of the accumulators. During braking the energy was recuperated by Machine 1 and since the pressure in the accumulators

## 4. Results and Discussion

was increasing, the pressure on Machine 2 should also increase in order to further assist HFW, and hence we can observe that the Machine 2 pressure also increased during braking similar to optimized DP tool results shown in Figure 4.7. Since, a filter was used before the training of the Machine 2 pressure, the Machine 2 pressure data was a lot smoother than the optimized DP tool data. Also, in comparison with the results of NN training in Figure 4.13 and 4.15, the Machine 2 pressure was always 0 bars during idle condition.

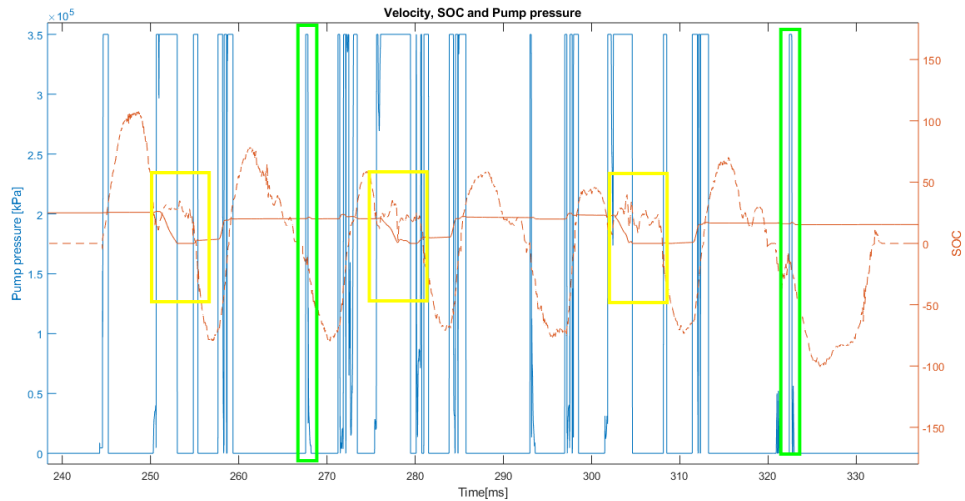


Figure 4.17: Comparison of Machine 2 pressure (pump pressure), SOC, Velocity obtained by integrating ML model in Simulink model

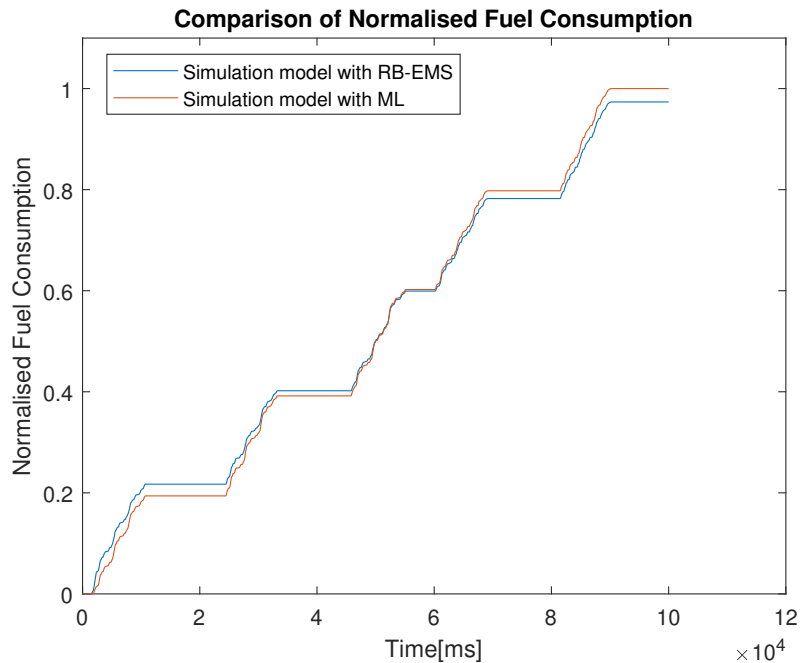


Figure 4.18: Comparison of normalised fuel consumption between simulation model with RB-EMS and simulation model with online ML.

Figure 4.18 shows comparison of the normalized fuel consumption between the simulation model with RB-EMS [4] and simulation model with online ML for 5 SLC. There was no significant difference in fuel consumption between the two models. The fuel consumption of the model with online ML is marginally higher than model with RB-EMS [4]. This might be due to assumptions made while developing the DP tool, DP model developed was much simpler than the one built in Simulink model which in turn affects ML training as it was trained using DP results, the deployed online machine learning model might have predicted incorrectly for a few cases which it had not encountered during the training. The engine speed limit which was not calculated during all the hybrid phases is also a reason for reduced fuel efficiency.



# 5

## Conclusion

From the results and discussion, the following conclusions can be drawn:

The existing RB-EMS [4] can be replaced by NN which significantly improves computational time by 18% and reduces the complexity in the model. A simple MATLAB function consisting of trained NN is able to learn and replace functionality of existing RB-EMS. The NN implemented is quite flexible wherein, it can be trained offline if there is an update on driving cycles and is easily implementable on the Simulink model.

The online ML model deployed in the Simulink, might sometimes encounter untrained cases. Hence care should be taken during the training of the NN so that it does not over-fit or under-fit the cases. If a new cycle data is available, then the NN should be trained and the updated model should be deployed in the Simulink.

The results from the optimized DP tool shows promising results in efficient utilization of the SHHS. The NN regression model rightly learnt to predict hybrid torque by training on the optimized DP cycles. Although the NN regression model had an average accuracy in predicting the Machine 2 pressure, the predictions followed the trend of the optimized Machine 2 pressure in DP tool. Both the NN can be used to predict the hybrid torque and Machine 2 pressure in the model, as they show possibilities of improving fuel efficiency and reduce the cost of computation as they are very simple to implement.

The models in the DP and Simulink model are not exactly same, there are differences in model parameters and efficiency maps. There are also situations in Simulink model, where the NN encountered the untrained cases. Hence there is a significant deviation from the training to deployment of the ML models in the Simulink model. Having more training data might be helpful in better generalisation, but it will definitely improve the accuracy. Hence its recommended to collect more training data.

The optimization process in the DP depends on the model parameters. A recommendation is to improve the DP tool so that the model and parameter calculations matches with the simulation model. By doing so, we will be addressing the actual optimization problem in the model and can significantly lead to improvement in the fuel efficiency. Sufficient amount of feature data can be obtained from the improved DP tool, which will further improve the training accuracy of the NN regression models.

**Q1:** Will the ML strategy be able to learn and replace the functionality of Rule Based Energy Management Strategy (RB-EMS)?

Yes, the ML strategy is able to learn the transition rules and replace the functionality of RB-EMS, eventually reducing the computational cost by 18%.

**Q2:** Which ML strategies are efficient and are used to replace different control systems?

Neural network for multi-class classification model and SVM for multi-class classification are suitable to replace the RB-EMS. But the Neural network for multi-class classification model was used in the model, because it is simple in composition and the computation cost to train is less than that of SVM model.

Neural network regression model is suitable for learning the values of hybrid torque and Machine 2 pressure obtained from DP tool. These trained neural networks are suitable for replacing the rule based system used for calculating the hybrid torque and pump pressure.

**Q3:** Does implementation of ML technique using optimized results of DP have the potential of improving fuel efficiency?

Yes, the neural network regression models can accurately predict the hybrid torque and Machine 2 pressure, similar to the results of DP tool. By building the DP model similar to the Simulink model and by deploying the neural network regression models trained on results of the improved DP tool, there is a potential of improving fuel efficiency.

**Q4:** What are the challenges involved in the implementation of online ML in the existing energy management system?

The data used for training plays a key role in the implementation of online ML. The training data should cover all cases that will possibly happen, this helps the ML model to predict accurately in all scenarios. Since the training data did not cover all the driving scenarios, it was a major challenge faced during online implementation, some cases involved extreme scenarios like very low SOC which was not covered during training, by addressing this problem there is a possibility of improving fuel efficiency.

# Bibliography

- [1] Wei Zhang, Jixin Wang , Shaofeng Du, Hongfeng Ma, Wenjun Zhao 2 and Haojie Li. MDPI . Energy Management Strategies for Hybrid Construction Machinery: Evolution, Classification, Comparison and Future Trends. 27 May 2019.
- [2] Rydberg, K. 2009. Energy Efficient Hydraulic Hybrid Drives. The 11th Scandinavian International Conference on Fluid Power, SICFP'09, June 2, 2009, Linköping, Sweden.
- [3] Uebel, K., Raduenz, H., Krus, P., de Negri, VJ. 2018. Design Optimisation Strategies for a Hydraulic Hybrid Wheel Loader. p. 1-3 Proceedings of the BATH/ASME 2018 Symposium on Fluid Power and Motion Control FPMC2018 September 12, 2018,Bath, UK.
- [4] Christopher Reichenwallner and Daniel Wasborg. "Control of a Hydraulic Hybrid System for Wheel Loaders". LIU-IEI-TEK-A-19/03400.SE, 2019.
- [5] Henrique Raduenz et al. "Energy Management Based on Neural Networks for a Hydraulic Hybrid Wheel Loader".
- [6] Guzzella, Lino and Sciarretta Antonio. "Vehicle Propulsion Systems: Introduction to Modeling and Optimization (Third Edition)".
- [7] Vjekoslav Tvrdić et al. "Hydraulic hybrid vehicle configurations and comparison with hybrid electric vehicle". CIET, 2018.
- [8] Martin Rohdin and Simon Öijwall. "Development of Control Strategies for a Hydraulic Hybrid Wheel Loader". LIU-IEI-TEK-A-16/02529.SE, 2016.
- [9] HOANG THAI DO, "Energy Management of Parallel Hydraulic Hybrid Wheel Loader". KTH, Stockholm, Sverige. 2018.
- [10] Siddharth Dadhich and Ulf Bodin. "Key challenges in automation of earth-moving machines", 2016.
- [11] "The Royal Society, Machine learning: the power and promise of computers that learn by example.",2017.[Online] Available: [www.royalsociety.org/machine-learning](http://www.royalsociety.org/machine-learning).
- [12] Machine Learning [Online] Available: <https://expertsystem.com/machine-learning-definition/#:~:text=Machine%20learning%20is%20an%20application,use%20it%20learn%20for%20themselves>.
- [13] Stuart Geman, Elie Bienenstock Rene Doursat. "Neural Networks and the Bias/Variance Dilemma" Massachusetts Institute of Technology, 1992.
- [14] Gulden Kaya Uyanik, Nese Guler."A study on multiple linear regression analysis" ELSEVIER,2013.
- [15] Iqbal H. Sarker, A. S. M. Kayes and Paul Watters. "Effectiveness analysis of machine learning classification models for predicting personalized

- context-aware smartphone usage". *Journal of Big Data*, Australia (2019). DOI:<https://doi.org/10.1186/s40537-019-0219-y>.
- [16] Sigmoid Function [Online] Available: [https://en.wikipedia.org/wiki/Sigmoid\\_function](https://en.wikipedia.org/wiki/Sigmoid_function).
- [17] Andrew Ng. "Machine Learning Course in Coursera".
- [18] "Machine Learning - Logistic Regression - Multi-class Classification". [Online] Available: <https://blog.csdn.net/iracer/article/details/50685873>.
- [19] Kevin Gurney. "An introduction to neural networks". University of Sheffield, 1997.
- [20] Neuron [Online] Available: <https://en.wikipedia.org/wiki/Neuron>.
- [21] Yongli Zhang. "Support Vector Machine Classification Algorithm and Its Application". Springer-Verlag Berlin Heidelberg, 2012.
- [22] SVM "Understanding Support Vector Machine(SVM) algorithm from examples (along with code)". [Online] Available: [https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/#:~:text=%E2%80%9CSupport%20Vector%20Machine%E2%80%9D%20\(SVM,mostly%20used%20in%20classification%20problems.&text=Support%20Vectors%20are%20simply%20the%20co%2Dordinates%20of%20individual%20observation](https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/#:~:text=%E2%80%9CSupport%20Vector%20Machine%E2%80%9D%20(SVM,mostly%20used%20in%20classification%20problems.&text=Support%20Vectors%20are%20simply%20the%20co%2Dordinates%20of%20individual%20observation).
- [23] Suzanna Becker. "Unsupervised learning procedures for neural networks". *International Journal of Neural Systems*, 1991.
- [24] Huda Hamdan Ali, Lubna Emad Kadhum. "K- Means Clustering Algorithm Applications in Data Mining and Pattern Recognition". *International Journal of Science and Research (IJSR)*, 2015.
- [25] Regression [Online] Available: <https://machinelearningmastery.com/classification-versus-regression-in-machine-learning/>.
- [26] Deep Learning [Online] Available: <https://medium.com/@rajatgupta310198/>.
- [27] Neumann K. (1993) Dynamic Programming Basic Concepts and Applications. In: Frauendorfer K., Glavitsch H., Bacher R. (eds) *Optimization in Planning and Operation of Electric Power Systems*. Physica, Heidelberg. [https://doi.org/10.1007/978-3-662-12646-2\\_2](https://doi.org/10.1007/978-3-662-12646-2_2)
- [28] Liermann, Matthias. "Backward simulation - A tool for designing more efficient mechatronic systems". 9th International MODELICA Conference, Munich, Germany, 2012.
- [29] Construction Equipment [Online] Available: <https://www.constructionequipment.com/wheel-loaders>.
- [30] Daniel Berrar. "Cross-validation". Researchgate, Tokyo. DOI: 10.1016/B978-0-12-809633-8.20349-X.
- [31] "Confusion Matrix for your Multi-Class Machine Learning Model". [Online] Available: <https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826>.
- [32] Data processing [Online] Available: [https://www.researchgate.net/post/how\\_to\\_normalize\\_data\\_using\\_matlab\\_How\\_do\\_you\\_now\\_use\\_the\\_normalized\\_data\\_set\\_to\\_forecast\\_using\\_ann\\_on\\_matlab](https://www.researchgate.net/post/how_to_normalize_data_using_matlab_How_do_you_now_use_the_normalized_data_set_to_forecast_using_ann_on_matlab).

- [33] Equation governing tansig activation function [Online] Available: <https://se.mathworks.com/help/deeplearning/ref/tansig.html>.
- [34] Purelin activation function [Online] Available: <https://se.mathworks.com/help/deeplearning/ref/purelin.html>.
- [35] Explanation for purelin activation function [Online] Available: <https://se.mathworks.com/matlabcentral/answers/455575>.

