PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH MOHAMED BOUDIAF UNIVERSITY - M'SILA

FACULTY OF TECHNOLOGY
DEPARTMENT OF ELECTRICAL ENGINEERING
N° :

FIELD: SCIENCE AND TECHNOLOGY
SECTOR: ELECTROMECHANIC
OPTION : ELECTROMECHANIC

Thesis presented for obtaining
Academic Master's degree
Presented by:

LAMARA Rabiha

Entitled:

Development of an intelligent diagnostic system for electromechanical systems

Publicly defended in the presence of the examining jury composed of:

Mr. Samir ZEGHLACHE	**Professor**	**President**
Mr. Djalaleddine KHODJA	**Professor**	**Supervisor**
Mr. Loutfi BENYETTOU	**Professor**	**Co-Supervisor**
Mr. Ali DJRIOUI	**Professor**	**Examiner**
Mr. Helali LALOUANI	**M.C.B**	**Examiner**

College year : 2023/202

Dedication:

To my beloved family, whose unwavering support and encouragement have been my greatest strength:

To my father and mother, your endless love, guidance, and sacrifices have paved the way for my success. I owe everything to your nurturing and wisdom.

To my three wonderful sisters and their families, your constant cheer and belief in me have been a source of inspiration. To my nieces and nephews, may you always chase your dreams with the same fervor and determination.

To my dear brother and his wife, your encouragement and faith in me have been invaluable. Thank you for always standing by my side.

To my friends, whose companionship and support have made this journey worthwhile. Your friendship means the world to me, and I am grateful for each one of you.

This work is dedicated to all of you, with love and

Acknowledgements:

I extend my deepest gratitude to Mr. Khodja Djalel-Eddine for proposing this topic and for his constant guidance, valuable suggestions, and knowledge that have significantly contributed to and facilitated the completion of this thesis. It has been an honor to be mentored by him, and I greatly respect his contribution to my education.

I also thank the jury members for examining and evaluating this humble work.

My heartfelt thanks go to all the teachers at the university who have contributed to my education, particularly those involved in my electromechanical training.

Lastly, I would like to express my warmest thanks to everyone who has supported me, directly or indirectly, in the realization of this thesis.

Abstract:

This project develops an intelligent fault diagnosis system for asynchronous motors using neural networks and decision trees, implemented on FPGA circuits. Neural networks use multilayer perceptrons with backpropagation to classify faults from RMS voltage inputs. Decision trees provide interpretable and efficient fault detection. The FPGA implementation ensures rapid, real-time fault diagnosis. The study demonstrates improved fault detection accuracy and responsiveness, highlighting the potential of combining AI algorithms with FPGA technology to enhance the reliability of electromechanical systems.

ملخص:

يهدف هذا المشروع إلى تطوير نظام ذكي لتشخيص الأعطال في المحركات غير المتزامنة باستخدام الشبكات العصبية وأشجار القرار، المطبق على الدوائر المتكاملة القابلة للبرمجة (FPGA). تستخدم الشبكات العصبية شبكات متعددة الطبقات مع خوارزمية التغذية العكسية لتصنيف الأعطال من قيم الجهد الفعالة (RMS). توفر أشجار القرار اكتشافًا للأعطال بكفاءة ووضوح. يضمن تطبيق النظام على دوائر الـ FPGAتشخيص الأعطال بسرعة وفي الوقت الحقيقي. توضح الدراسة تحسين دقة اكتشاف الأعطال وسرعة الاستجابة، مما يبرز إمكانيات الجمع بين خوارزميات الذكاء الاصطناعي وتقنية الـ FPGAلتعزيز موثوقية الأنظمة الكهروميكانيكية.

Résumé:

Ce projet développe un système intelligent de diagnostic des pannes pour les moteurs asynchrones en utilisant des réseaux neuronaux et des arbres de décision, implémentés sur des circuits FPGA. Les réseaux neuronaux utilisent des perceptrons multicouches avec rétropropagation pour classifier les pannes à partir des valeurs RMS des tensions. Les arbres de décision offrent une détection des pannes interprétable et efficace. L'implémentation sur FPGA assure un diagnostic des pannes rapide et en temps réel. L'étude démontre une amélioration significative de la précision de détection des pannes et de la réactivité, soulignant le potentiel de la combinaison des algorithmes d'IA avec la technologie FPGA .pour améliorer la fiabilité des systèmes électromécaniques

Table of content

Table of content

List of figures:

LIST OF FIGURES: VI

List of tables

List of tables:

List of symbols

List of symbols:

R_s	Resistance of stator’s winding	[□]
L_{sf}	Leakage inductance of stator	[H]
M_{ss}	Mutual inductance between two stator phases	[H]
M_{SR}	Mutual inductance between a stator phase and a rotor phase	[H]
M_{RS}	Mutual inductance between a rotor phase and a stator phase	[H]
f_{sa} ,f_{sb}, f_{sc}	short circuit coefficient	-
M_s	Mutual inductance between two stator phases	[H]
M	Inductance mutual between phase and stator	[H]
θ	Angular position between stator and rotor	[°]
φ_S	Stator flow	Weber
I_S	Statoric current	[A]
U_s	Statoric tension	[V]
$\mathrm{W_i}$	The strength of the connection simulating the synaptic weights of the neurons	-
T_{pi}	The desired output (Target)	-
$\mathrm{O_{pi}}$	Network output	-
S_{pi}	The error committed at the output of neuron i for the example P	-
F'i (Opi)	the derivative of the activation function of neuron i	-
W	Synaptic weight vector	-

List of abreviations

List of abbreviations:

AI	**Artifitial intelligence**
IM	**Induction Motor.**
IEEE	**the Institute of Electrical and Electronics Engineers.**
EPRI	**Electric Power Research Institute.**
ABB	**sigle d'ASEA Brown Boveri.**
ANNs	**Artifitial Neural Network.**
CNNs	**Convolutional Neural Network.**
ID3	**Interactive Dichotomizer 3.**
CART	**Classification And Regression Trees.**
CHAID	**Chi-square Automated Interaction Detection.**
MARS	**Multi Variate Adaptive Splines.**
NN	**Neural Network.**
LIME	**Local Interpretable Model-agnostic Explanations.**
RMSE	**Root Sean Square Error.**
TPR	**True Positive Rate.**
FNR	**False Negative Rate.**
ROC	**Receiver Operating Characteristic.**
AUC	**Area Under Curve.**
C4.5	**abbreviation refers to a decision tree algorithm in machine learning.**

General introduction

General introduction:

The advancement of intelligent diagnostic systems for electromechanical systems, such as induction motors, represents a significant leap in the field of industrial automation and maintenance. These systems aim to enhance the reliability, efficiency, and predictive maintenance capabilities of critical machinery, thereby reducing downtime and operational costs.

The core of this research lies in the integration of advanced artificial intelligence (AI) techniques with traditional electromechanical models. Induction motors, which are widely used in various industrial applications due to their robustness and efficiency, present complex diagnostic challenges. Fault detection and performance optimization in these motors require sophisticated approaches that can handle non-linearities and intricate dynamic behaviors.

In this context, neural networks and decision trees emerge as powerful tools for developing intelligent diagnostic systems. Neural networks, with their ability to learn from vast amounts of data and model complex patterns, offer a robust solution for predicting faults and diagnosing issues in real-time. Decision trees, on the other hand, provide a transparent and interpretable framework for classification and regression tasks, making them valuable for diagnostic applications where understanding the decision-making process is crucial.

This thesis explores the mathematical modeling of induction motors and the integration of AI techniques for diagnostic purposes. By implementing these models in simulation environments like Simulink, we can visualize and analyze motor behavior under various operating conditions. The comparative study between neural networks and decision trees provides insights into the strengths and limitations of each method, guiding the selection of appropriate AI techniques for specific diagnostic tasks.

Ultimately, the goal of this research is to develop a comprehensive diagnostic system that leverages the predictive power of AI to enhance the operational efficiency and reliability of electromechanical systems. The findings from this study not only contribute to the academic understanding of intelligent diagnostics but also pave the way for practical applications in industrial maintenance and automation.

Chapter 1:

State of art of induction machine fault diagnosis

I.1 Introduction:

The development of an intelligent diagnostic system for electromechanical systems necessitates a thorough understanding of the theoretical principles underlying both the electromechanical components and the artificial intelligence techniques employed. This chapter serves as the foundation for our research, providing essential definitions, mathematical models, and theoretical frameworks that are critical for the subsequent development and implementation of our diagnostic system.

We begin by defining key concepts related to induction motors, including their operational principles and the mathematical equations that describe their behavior. Understanding these fundamentals is crucial for modeling the dynamic performance of induction motors accurately, which is essential for effective fault diagnosis and predictive maintenance.

Following this, we delve into the theoretical aspects of the AI methodologies used in this study: neural networks and decision trees. We provide detailed definitions and explanations of their structures, learning algorithms, and their respective strengths and weaknesses in the context of diagnostic applications. Neural networks are highlighted for their ability to model complex, non-linear relationships through layers of interconnected nodes, while decision trees are noted for their interpretability and efficiency in classification and regression tasks.

The integration of these AI techniques with the mathematical models of induction motors forms the core of our diagnostic system. We discuss how these models are implemented in simulation environments, such as Simulink, to enable dynamic analysis and real-time fault detection.

This theoretical foundation is essential for understanding the complexities involved in developing an intelligent diagnostic system. By establishing a clear understanding of both the electromechanical and AI components, this chapter sets the stage for the practical implementation and results that will be discussed in the subsequent chapters.

I.2 Induction motors:

three-phase induction motors are the motors most frequently encountered in industry. They are simple, rugged, low-priced, and easy to maintain. They run at essentially constant speed from zero to full load. The speed is frequency-dependent and, consequently, these motors are not easily adapted to speed control. However, variable-frequency electronic drives are being used increasingly to control commercial induction motors' speed. [1]

But that's not a guarantee against engine failure, which is why diagnostic systems are in place to prevent breakdowns.

I.3 Structure of induction motor:

A 3-phase induction motor has two main parts: a stationary stator and a revolving rotor. The rotor is separated from the stator by a small air gap ranging from 0.4 mm to 4 mm, depending on the motor's power. The *stator* consists of a steel frame that supports a hollow, cylindrical core made up of stacked laminations. Several evenly spaced slots punched out of the internal circumference of the laminations, provide the space for the stator winding. The rotor is also composed of punched laminations. These are carefully stacked to create a series of rotor slots to provide space for the rotor winding.[1]

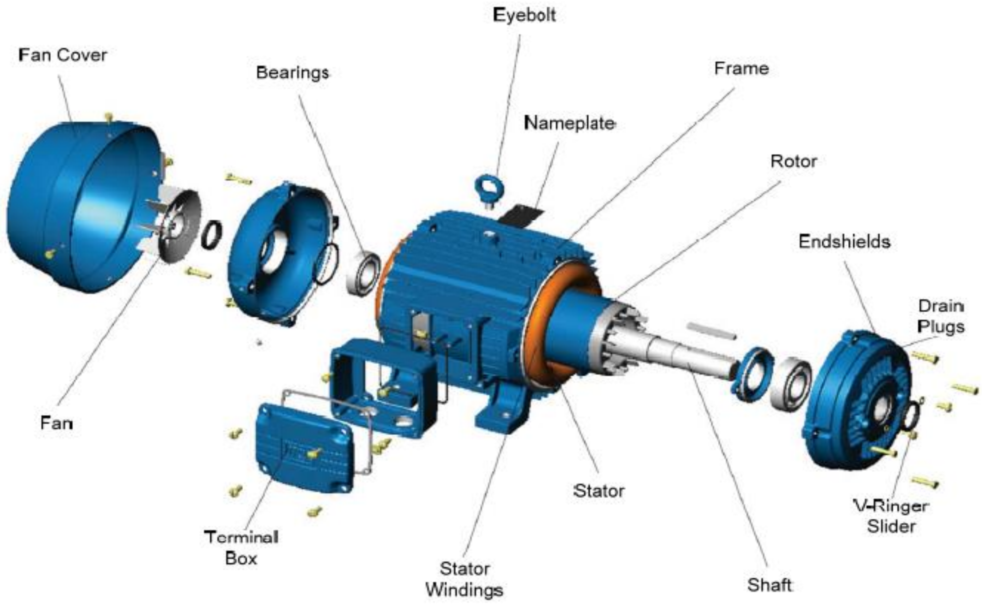


Figure.I.1A typical 3-phase induction motor

I.4 Faults of induction motor:

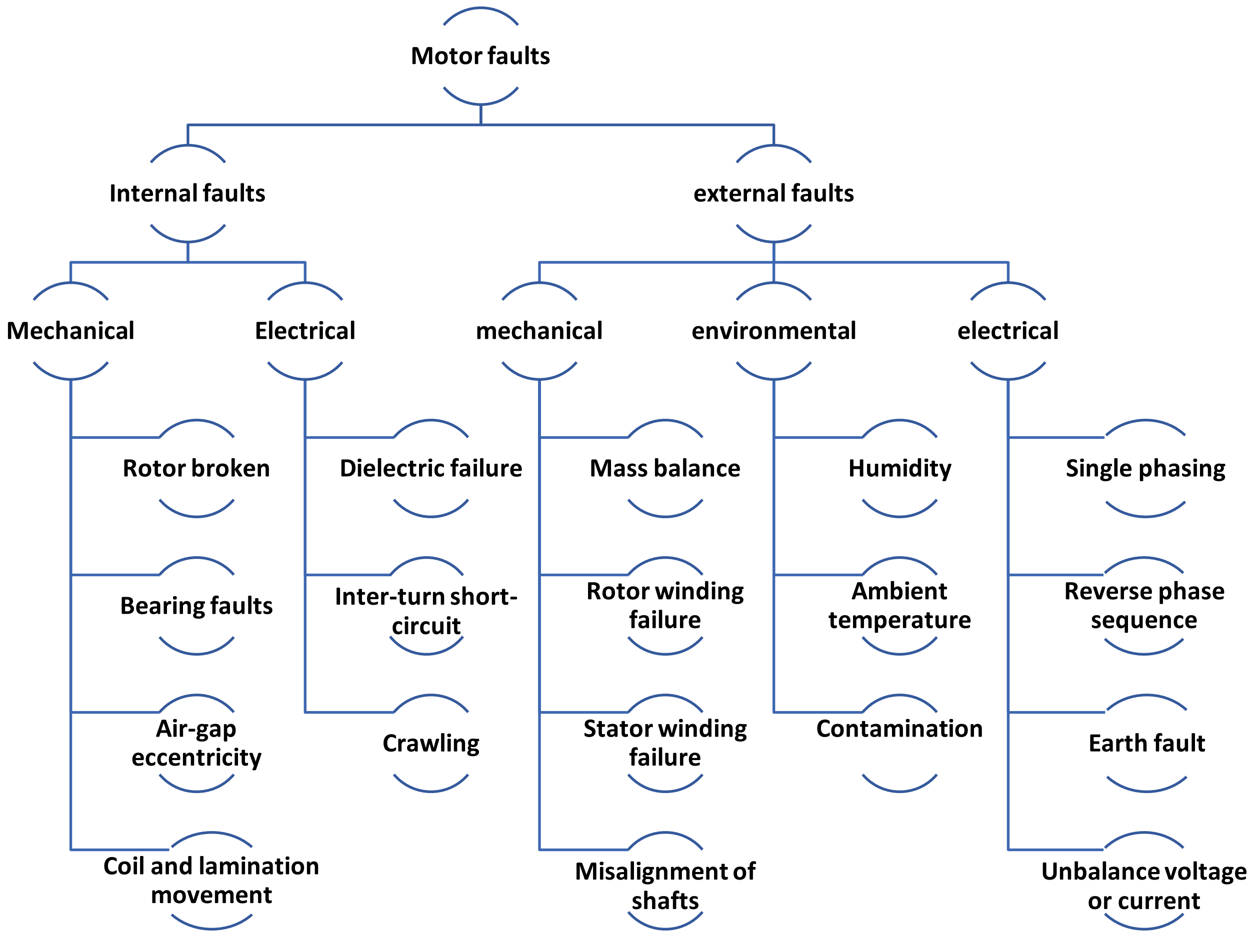


Figure.I.2 Classification of faults in induction motor

I.4.1 Mechanical faults:

the main cause of the fault has a mechanical nature.

I.4.1.1 Bearing faults:

The most common fault in IM is bearing failure. The bearing fault is one of the reasons for excessive vibration in the motor, the ultimate effects of bearing faults are rotor bar failures, which produce a premature breakdown of IMs.[2] The bearing fault occurring in IM during operation, according to IEEE and EPRI, are 41% and 42% respectively as shown in Figure.I.4

I.4.1.2 Eccentricity faults:

The unequal air gap that exists between stator and rotor is known as machine eccentricity. The eccentricity fault produces the problems of vibrations and noise.[3] The rotor fault which occur in IM during operation, according to EPRI and IEEE, are 8% and 9% respectively.

I.4.2 Electrical faults:

Any type of fault which has an electrical source is called an ‘‘electrical fault.’’.

I.4.2.1 Stator faults:

The most common faults in IM are stator inter-turn fault due to heavy current flow in the short-circuited coils and insulation downgrading [4-6]

As per the study by EPRI and IEEE, faults occur in the stator winding of IM are 36% and 28% respectively.

I.4.2.2 Rotor faults:

The rotor is the inner part of an IM which is made by bars of solid copper or aluminum that spans the rotor length and is connected through a ring at each end.[7] It could be susceptible to several faults. These faults are caused by rotor winding. The rotor faults are mainly broken rotor bars because of pulsating load and direct on-line starting.[3]

The rotor fault which occurs in IM during operation, according to EPRI and IEEE, are 8% and 9% respectively.

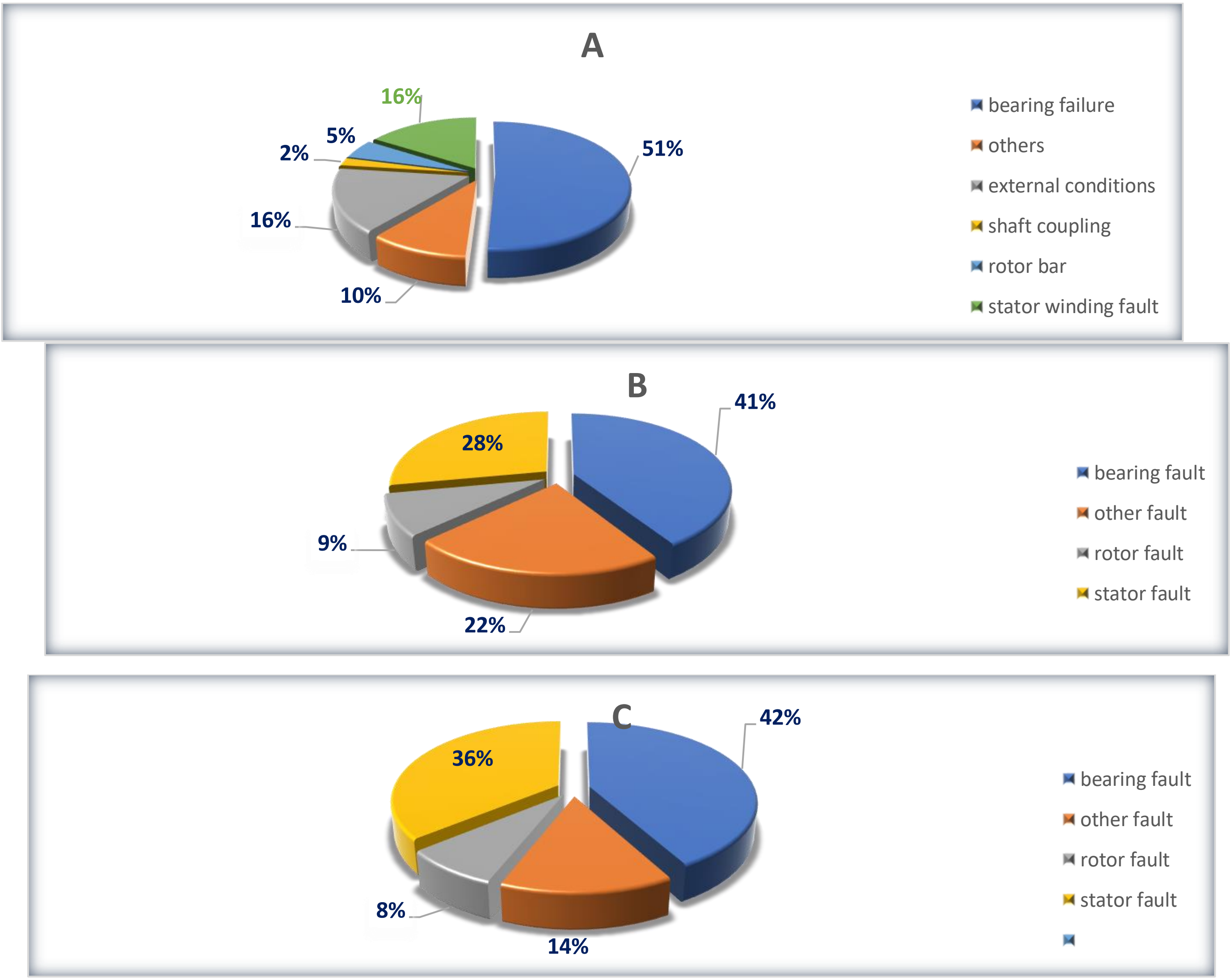


Figure I.3. Study on induction motor faults. **a** ABB, **b** IEEE, **c** EPRI[7]

I.5 Fault Diagnosis of induction motor:

Different research works are going on to study various types of motor faults at various research places. The researchers have introduced different fault detection techniques. Some methods are based on analysis of nonelectrical parameters. These include vibration signal analysis to detect electrical faults, analysis of air gap flux to detect motor eccentricity, and acoustic diagnosis technique to detect machine insulation faults. The vibration signal has been analyzed using a neural network for electrical fault detection, and thermal analysis of induction motor has been done to study the fault of an induction motor.[8]

I.6 Fault detection methods:

I.6.1 Model-based approaches:

The effect of particular faults on parameters such as the machine output current can be predicted using analytical [9] or finite element modeling approaches [10]

I.6.2 Trending:

This involves observing a fault parameter over a while so that it will be able to detect sudden changes that would be associated with the presence of faults. [10]

I.6.3 Fault thresholds:

The simplest fault detection algorithm is to use a threshold for a given parameter.

I.6.4 Multi-dimensional space:

Multiple fault parameters can be taken into account by representing each fault parameter as one dimension of a multiple-dimensional space. A given set of parameters corresponds to a point in this space. Points for healthy operation are located in different regions in this space from points for faulty operation.[3]

The support vector classification approach [11-12] tries to find a linear combination of parameters (geometrically represented by a hyperplane) that will separate the healthy data from the faulty data.

I.6.5 Fuzzy logic:

This involves making decisions based on classifying signals into a series of bands (fuzzy values) rather than simply as healthy or faulty based on a single threshold.[3]

I.6.6 Expert systems:

Expert systems seek to represent the knowledge of a human expert by defining a series of rules from which conclusions can be drawn.[10,13]

I.6.7 Artificial intelligence-based fault diagnosis:

The AI-based diagnostic system consists of signal-based methods and classification tools.[2] If properly established and effectively implemented, AI-based fault diagnosis can significantly reduce maintenance costs by reducing the number of unnecessary scheduled preventive maintenance operations [15,16,17]. It comprises five steps: data acquisition, feature extractions, feature selection, diagnosis of faults, and prognosis of systems, as shown in **Figure I.4.**

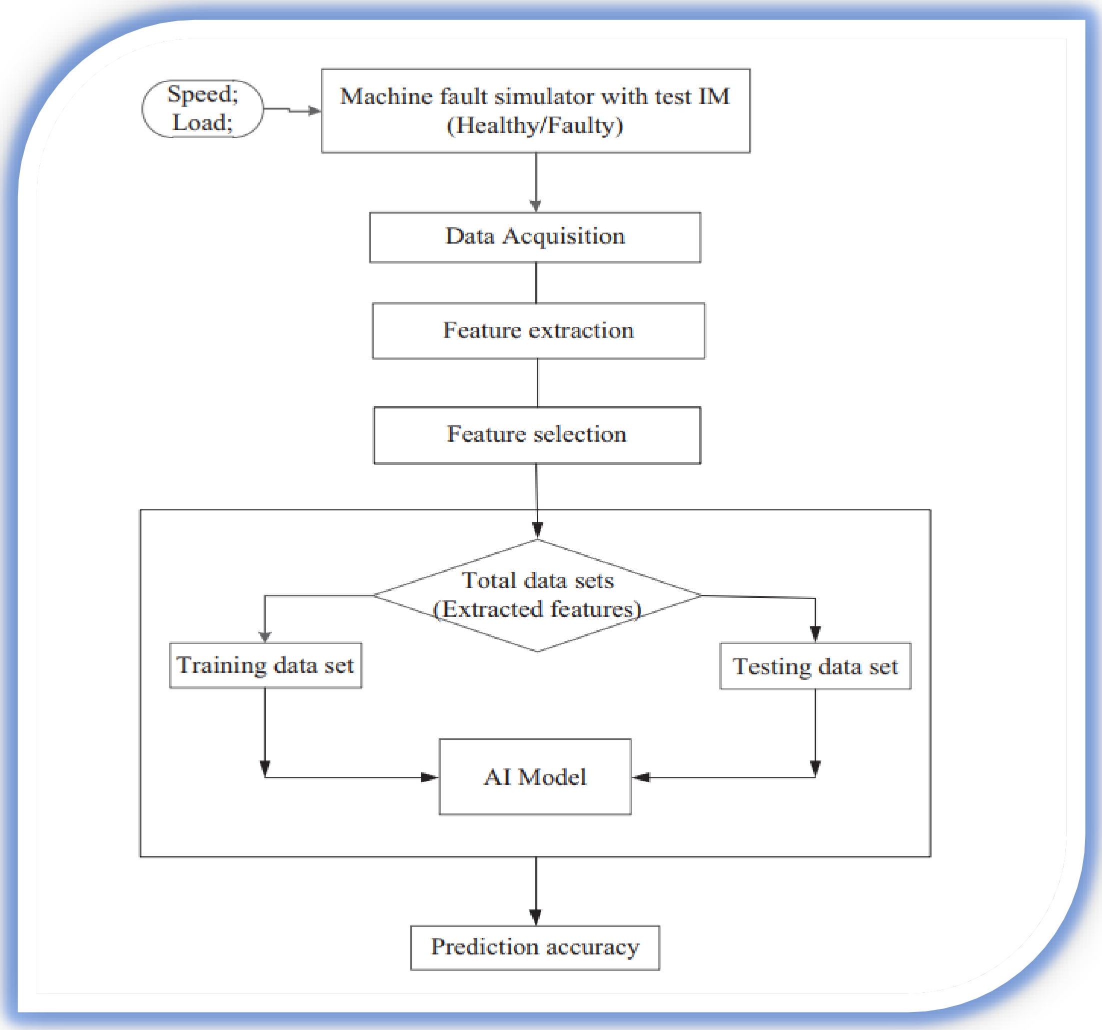


Figure I.4. AI-based fault diagnosis.

I.7 Artificial intelligence-based fault diagnosis:

I.7.1 Neural networks:

In recent times artificial neural networks (ANNs) has become popular and helpful model for classification, clustering, pattern recognition and prediction in many disciplines. The neural network can be artificially created, was first proposed by Walter Pitts and Warren McCulloch in paper entitled as ―A logical calculus of ideas immanent in nervous activity‖ in 1943.The model proposed is often referred to as McCulloch – Pitts. [18] They are modeled on the neural connections in the human brain. Each artificial neuron accepts several inputs, applies preset weights to each input, and generates a non-linear output based on the result. The neurons are connected in layers between the inputs and outputs [12-14]

I.7.1.1 Neural network architecture:[19]

In general, an artificial neural network can be divided into three parts, named layers, which are known as:

a) Input layer:

This layer is responsible for receiving information (data), signals, features, or measurements from the external environment. These inputs (samples or patterns) are usually normalized within the limit values produced by activation functions. This normalization results in better numerical precision for the mathematical operations performed by the network.

b) Hidden, intermediate, or invisible layers:

These layers are composed of neurons which are responsible for extracting patterns associated with the process or system being analyzed. These layers perform most of the internal processing from a network.

c) Output layer

This layer is also composed of neurons, and thus is responsible for producing and presenting the final network outputs, which result from the processing performed by the neurons in the previous layers. The main architectures of artificial neural networks, considering the neuron disposition, as well as how they are interconnected and how its layers are composed, can be divided as follows:

(i) single-layer feedforward network,(ii) multilayer feedforward networks, (iii) recurrent networks and (iv) mesh networks.

I.7.1.2 Types of neural networks:

There are many types of neural networks, including:

a) **Multi-layer perceptron:**

The multilayer perceptron is the most known and most frequently used type of neural network. On most occasions, the signals are transmitted within the network in one direction: from input to output. There is no loop, the output of each neuron does not affect the neuron itself.[20]

b) **Convolutional neural networks:**

are analogous to traditional ANNs in that they are comprised of neurons that self-optimise through learning. Each neuron will still receive an input and operate (such as a scalar product followed by a non-linear function) - the basis of countless ANNs. From the input raw image vectors to the final output of the class score, the entire of network will still express a single perceptive score function (the weight).The last layer will contain loss functions associated with the classes, and all of the regular tips and tricks developed for traditional ANNs still apply.

The only notable difference between CNNs and traditional ANNs is that CNNs are primarily used in the field of pattern recognition within images.[21]

c) **Feed-forward neural network:**

There are two main categories of network architectures depending on the type of connections between the neurons, “feed-forward neural networks” and “recurrent neural networks”. If there is no “feedback” from the outputs of the neurons towards the inputs throughout the network, then the network is referred as a “feed-forward neural network”. Otherwise, if there exists such a feedback, then the network is called a “recurrent neural network”.[22]

I.7.1.3 Neural networks algorithms:

a) Error Calculation Algorithms :

-backpropagation :

Backpropagation through time is a potent tool, with applications to pattern recognition, dynamic modeling, sensitivity analysis, and the control of systems over time, among others.[23] In backpropagation neural networks, the mathematical relationships between the various variables are not specified. Instead, they learn from the examples fed to them. In addition, they can generalize correct responses that only broadly resemble the data in the learning phase.[24]

b) Optimization Algorithms :

-Gradient Descent and its Variants:

Batch normalization is a recently popularized method for accelerating the training of deep-feed forward neural networks. Apart from speed improvements, the technique reportedly enables the use of higher learning rates, less careful parameter initialization, and saturating nonlinearities. The authors note that the precise effect of batch normalization on neural networks remains an area of further study, especially regarding their gradient propagation.[25]

c) Regularization Techniques:

The methods that prevent overfitting are called regularization methods. Most of them use prior knowledge about the data to decrease the model complexity in order to decrease the variance, but some of them, like data augmentation, directly decrease the variance without affecting complexity.[26]

- Noise injection:

Adding random noise, like data augmentation, can make the classifier more resilient to fluctuations in the input vector. It can be achieved by introducing noise before each iteration of the backpropagation algorithm.

- Dropout:

This method is very simple to implement and demonstrates a significant improvement on a large variety of tasks.[26]

-Batch Normalization:

One of the most recently proposed regularization methods is called *batch normalization*[27]. It assumes normalization of each layer neurons to have zero mean and unit variance over each training batch. It allows us to use much higher learning rates and saturating nonlinearities (such as sigmoid functions) without risk of divergence or being stuck in the saturated regime.[26]

d) Transfer Learning:

Transfer learning is a method used to transfer knowledge acquired from one task to resolve another. This procedure can help for example to improve accuracy or reduce the time of training.[28]

I.7.2 Decision tree:

Decision tree learning is a learning method that is very popular and widely used practically [29]. This method is a method that seeks to find functions that have a discrete value and are resistant to data that has errors (data noise) and can learn disjunctive expressions.[30]

I.7.2.1 Types of Decision Trees:

a) **Classification Trees :**

where the target variable is categorical and the tree is used to identify the "class" within which a target variable would likely fall into. Classification trees are very similar to regression trees, only that classification trees are used to predict a discrete category (qualitative response) rather than a numeric value (quantitative values). The input variables used for classification can be numerical or categorical variables.[31]

b) Regression Trees :

where the target variable is continuous and tree is used to predict its value.[31]

I.7.2.2 Algorithms for Building Decision Trees:

a) ID3 (Iterative Dichotomiser 3) :

ID3 is a supervised learning algorithm, builds a decision tree from a fixed set of examples. [34]The resulting tree is used to classify future samples. It determines the classification of objects by testing the values of the properties. It builds a decision tree for the given data in a *top-down* fashion, starting from a set of objects and a specification of properties. At each node of the tree, one property is tested based on maximizing information gain and minimizing entropy, and the results are used to split the object set.[33]

b) C4.5 :

C4.5 decision tree algorithm was proposed by J.R. Quinlan in 1993. It is a decision tree generation algorithm based on ID3 algorithm. [34] The decision tree grows using *Depth-first* strategy. The C4.5 algorithm considers all the possible tests that can split the data and selects a test that gives the best information gain. [35]C4.5 handles both categorical and continuous attributes to build a decision tree[38].

c) CART (Classification and Regression Trees):

CART stands for Classification and Regression Trees (Breiman *et al*., 1984). The splits are selected using the twoing criteria and the obtained tree is pruned by cost–complexity Pruning. CART can consider misclassification costs in the tree induction. An important feature of CART is its ability to generate regression trees. Regression trees are trees where their leaves predict a real number and not a class.[35]

d) CHAID (Chi-squared Automatic Interaction Detector):

The chi-squared automatic interaction detector (CHAID) algorithm is considered to be one of the most used supervised learning methods. which is a tool used to discover relationships between variables, a decision tree technique based on an adjusted significance test (Bonferroni test) [37-38]. It divides the respondents into several groups according to the relationship between the underlying variable and the dependent variable, and then each group into several groups.[39]

e) MARS (Multivariate Adaptive Regression Splines) :

This method is a nonparametric regression method that can accommodate additive and interaction effects between predictor variables.[40] MARS has been used for modeling pairs of data with continuous or categorical responses.[39]

I.7.2.3 Ensemble Methods Using Decision Trees

a) Random Forests and Gradient Boosting Trees:

According to Rokach (2010) bagging, random forest and boosting are machine learning ensembles designed to improve the accuracy of machine learning algorithms for statistical classification and regression. They are most commonly applied to decision tree methods as building blocks in the creation of very powerful predictive models.[30]

b) Extra Trees (Extremely Randomized Trees):

The Extra-Trees algorithm builds an ensemble of unpruned decision or regression trees according to the classical top-down procedure. Its two main differences with other tree-based ensemble methods are that it splits nodes by choosing cut-points fully at random and that it uses the whole learning sample (rather than a bootstrap replica) to grow the trees.[41]

I.8 Conclusion:

In this chapter, we established the theoretical foundations essential for the development of an intelligent diagnostic system for electromechanical systems. We began by defining the core concepts and mathematical models that describe the dynamic behavior of induction motors, providing a crucial understanding necessary for accurate fault diagnosis and predictive maintenance.

We then explored the theoretical underpinnings of the AI techniques utilized in this study: neural networks and decision trees. Detailed explanations of their architectures, learning algorithms, and application contexts highlighted their respective strengths and suitability for different diagnostic tasks.

By integrating these AI methodologies with the mathematical models of induction motors, we laid the groundwork for creating a robust diagnostic system capable of real-time analysis and fault detection. The theoretical insights provided in this chapter are pivotal for the practical implementation and simulation discussed in the following chapters.

This comprehensive exploration of both the electromechanical and AI components sets a solid foundation for the development and evaluation of our intelligent diagnostic system, ensuring a clear understanding of the complexities and challenges involved in this innovative approach.

Chapter II :

Application of AI techniques for electrical drive diagnostics

II.1 Introduction:

This chapter is focuses to the mathematical modelling and comparative analysis of two AI-based methods of identifying induction motor problems: namely, neural networks and decision trees. We commence by presenting the mathematical equations that define the dynamic behaviour of induction motors, which serve as the analytical basis for our diagnostic models.

We then detail the implementation of these models in Simulink, enabling dynamic simulation and integration of our AI algorithms. The core of this chapter compares the performance of neural networks and decision trees, both trained on historical motor data, in diagnosing faults and predicting motor performance.

The results section presents a comparative analysis of the effectiveness of each method, highlighting their respective strengths and weaknesses. This study demonstrates the potential of AI techniques in enhancing electromechanical system diagnostics, paving the way for future advancements in the field.

II.2 Three phase-model of Induction motor:

II.2.1 Mathematical model:[42]

the model of the induction motor is in the equations below.

II.2.1.1 The matrices:

The matrices $[R_s]$,$[L_{sf}]$, $[M_{ss}]$, $[M_{SR}]$ and $[M_{RS}]$depend of three coefficients and the inductances $f_{sa},\ f_{sb},\ f_{sc}$are given by the following expressions:

$$[L_{sf}] = \begin{bmatrix} f_{sa}^2 L_{sf} & 0 & 0 \\ 0 & f_{sb}^2 L_{sf} & 0 \\ 0 & 0 & f_{sc}^2 L_{sf} \end{bmatrix}$$

$$[M_s] = M_{ss} \begin{bmatrix} f_{sa}^2 & -f_{sa}f_{sb}/2 & -f_{sa}f_{sc}/2 \\ -f_{sa}f_{sb}/2 & f_{sb}^2 & -f_{sc}f_{sb}/2 \\ -f_{sa}f_{sc}/2 & -f_{sc}f_{sb}/2 & f_{sc}^2 \end{bmatrix}$$

$$[M_{SR}] = M \begin{bmatrix} f_{sa}\cos(\theta) & f_{sa}cos\left(\theta + \frac{2}{3}\pi\right) & f_{sa}cos\left(\theta - \frac{2}{3}\pi\right) \\ f_{sb}cos\left(\theta - \frac{2}{3}\pi\right) & f_{sb}\cos(\theta) & f_{sb}cos\left(\theta + \frac{2}{3}\pi\right) \\ f_{sc}cos\left(\theta + \frac{2}{3}\pi\right) & f_{sc}cos\left(\theta - \frac{2}{3}\pi\right) & f_{sc}\cos(\theta) \end{bmatrix}$$

With: $[M_{SR}] = [M_{RS}]^T$

The stator resistance matrix [Rs] can be presented as:

$$[R_S] = R_S \begin{bmatrix} f_{sa} & 0 & 0 \\ 0 & f_{sb} & 0 \\ 0 & 0 & f_{sc} \end{bmatrix}$$

II.2.1.2 The first equation:

$$[\varphi_S] = ([M_S] - [M_{SR}^S][M_R^S]^{-1}[M_{RS}^S])[I_S] + [M_{SR}^S][M_R^S]^{-1}[M_R^S] \quad (2.1)$$

II.2.1.3 The second equation:

$$[U_s] = [R_s][I_s] + ([M_S] - [M_{SR}^S][M_R^S][M_{RS}^S])P[I_S] + [M_{SR}^S][M_R^S]P[\varphi_R^S] \quad (2.2)$$

Explanation: Represent a three-phase motor model.

Relevance: The equations previously supplied are essential for modeling and diagnosing the performance of an induction motor because they capture the fundamental relationships that regulate the motor's electrical and magnetic behavior.

II.2.2 Simulink model:

II.2.2.1 Model Description:

The Induction Motor implements a three-phase induction motor. The block uses the three-phase input voltages to regulate the individual phase currents, allowing control of the motor torque or speed.

II.2.2.2 Diagram:

The Simulink model of the induction motor is in figure II.1:

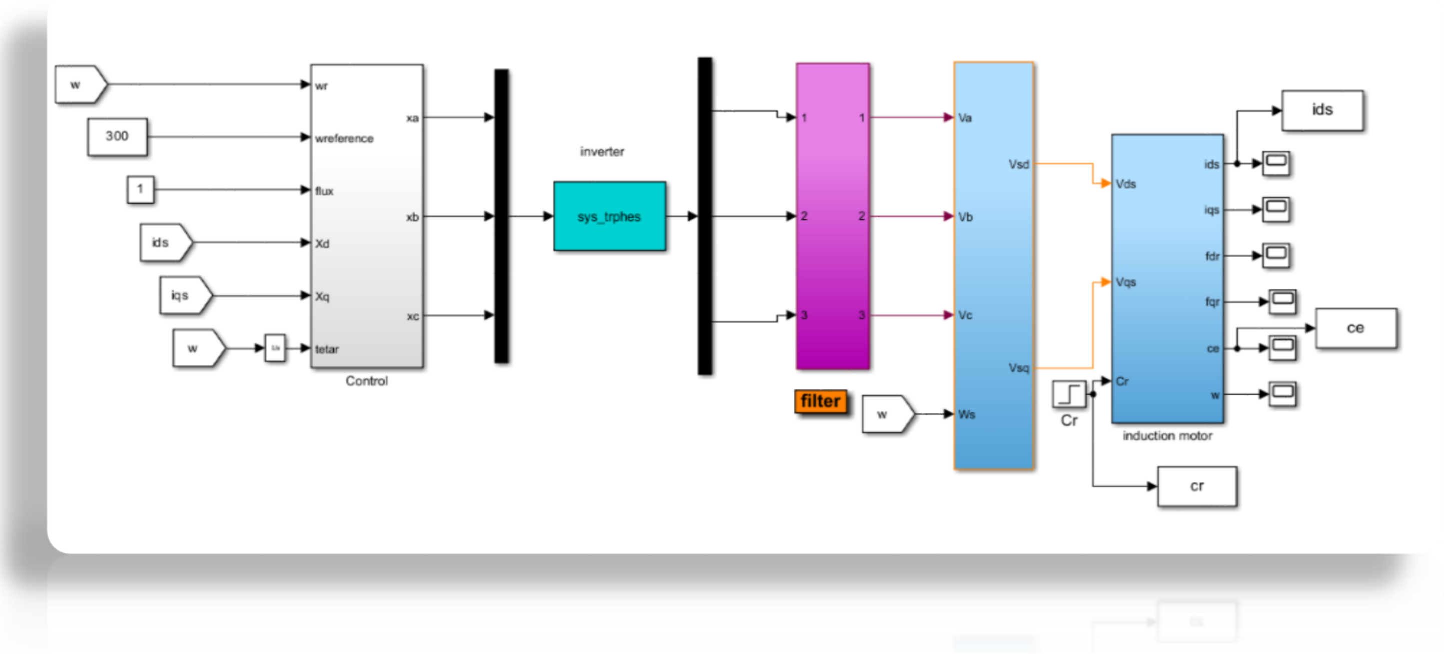


Figure.II.1. The Simulink model of the induction motor.

II.3 The neural network:

II.3.1 Algorithm Description:

II.3.1.1 Neural Network Architecture:

Regression model type: Trilayered Neural Network

Interpretability: hard

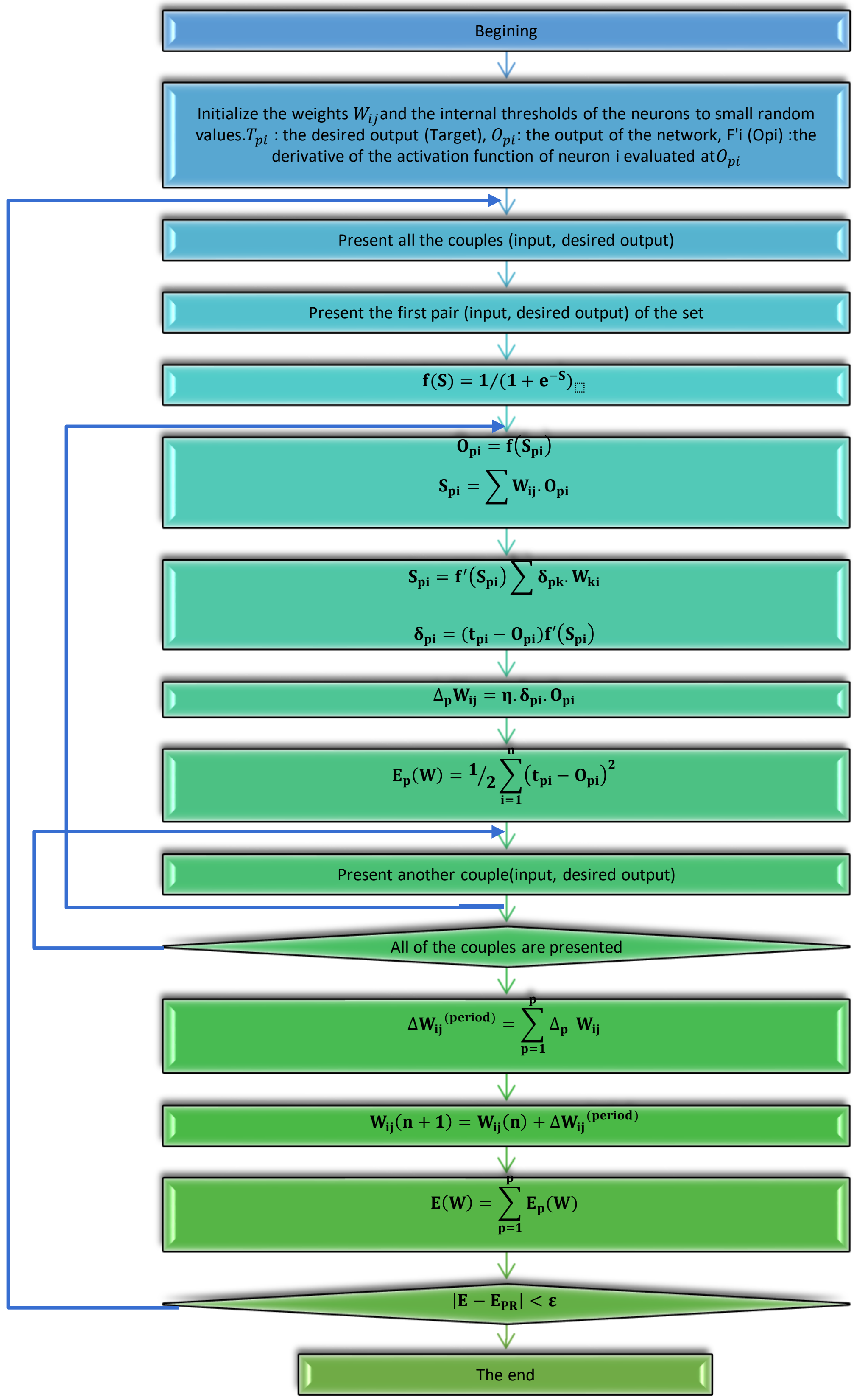
Model flexibility: High- increases with the **First layer size**, **Second layer size**, and **Third layer size** settings

the number of layers, neurons per layer, activation functions, etc.

II.3.1.2 Training Process:

The model is a feedforward, fully connected neural network for regression. The first fully connected layer of the neural network has a connection from the network input (predictor data), and each subsequent layer has a connection from the previous layer. Each fully connected layer multiplies the input by a weight matrix and then adds a bias vector. An activation function follows each fully connected layer, excluding the last. The final fully connected layer produces the network's output, namely predicted response values.[43]

II.3.1.3 Neural network algorithm: (back propagation)[42]



II.3.2 Simulink Implementation:

II.3.2.1 Integration with Induction Motor Model: These streamlined steps illustrate adding a neural network model to an induction motor in Simulink:

a) Neural Network Training:

-Data Collection: Collect motor performance under different conditions.

-Training: Train the neural network on for predicting motor performance, diagnosing issues using this data.

-Model Exportation: Train Neural Network: After training, export the trained neural network model from the training environment to a Simulink compatible format.

-Simulink Integration: Simulink Model: Include the model in Simulink as a sub-system block.

-Connect with Motor Model: Combine the neural network sub-system with the Simulink induction motor model.

-Run Simulations: Perform simulations on the integrated system to verify the ability to perform real-time diagnostic and control.

With this integration, it gets easier to monitor and control the performance of motor.

II.3.2.2 Neural network Diagram:

a) Classification neural network :

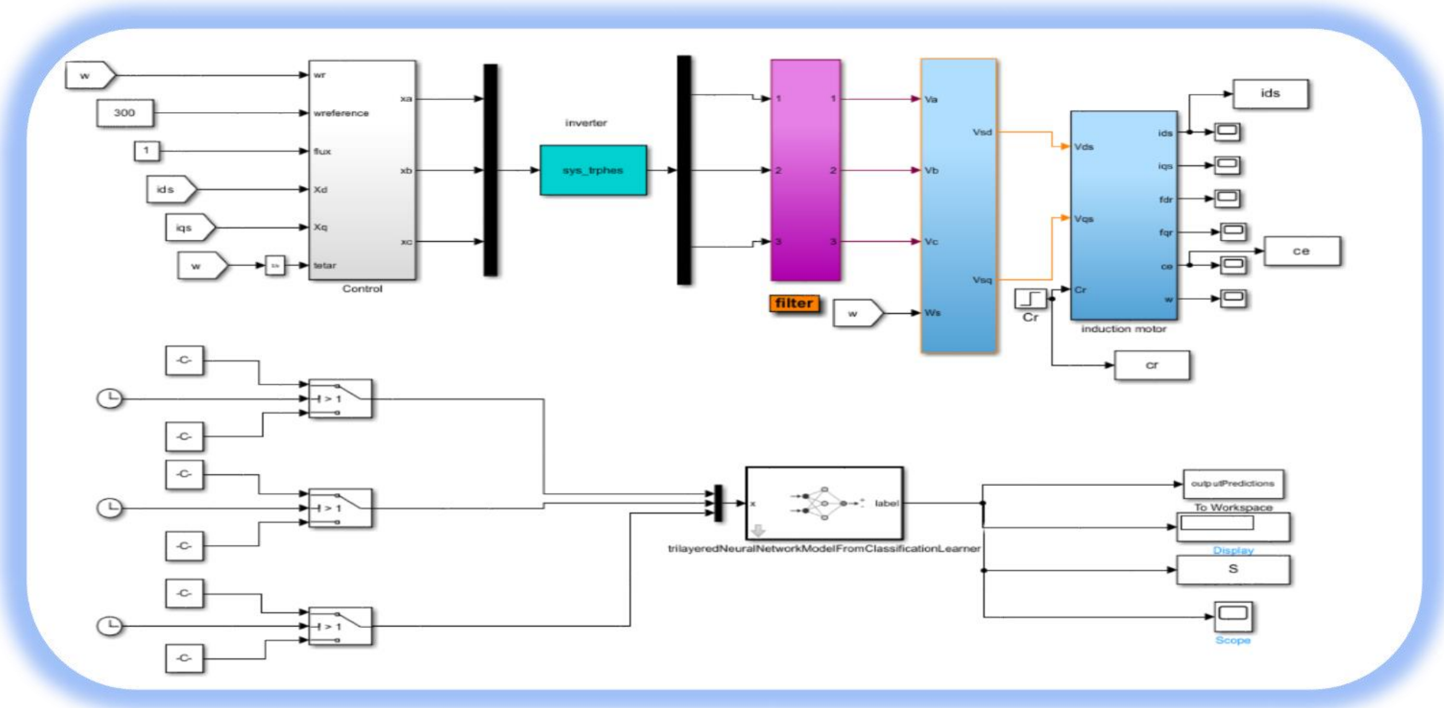


Figure II.2. Classification neural network model integrated with induction motor

b) Regression neural network:

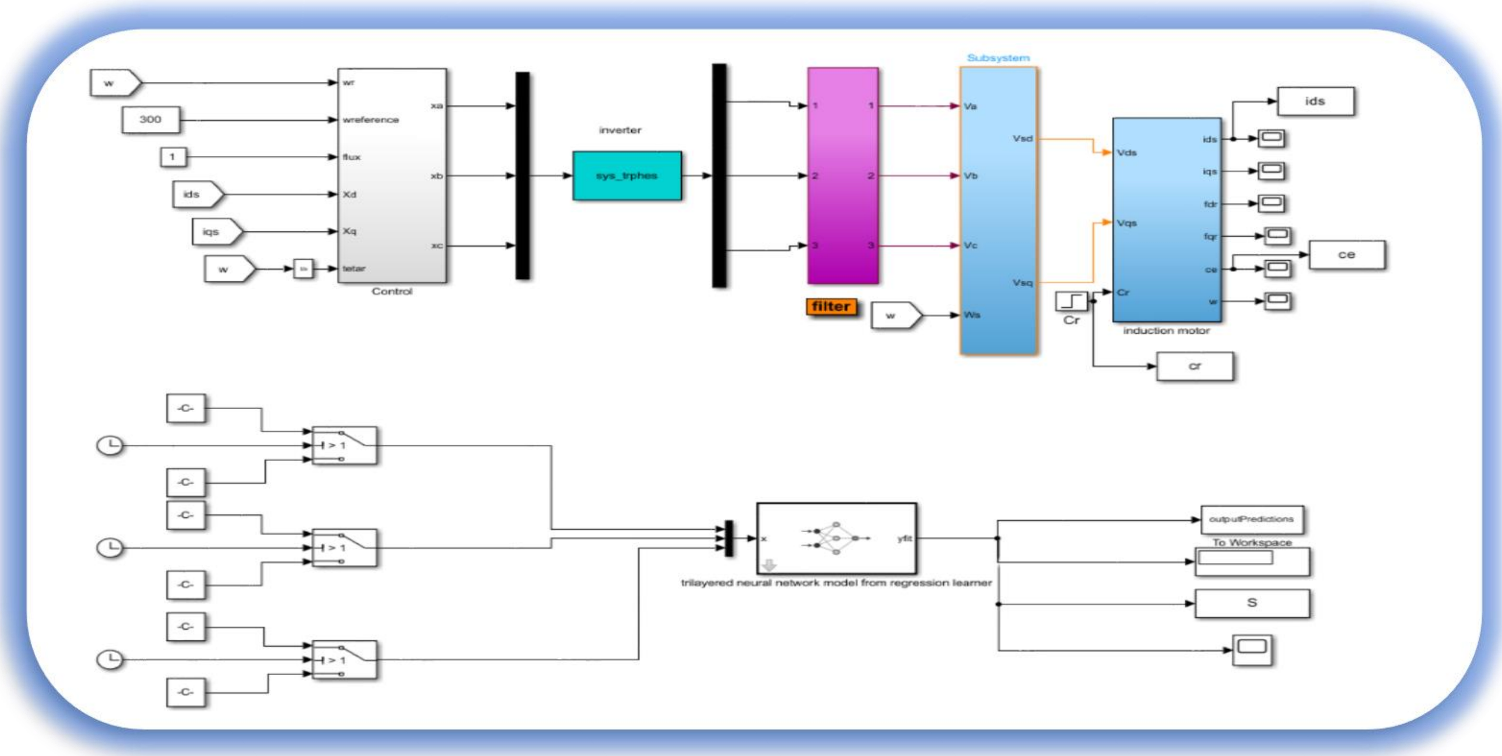


Figure II.3. Regression neural network model integrated with induction motor

II.3.3 Results:

II.3.3.1 Training and Testing Results:

a) Regression:

Training RMSE (validation): 0.53300 **Test RMSE (validation):** 0.35635

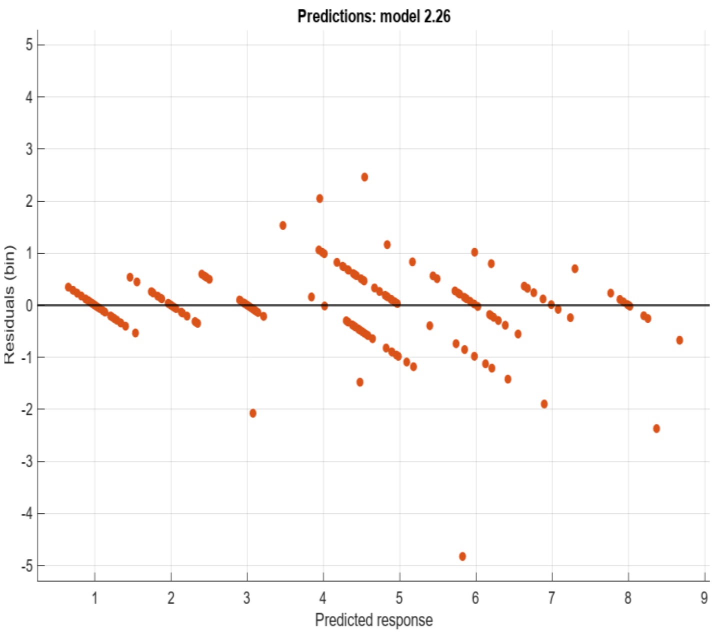


figure II.4.Regresssion trilayered NN Validation training residual plot markers predicted response

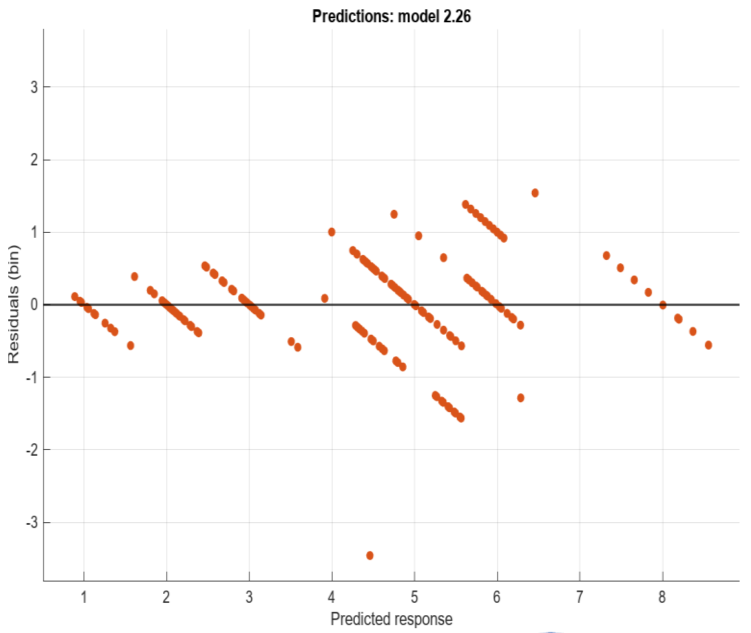


figure II.5.Regression trilayered NN validation test residual plot markers predicted response

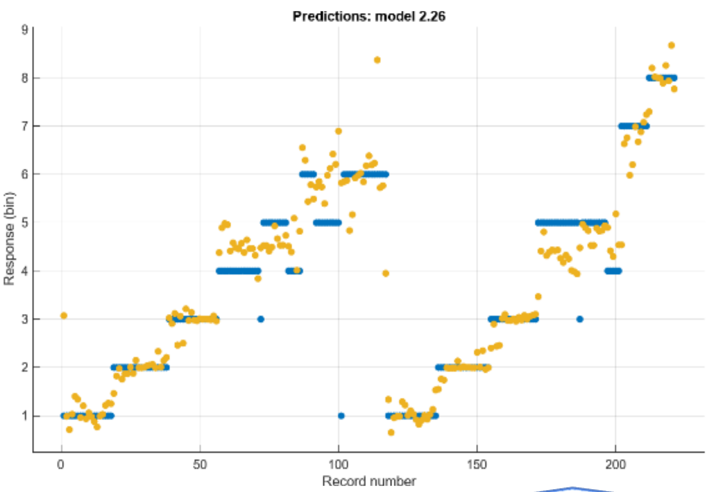


figure II.6.Regression trilayered NN training response plot

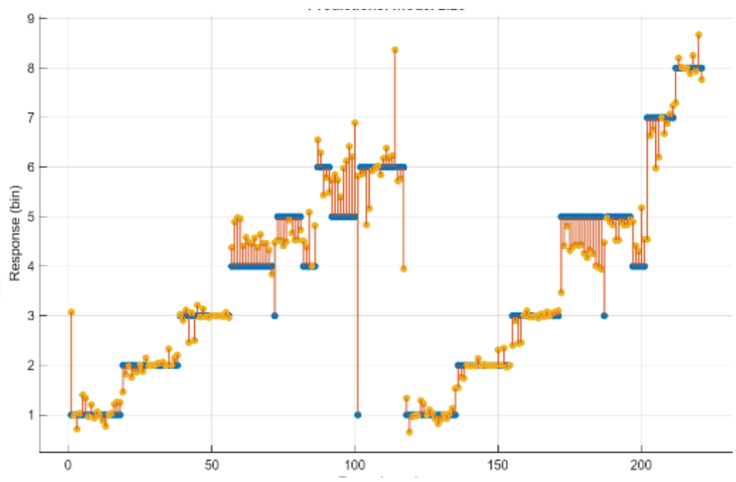


figure II.7.Regression trilayered NN training response plot with errors

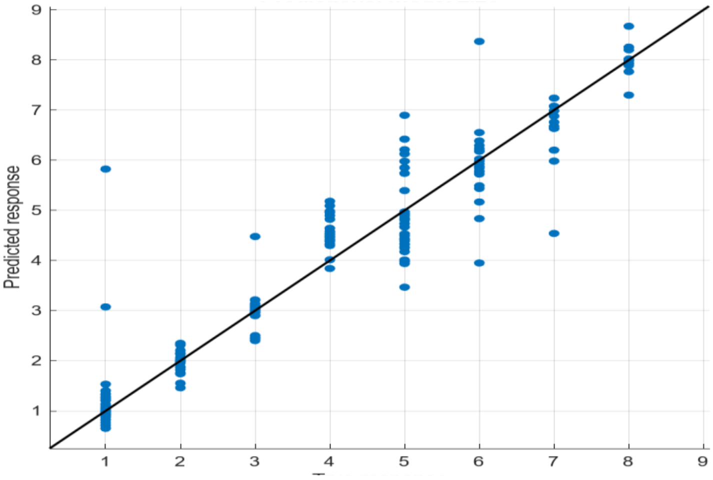


figure II.8.Regression trilayered NNtraining validation predicted vs actual plot

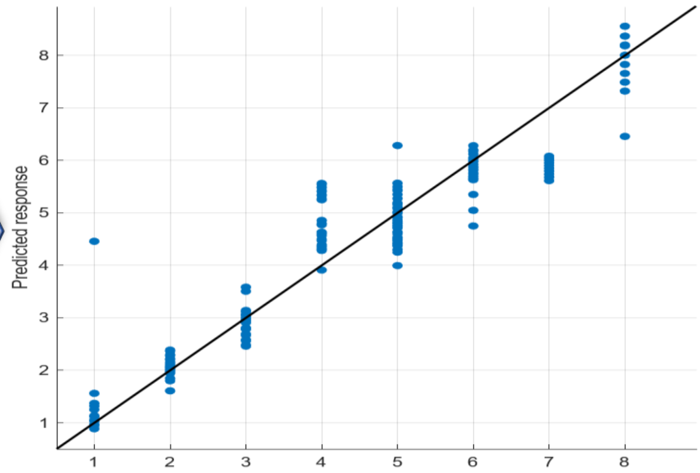


figure II.9.Regression trilayared NN test validation predicted vs actual plot

b) Explain :

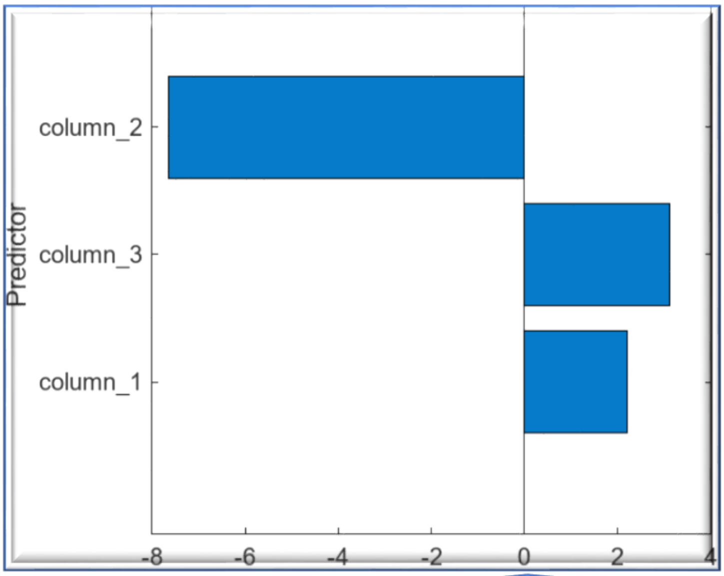


Figure II.10.Regression trilayered NN training shapley

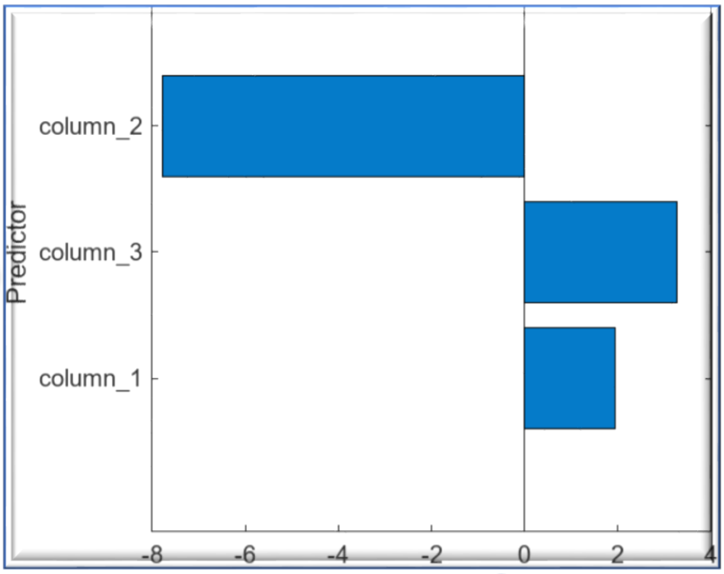


Figure II.11.Regression trilayered NN test shapley

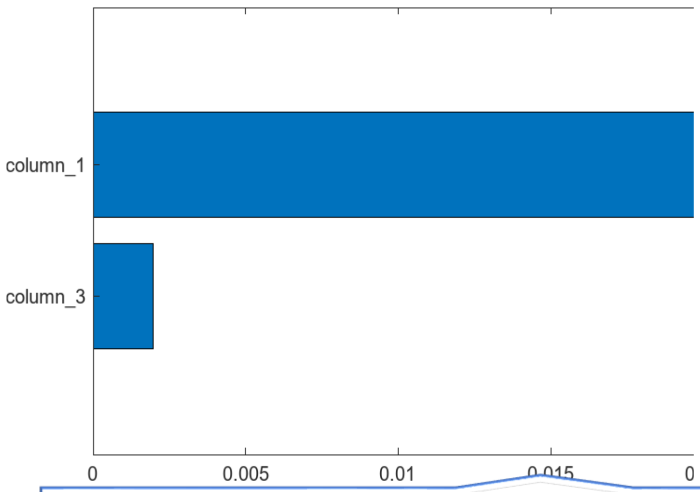


Figure II.12.Regression trilayered NN training lime

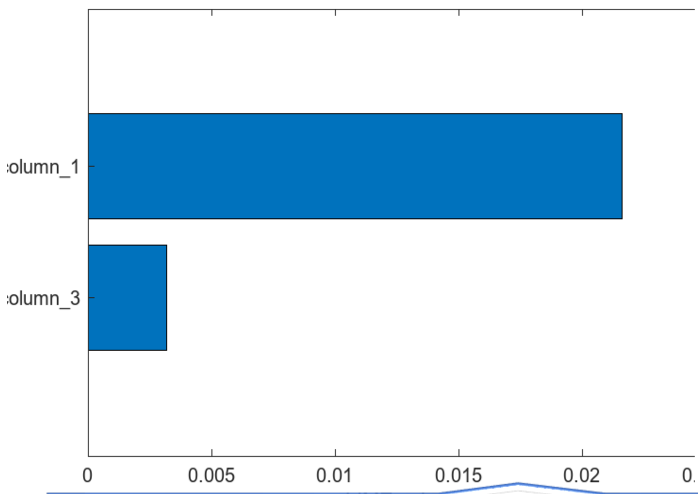


Figure II.13.Regression trilayered NN test lime

c) Classification :

Training RMSE (validation): 87.8%

Test RMSE (validation): 91.9%

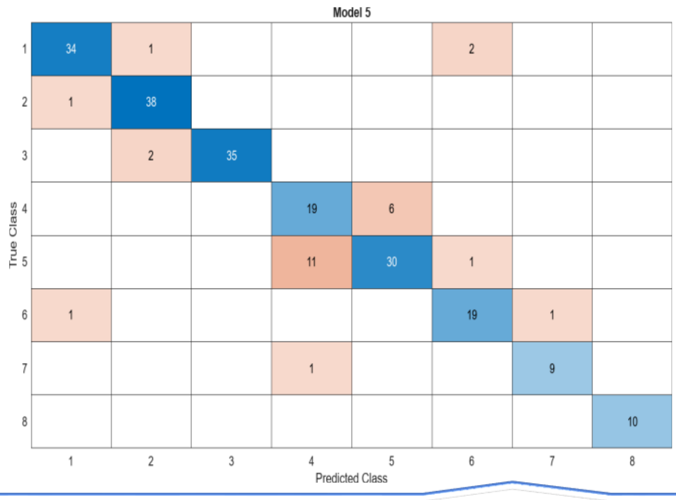


Figure II.14.Classification trilayered NN training validation confusion matrix number of observations

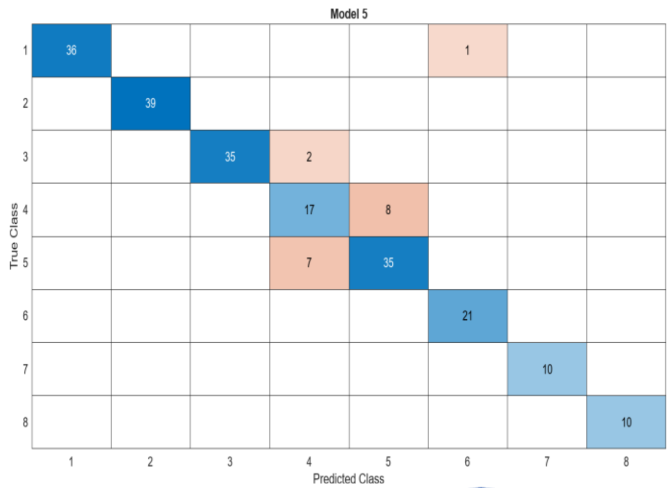


Figure II.15.Classification trilayered NN test validation confusion matrix number of observations

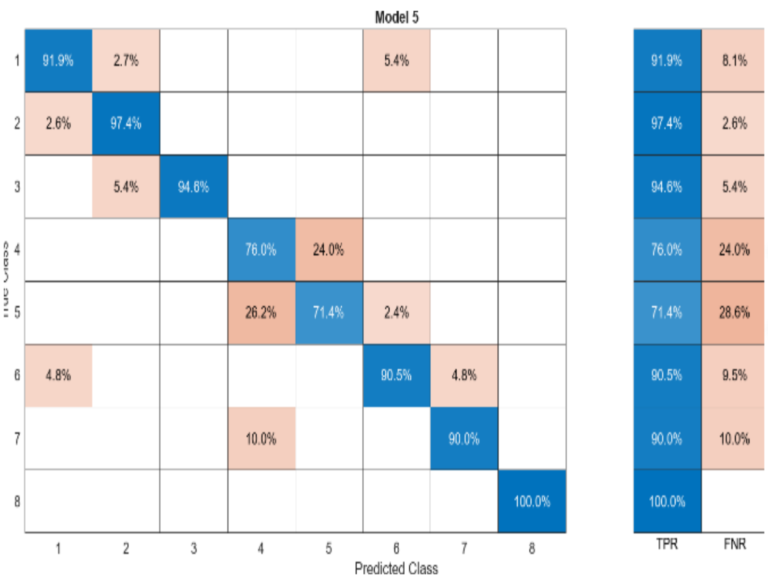


Figure II.16.Classification trilayered NN training validation confusion matrix TPR FNR

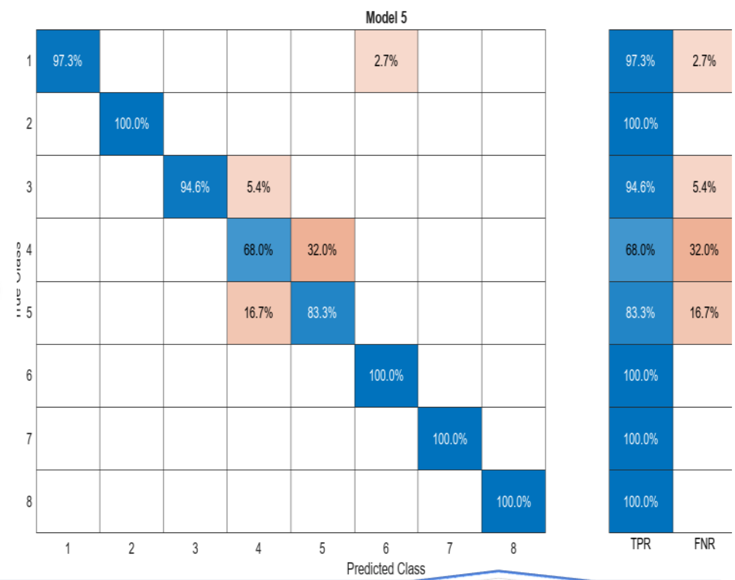


Figure II.17.Classification trilayered NN training validation confusion matrix TPR FNR

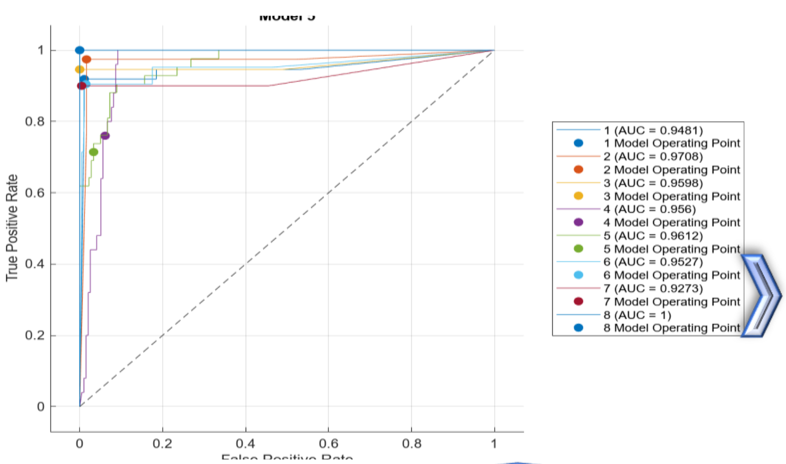


Figure II.18.Classification trilayered Validation training ROC curve

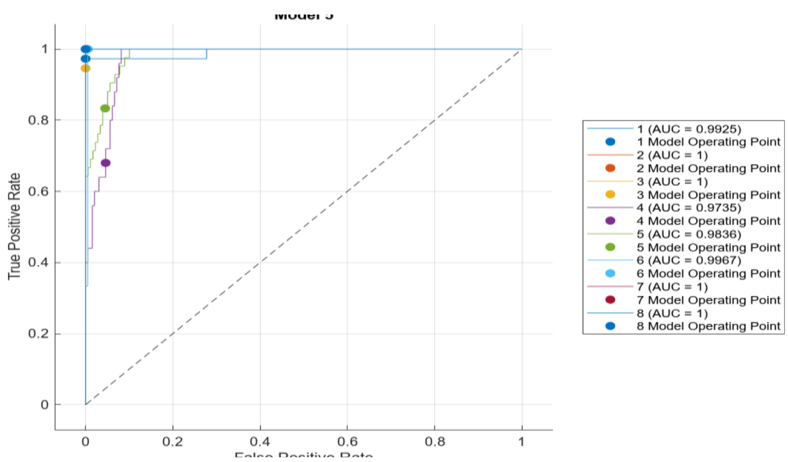


Figure II.19.Classification trilayered validation test ROC curve

d) Explain :

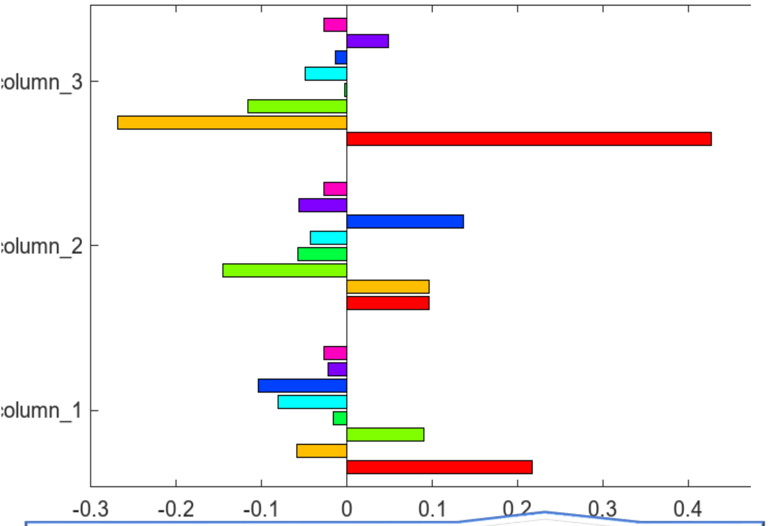


Figure II.20.Classification trilayered NN training local shapely

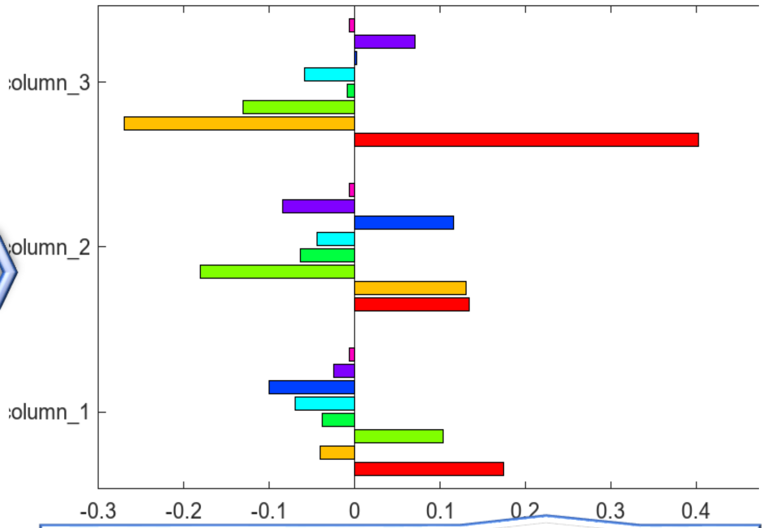


Figure II.21.Classification trilayered NN test local shapely

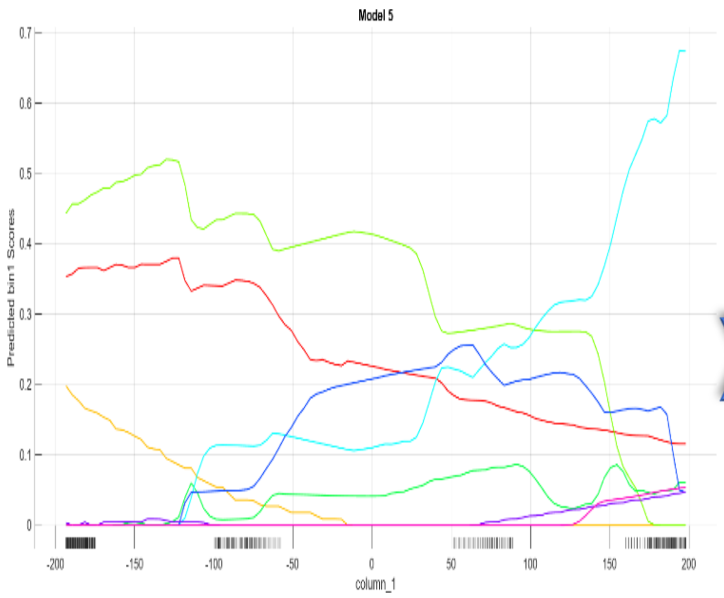


Figure II.22.Classification trilayered NN training partial depedence plot

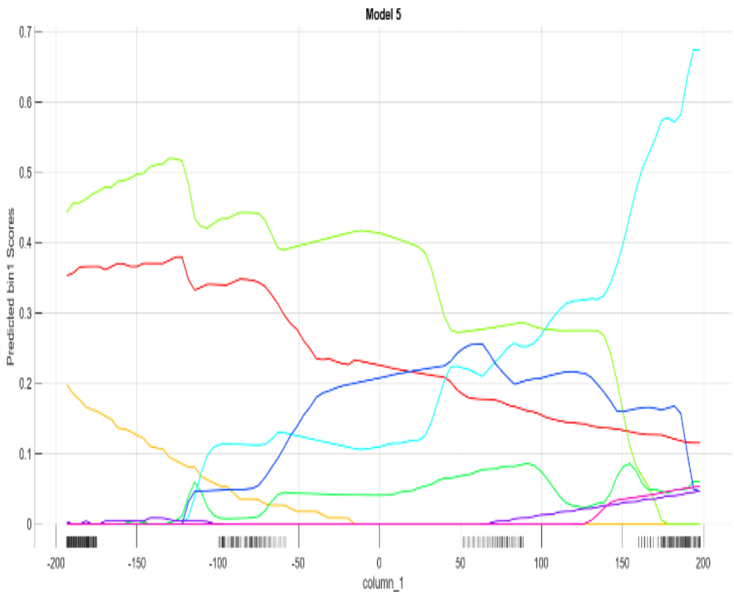


Figure II.23.Classification trilayered NN test partial dependence plot

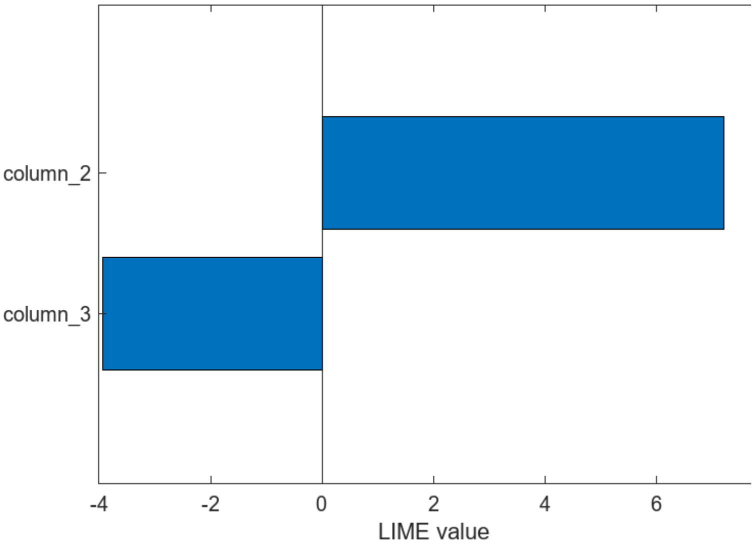


Figure II.24.Classification trilayered training lime

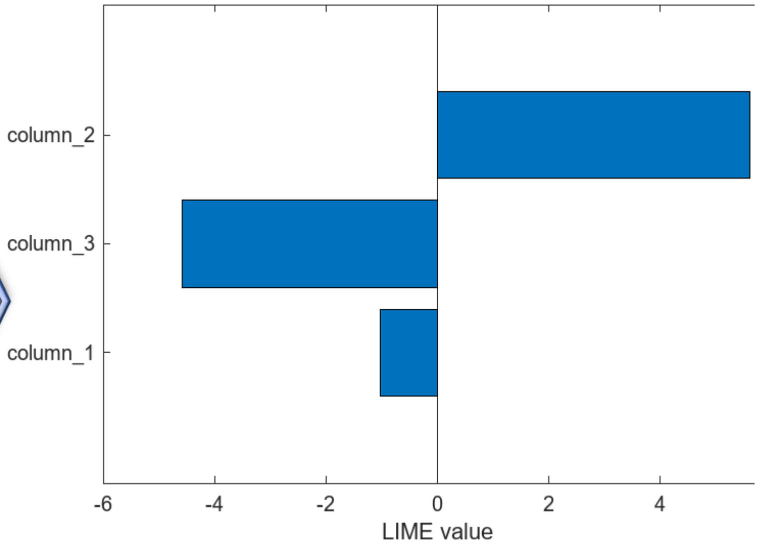


Figure II.25.Classification trilayered test lime

II.3.3.2 Discussion and Performance Analysis of Trilayered Neural Network:

a) Response plot:

Axis description:

X-axis: is the input which is 221 rows and 3 arrays of tension measures.

Y-axis: the response who ranges from 1 to 8, indicating the classified response into eight distinct classes

Data points:

the plot features are two types of data points. The first one is the orange who represent the predicted response and the second one is the blue data points who represent the actual data.

Error and class distribution: the errors are represented by orange vertical lines; the errors are not uniformly distributed. Errors appear to be present across all classes, but their magnitude varies.

Comparison:

The response plot for both training and test data reveals small differences, indicating high accuracy on the test set. This similarity occurs because the training data was also included in the test set, which contributed to the consistent performance seen.

b) Residual markers:

Description of the plot:

The x-axis indicates the genuine response range of 1 to 8. The y-axis represents the residuals, which are the differences between the actual and predicted values in the regression model.

Data points:

Each point on the plot shows a residual value for the accompanying true response. The distribution of these points provides insight into the model's performance across different true response values.The zero line helps to identify how the residuals are distributed, indicating whether the model underestimates or overestimates the responses. Analysis of the plot:

The residuals are scattered around the zero line, which is generally a good sign. Indicating that the model does not consistently overestimate or underestimate the true values. However, there is some noticeable spread in the residuals as the true response increases, suggesting potential heteroscedasticity which is data at widely varying distances from the line. Outliers:

A few points stand out from the general clustering; these are outliers, which occur when the model's predictions deviate significantly from the actual values. Similarities:

The distribution of residuals around the zero line is similar. indicating similar model behavior across different selections of data. Differences:

Tighter clustering of residuals around the zero line. This plot is significant because it plays an important role in the improvement of linear regression models by assisting in the assessment of the model's prediction accuracy.

c) validation predicted vs actual plot:

Linear Relationship:

The plot shows a strong linear relationship between the predicted and true response values, indicated by the proximity of data points to the diagonal line. This line represents perfect prediction accuracy, where predicted values exactly match the true values.

Mathematical representation:

The ideal relationship in this plot can be represented by the equation y = x and the deviation from this line represents prediction errors.

Data Distribution:

data points are spread across the entire range of true response values. Most of the data points are clustered around the line, with good model performance for the majority of predictions. There is some scatter away from the line, particularly in the mid-range of true values (around 4-6), indicating areas where the model's predictions are less accurate.

Model Performance:

The divergence from the mean, particularly in the intermediate range, suggests the presence of outliers in the data.

This plot is likely used to evaluate the performance of a regression model trained using a fine tree method, focusing on how well the model's predictions align with actual observed outcomes. The closer the points lie to the diagonal line, the more accurate the model is considered to be.

Comparison of training and test data:

The similarities are indicated by the consistent pattern of points around the line across different segments of the plot

d) confusion matrix:

There are two shapes of confusion matrix the first one is the confusion matrix number of observations and the second one is the validation confusion matrix TPR FNR. But the matrix being commented on is the second one. The concentration of points along the diagonal suggests that the model generally predicts values close to the actual values

This matrix is structured to show the model's performance across 8 different classes. Each cell in the matrix represents the percentage of predictions made by the model, where the rows represent the true classes and the columns represent the predicted classes.

The diagonal cells (from top left to bottom right) show the percentage of correct predictions for each class (True Positive Rate, TPR). These values are highlighted in blue, indicating higher accuracy.

The off-diagonal cells in each row show the percentage of instances that were misclassified, where the model predicted a different class than the true class.

The last two columns on the right side of the matrix provide the True Positive Rate (TPR) and False Negative Rate (FNR) for each class:

TPR is the percentage of actual positives that are correctly identified (also known as sensitivity). FNR is the percentage of actual positives that are not identified (also known as miss rate).

For example, in the test confusion matrix Class 1 has a TPR of 91.9% and an FNR of 8.1%. Class 2 has a TPR of 97.4% and an FNR of 2.6%. The highest TPR is for Class 8 at 100.0%, and its FNR is also 0%, indicating perfect prediction for this class.

This matrix is a useful tool for evaluating model's performance highlighting its strengths in predicting certain classes and its weaknesses in others.

e) Shapley:

The horizontal bar graph shows the Shapley values for all variables, sorted by their absolute values. Each Shapley value explains the deviation of the score for the query point from the average score of the predicted class, due to the corresponding variable. For regression models, predictions are response values. For a query point, the sum of the Shapley values for all predictors corresponds to the total deviation of the prediction from the average.[44]

These values are numerical representations used to quantify the contribution of each feature in a predictive model. The sum of Shapley values yields the difference between actual and average prediction.

The Shapley values in the regression range between -8 and 4. We have three bars; each bar represents a column in the input. The length and the direction of these bars indicate the magnitude and the direction of the contribution of each feature within this predictor to the model output. The first column had the largest bar, its direction was negative, this indicates that he is the most influential column on the negative side. Conversely, the second and third columns are less influential on the positive side. Therefore, it is recommended that the first line data be optimized in order to achieve a more favorable outcome.

The Shapley values in the classification range between -0.3 and 0.5. This plot is different from the first one. because the first one has only three bars, each bar defines a column. but this one has twenty-four bars because each bar has eight classes and here there is one Shapley for each class. the observed thing is that almost all the classes This has the potential to negatively impact the accuracy of the output. Still, some classes have a positive impact on the output of the classifier like the effect of the eight classes on the first column. This Shapley plot shows that the accuracy of the classification is low and we need to improve it.

f) ROC curve:

The ROC curve shows the true positive rate (TPR) versus the false positive rate (FPR) for different thresholds of classification scores, computed by the currently selected classifier. The Model Operating Point shows the false positive rate and true positive rate corresponding to the threshold value applied by the classifier to classify an observation.[44]

This graph plots the True Positive Rate (TPR) on the y-axis against the False Positive Rate (FPR) on the x-axis. Each curve represents a different class or operating point of the model, with a total of eight curves shown. Each curve is color-coded and has an associated Area Under Curve (AUC) value, which is a measure of the model's ability to distinguish between classes.

1. Blue Curve (AUC = 0.9481): This curve rises sharply towards the top-left corner, indicating a high true positive rate with a low false positive rate, suggesting excellent model performance for this class.
2. orange Curve (AUC = 0.9708): This curve is close to the top-left corner, showing that the model has a great performance with very high sensitivity and specificity.
3. yellow Curve (AUC = 0.9598): Also performs well, with a steep ascent and reaching close to the top-left, indicating high effectiveness in classification.
4. purple Curve (AUC = 0.956): This curve is performs well indicating a good ability to distinguish this class compared
5. green Curve (AUC = 0.9612): Similar to the purple curve but with a slightly higher AUC, indicating a better performance.
6. Light Blue Curve (AUC = 0.9527): Shows good performance, with a high true positive rate.
7. Dark red Curve (AUC = 0.9273): This curve is somewhat lower, indicating a slightly lesser ability to distinguish this class compared to others
8. Dark blue curve(AUC = 1): it’s an ideal top-left corner, indicating excellent model performance for this class.

The curves generally show high AUC values, suggesting that the model performs well across multiple classes. The orange and dark blue curves, with AUC values close to 1, are particularly indicative of excellent model performance. -Compared to typical ROC curves in similar classification tasks, these curves suggest that the model is highly effective, especially for the classes represented by orange and dark blue. In many classification tasks, achieving AUC values above 0.9 is considered excellent, and several curves here meet or exceed this threshold. - The presence of multiple curves with high AUC values also suggests that the model is robust across various classes, not just optimized for a single output.

Overall, this model \appears to be a highly effective model for classifying multiple classes, as indicated by the high AUC values across its ROC curves.

g) LIME:

This plot is useful for evaluating the performance and prediction spread of the model across different records, providing insights into its accuracy and the distribution of its predictions across the classified bins.[44]

Feature importance as assessed by LIME. A positive weight means the feature encourages the classifier to predict the instance as a positive and vice versa for the negative weight. Larger weights indicate greater feature importance.[45]

The chart displays three vertical bars each corresponding to a different predictor variable labeled as column-1, column-2, and column-3 some lime plots have two bars.

The y-axis is labeled as the predictor.

The x-axis represents the lime value.

In the regression column 1 is the most influential on the output positively and column 3 is less influential.

In the classification column 2 is the most crucial factor in the model's prediction, and column 3 has a negative lime value of -4.3 suggesting a strong negative influence but less than the first column and the first column had a less negative impact on the model's prediction.

Comparison between LIME plots:

Similarities: both types of plots use bar charts to represent the influence of each predictor, easy interpretation of the impact of each variable.

Differences: they show a different range of LIME values depending on the complexity and the interactions.

This plot is useful for identifying key drivers and detractors within the model.

h) Partial dependence :

The plot spans from -200 to 200 on the x-axis, representing the values of column 3.

Predicted Response:

The y-axis shows the predicted response, which starts at approximately 10 when column 3 is at -200. As the value of column 3 increases, the predicted response gradually decreases.

behavior of the Model: Initially, the response decreases slowly and becomes more stable around a value of 8. As column 3 continues to increase beyond 0, the response starts to decrease more sharply, reaching below 5 when column 3 is at 200.

Stability: The model's response is relatively stable in the middle range of column_3 but shows increased variability at the extremes. This could indicate that the model is less certain or potentially overfitting or underfitting at these extreme values.

Model Application: Understanding how changes in column_3 affect the output can help in fine-tuning the model and in feature engineering.

II.4 Decision tree:

II.4.1 Algorithm Description:

the algorithm helps make a decision, and its structure is similar to that of a tree. The decision tree contains hubs that create a known tree, which implies that it is a coordinated tree through a node called "root" with no imminent edges, while very varied hubs have just one imminent edge.[46]

II.4.2 Decision tree algorithm:[46]

In general, for simple classification the most usual algorithms are ID 3 and C 4.5. In this case, the presented algorithm is ID 3:

Inputs: R: a collection of non-target attributes, C: target attribute, S: training data.
Output: returns a decision tree

Step 1: Initialization of an empty tree;
Step 2: If S is empty, then
Step 3: Returns the failure value of a single node
Step 4: End If
Step 5: If S is rendered for values of the same target only, then
Step 6: Returns a single node with this value
Step 7: End if
Step 8: If R is empty, then
Step 9: Return the commonest value in S for the target attribute values.
Step 10: End if
Step 11: D ← attribute with the largest Gain (D, S) among all the attributes of R
Step 12: {dj j = 1, 2,..., m} ← Attribute values of D
Step 13: {Sj with j = 1, 2, ..., m} ← The subsets of S composed of the dj records attribute value D, correspondingly
Step 14: Return a tree whose root is D, and the arcs are labeled by d1, d2, ..., dm and going to sub-trees ID3 (R-{D}, C, S1), ID3 (R-{D} C, S2), .., ID3 (R-{D}, C, Sm)

II.4.3 Simulink Implementation:

II.4.3.1 Diagram:

a) Classification fine tree :

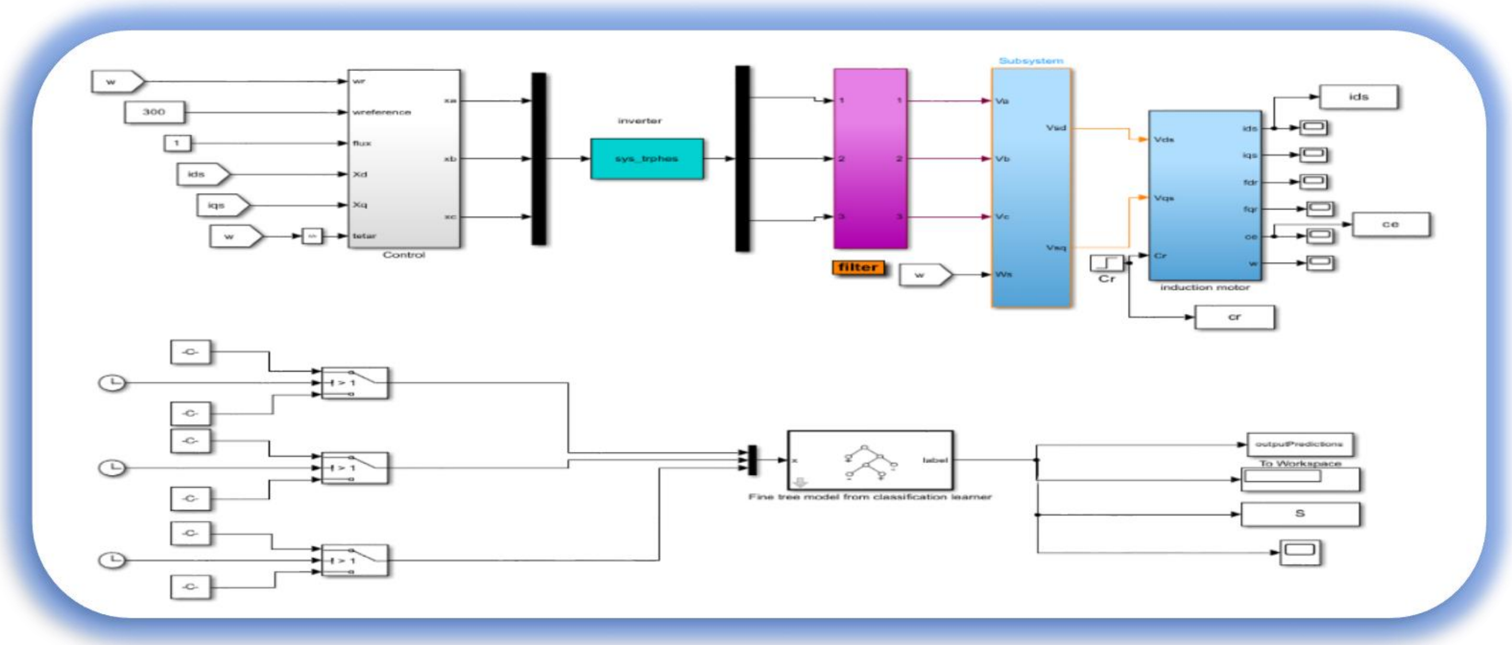


Figure II.26. Classification fine tree model integrated with induction motor

b) Regression learner :

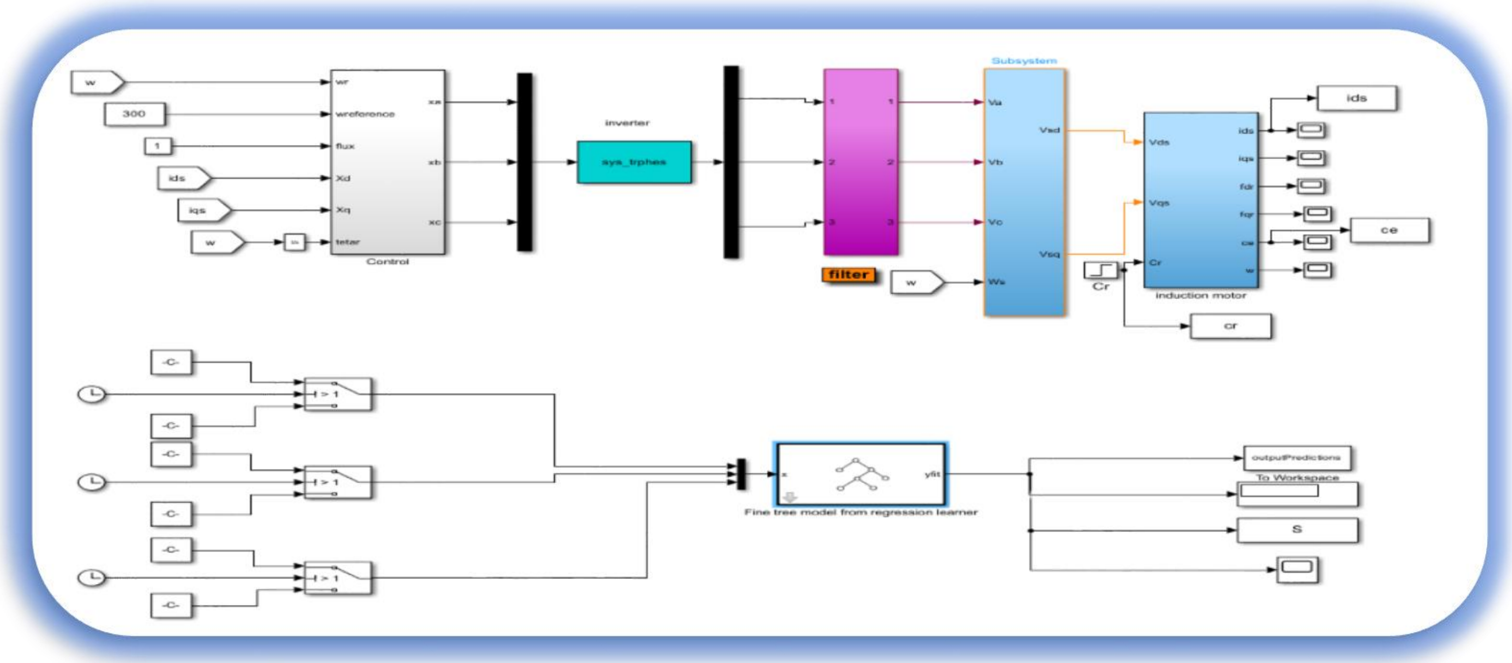


Figure II.27. Regression fine tree model integrated with induction motor

II.4.4 Results:

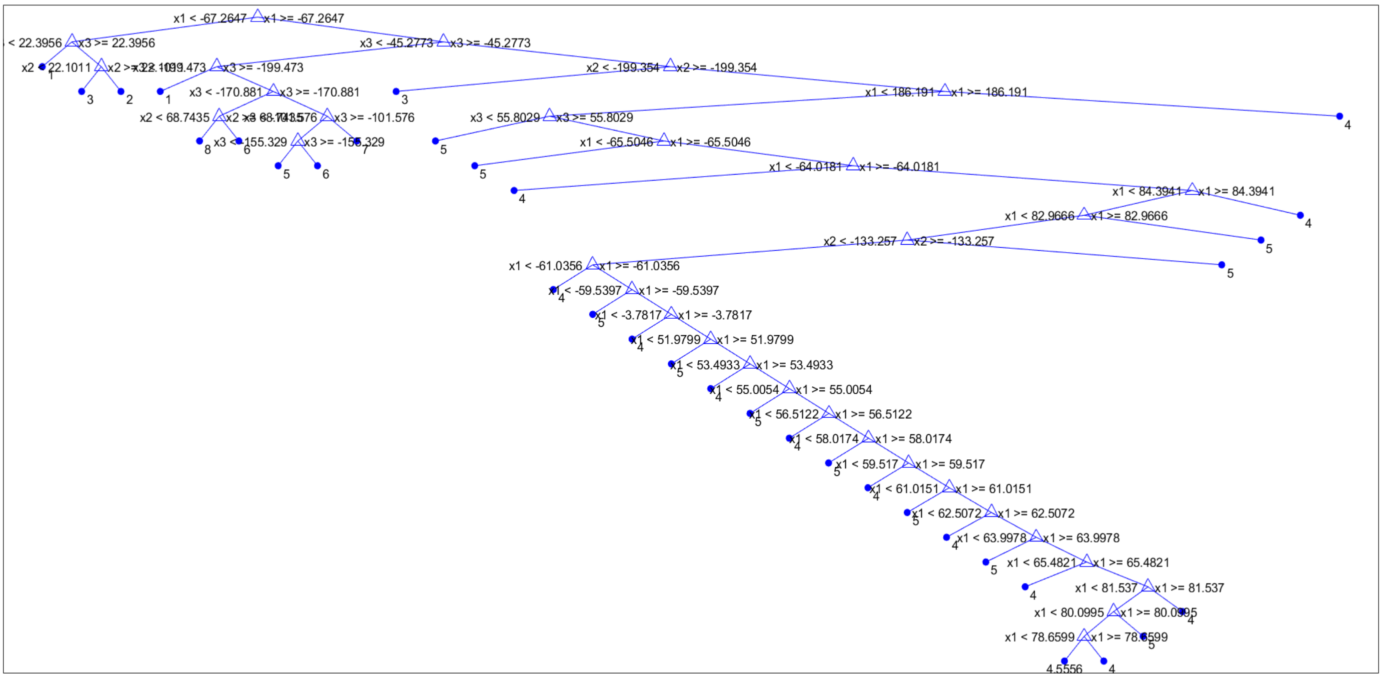


Figure II.28. regression tree for fine tree training

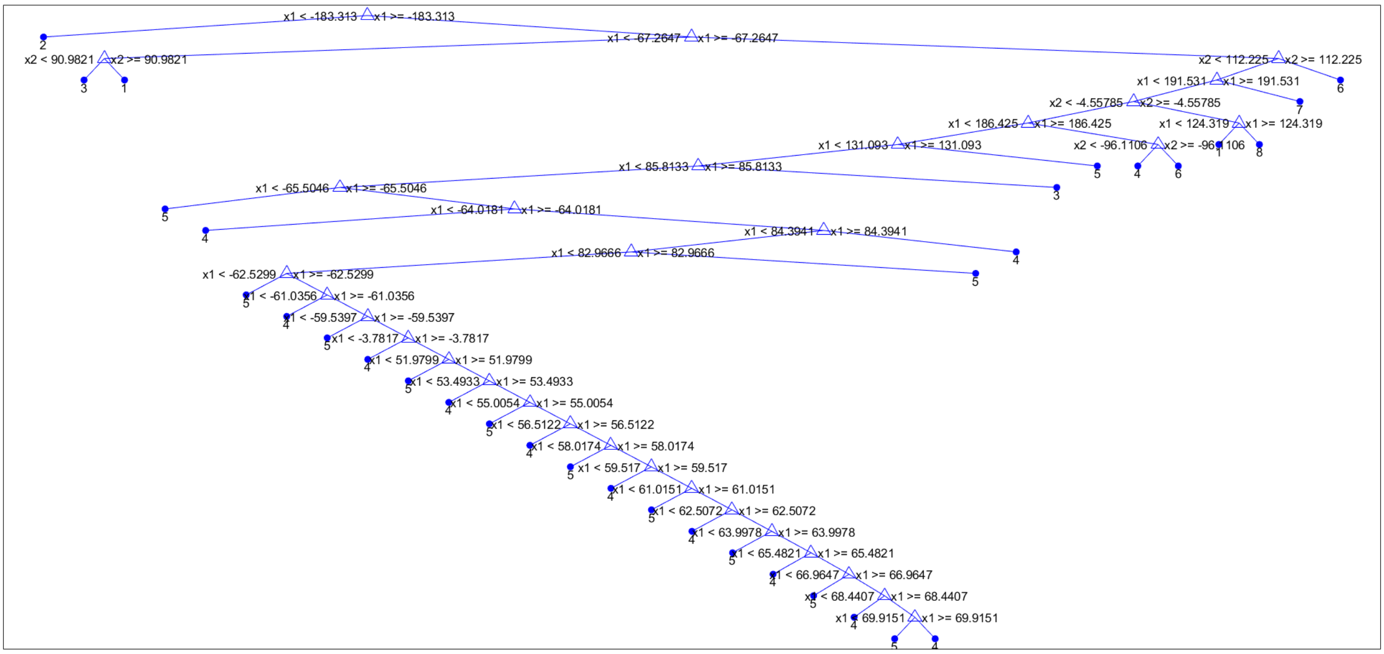


Figure II.29. classification tree for fine tree training

II.4.4.1 Training and Testing Results:

a) Regression :

Training: RMSE (validation) 0.53300

Test: RMSE (validation): 0.35635

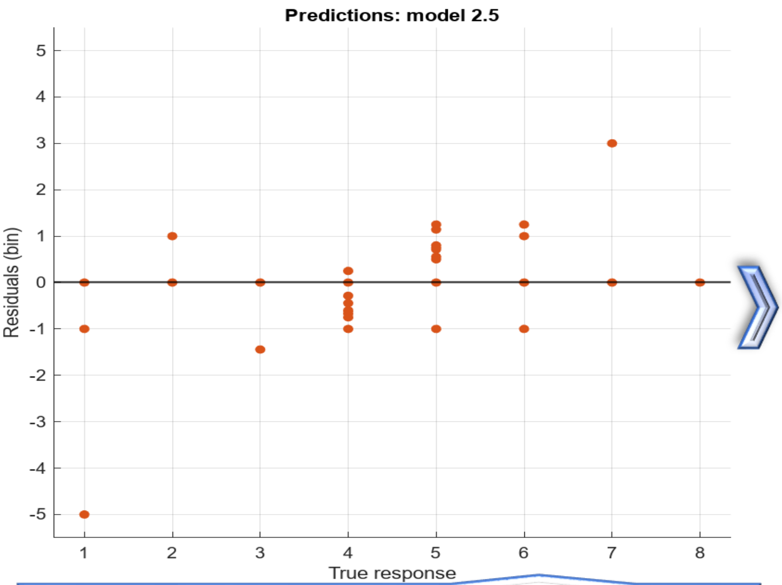


Figure II.30.Regression fine tree training validation residuals plot markers true response

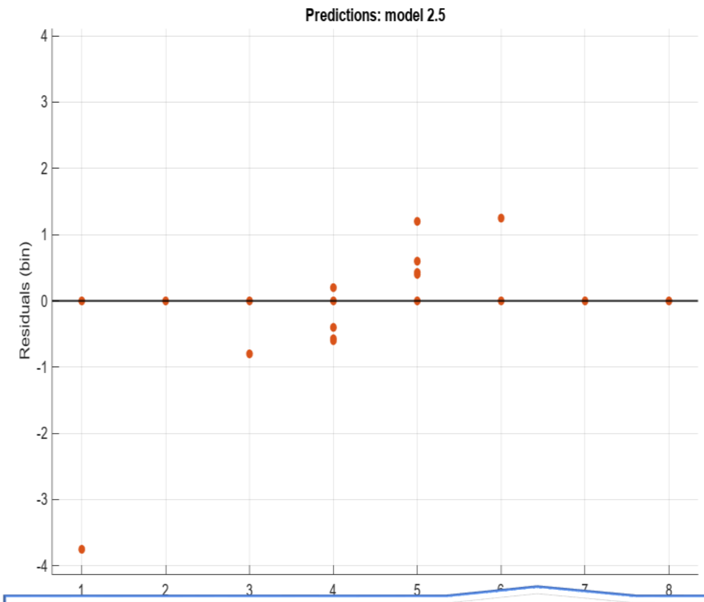


Figure II.31.Regression fine tree test validation residuals plot markers true response

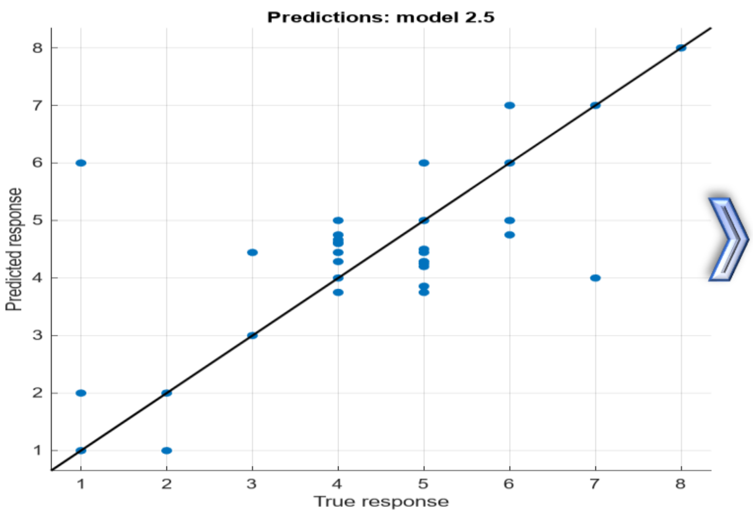


Figure II.32.Regression fine tree training validation predicted vs actual plot

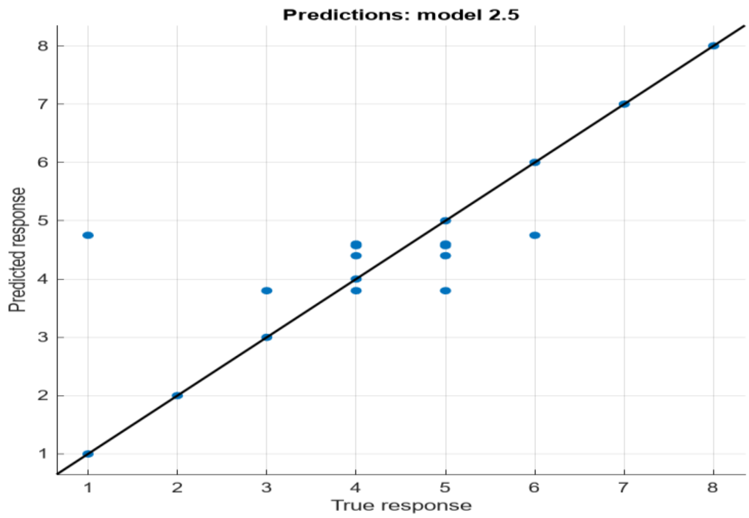


Figure II.33.Regression fine tree testvalidation predicted vs actual plot

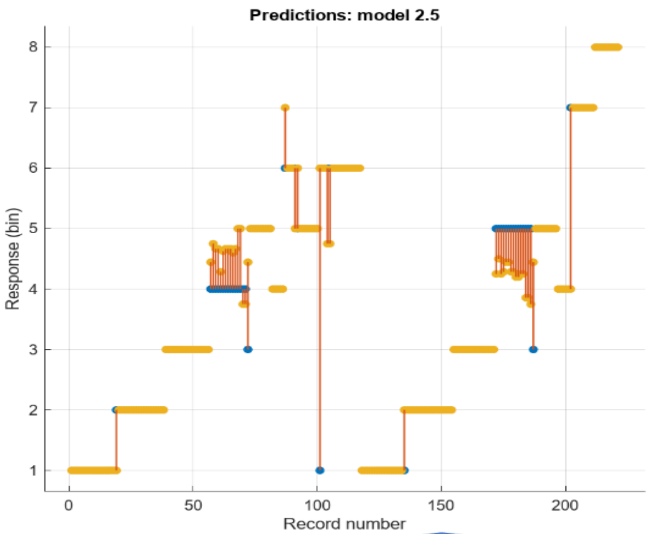


Figure II.34.Regression fine tree training response plot

b) Explain:

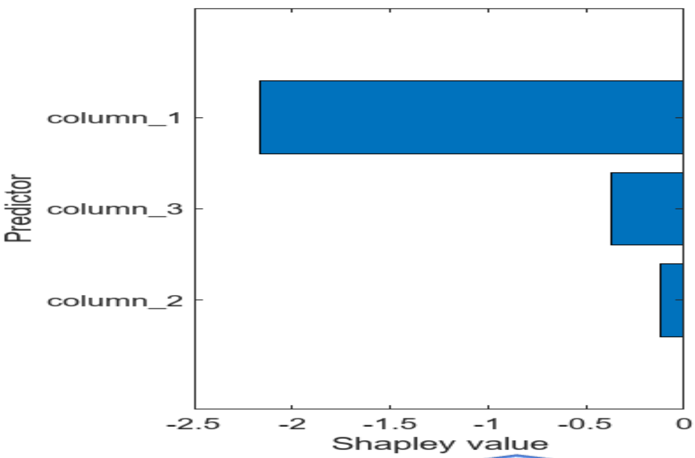


Figure II.35.Regression fine tree training local shapely explanation

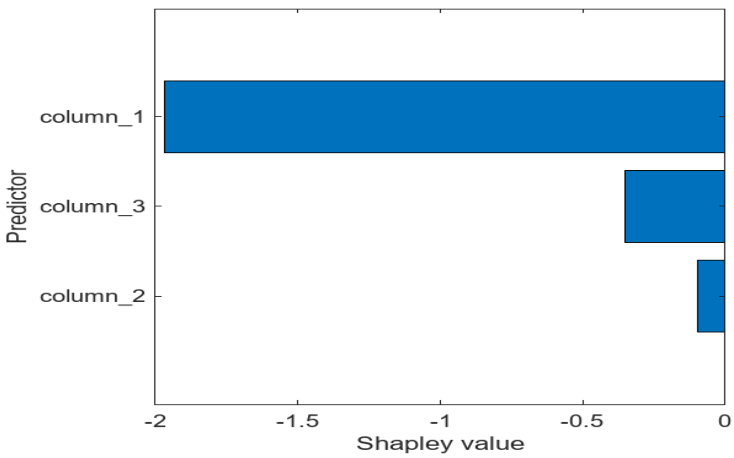


Figure II.36.Regression fine tree test local shapely explanation

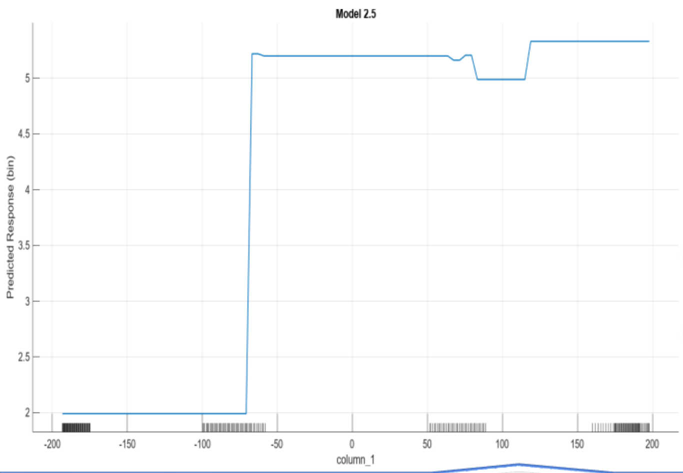


Figure II.37.Regression fine tree training partial dependence for column 1

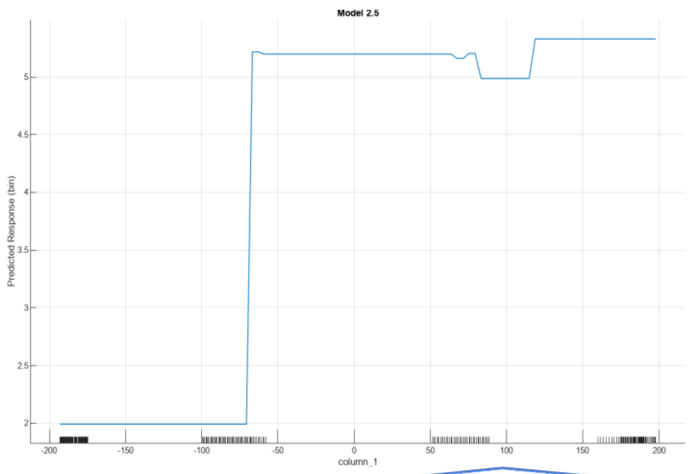


Figure II.38.Regression fine tree test partial dependence for column 1

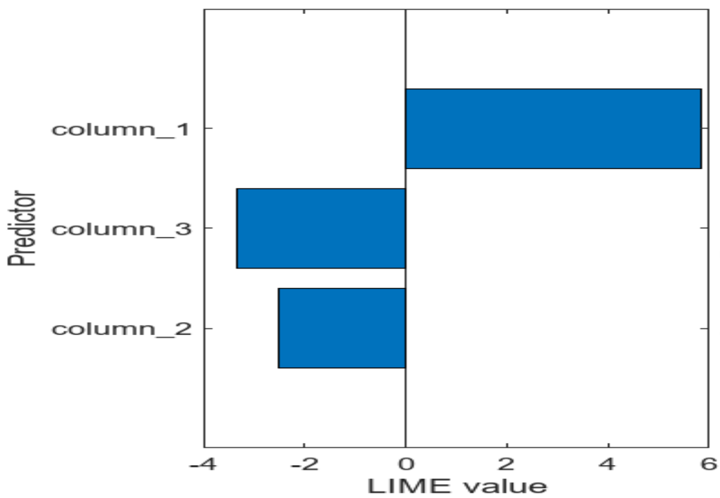


Figure II.39.Regression fine tree training lime

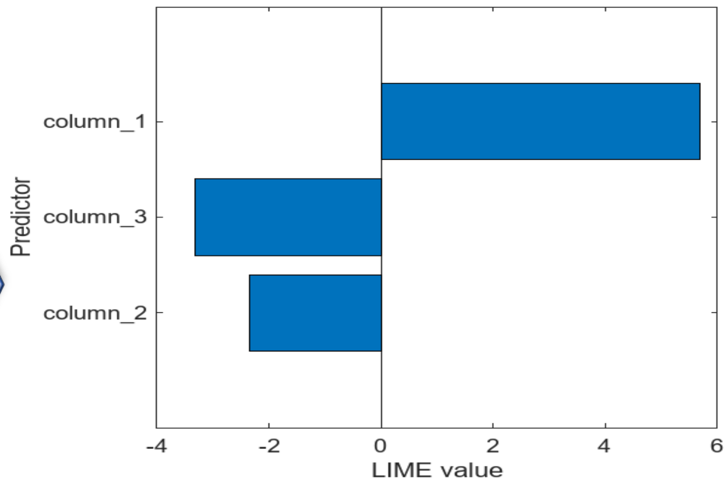


Figure II.40.Regression fine tree test lime

c) Classification

Training: accuracy validation = 83.7

Test: accuracy validation98.2

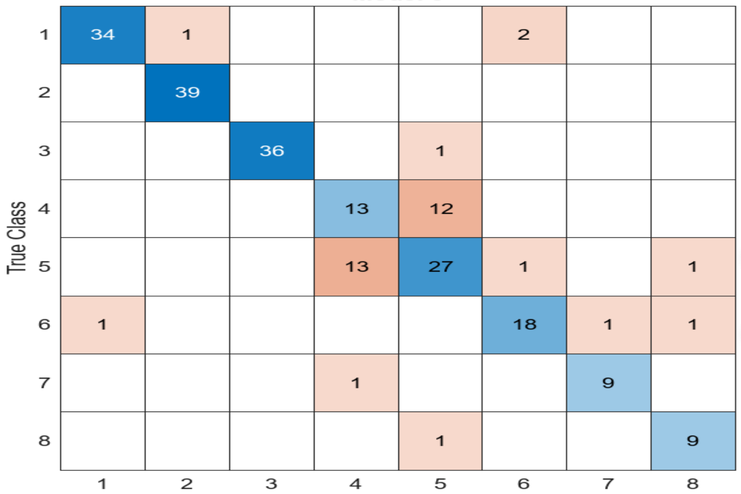


Figure II.41.Classification fine tree training confusion matrix

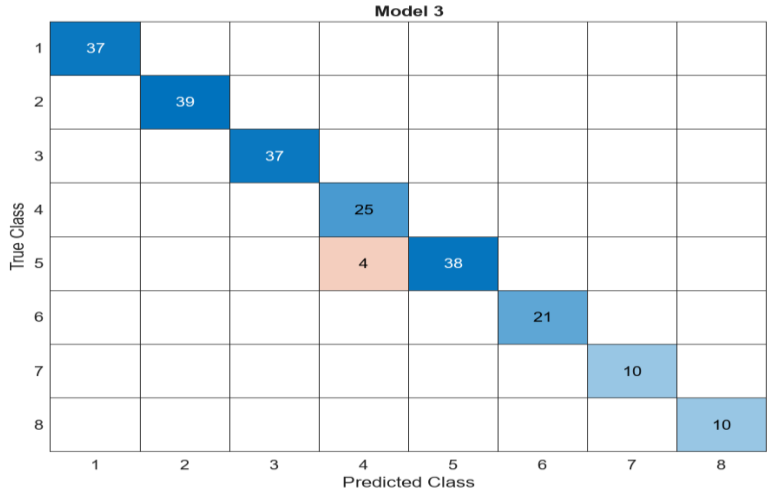


Figure II.42.Classification fine tree test confusion matrix

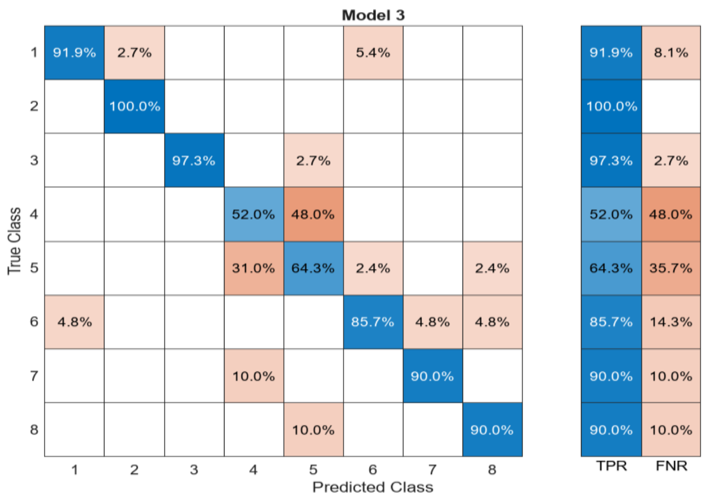


Figure II.43.Classification fine tree training validation confusion matrix TPR FNR

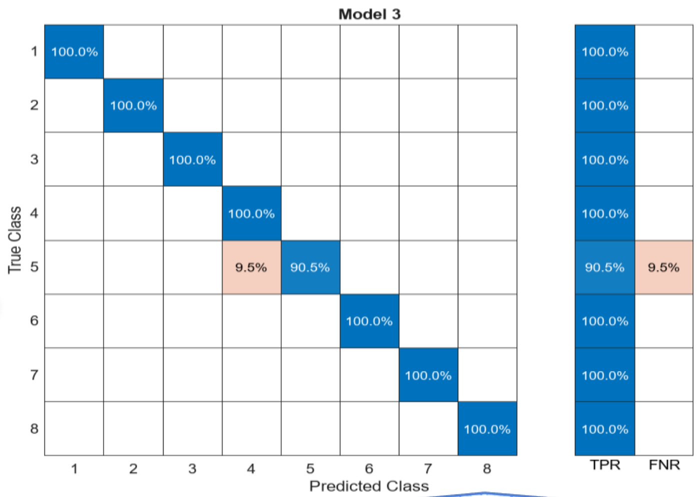
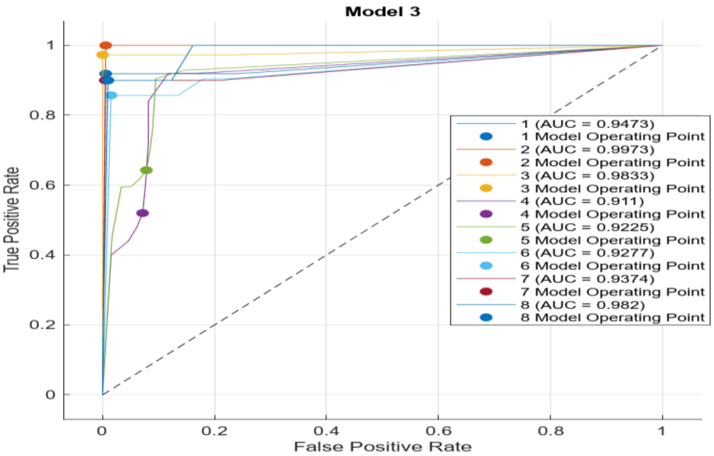
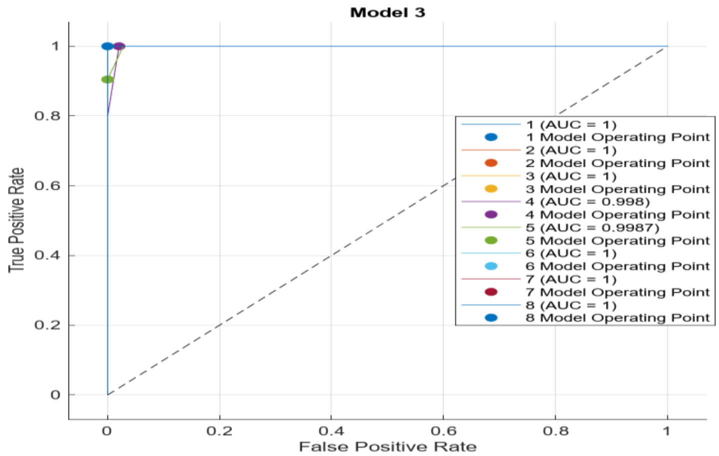


Figure II.44.Classification fine tree training validation confusion matrix TPR FNR



II.45.Classification fine tree training ROC curve



II.46.Classification fine tree test ROC curve

d) Explain :

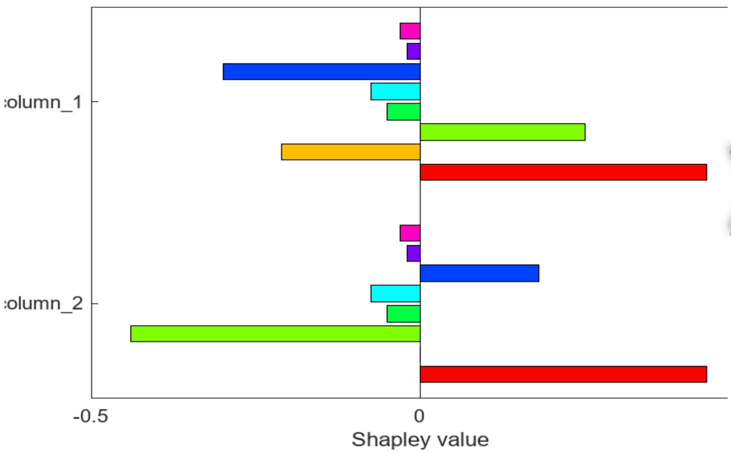


Figure .47.Classification fine tree training local shapely

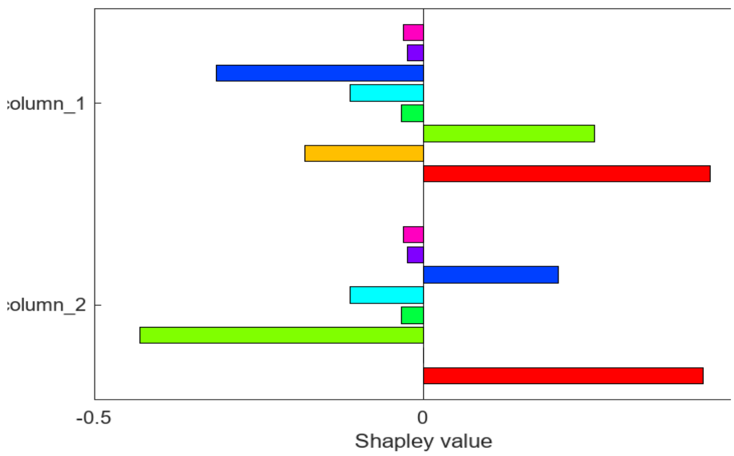


Figure .48.Classification fine tree test local shapely

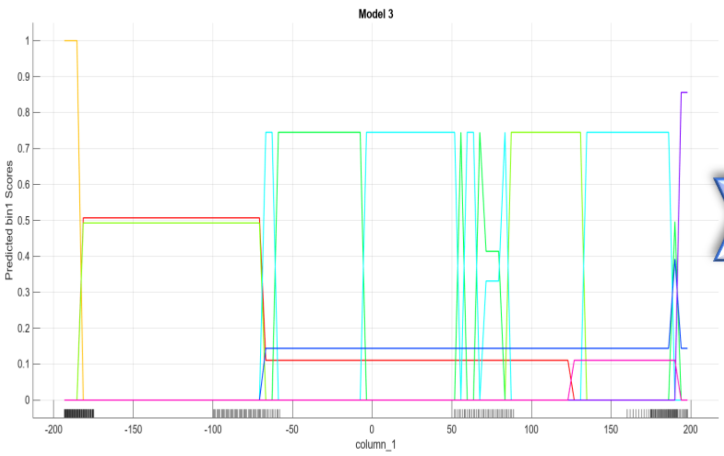


Figure II.49.Classification fine tree training partial dependence plot

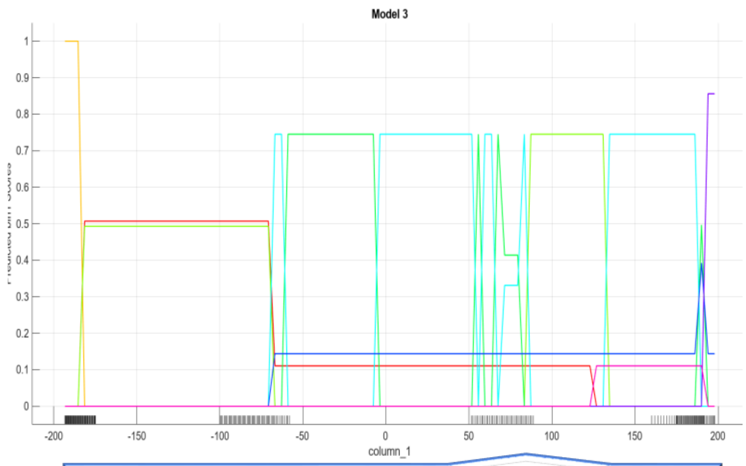


Figure II.50.Classification fine tree test partial dependence plot

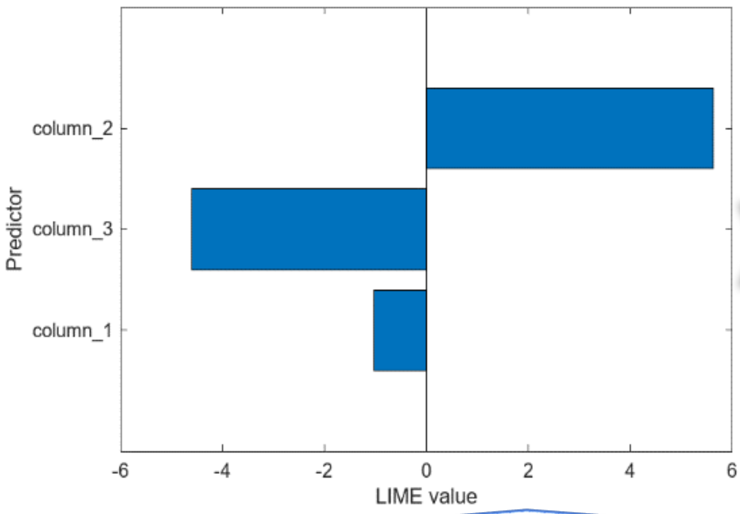


Figure II.51.Classification fine tree training lime

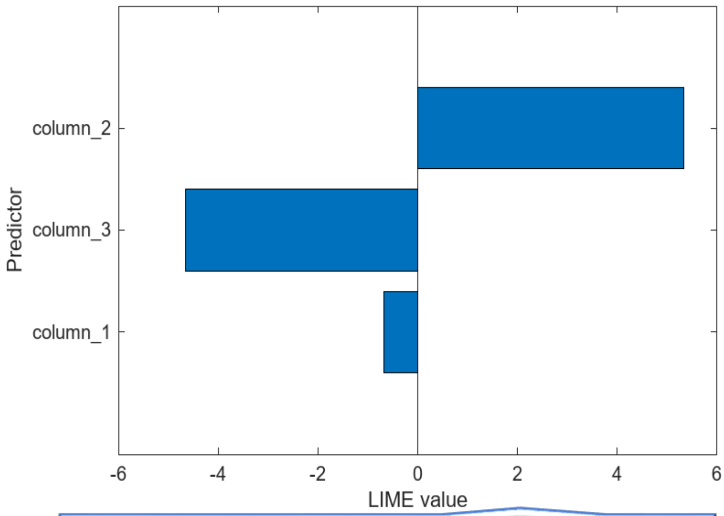


Figure II.52.Classification fine tree test lime

II.4.4.2 Discussion:

a) Response plot:

Axis description:

X-axis: is the input which is 221 rows and 3 arrays of tension measures.

Y-axis: the response who ranges from 1 to 8, indicating the classified response into eight distinct classes

Data points:

The plot displays two categories of data points. The orange points signify the predicted responses, while the blue points denote the actual observed data.

Error and Class Distribution: Errors are illustrated by orange vertical lines, indicating that they are not evenly distributed. These errors are evident across all classes, with varying magnitudes across different points.

Comparison:

The response plot for both the training and test data shows very small differences, indicating high accuracy on the test set. The consistent performance across both datasets highlights the model's effectiveness in making accurate predictions.

b) Residual markers:

Description of the plot:

The x-axis indicates the genuine response range of 1 to 8.

The y-axis represents the residuals, which are the differences between the actual and predicted values in the regression model.

Data points:

Each point on the plot shows how much the predicted values differ from the actual values. This distribution gives us an idea of how well the model performs across different true response values.The zero line helps us see whether the model tends to underestimate or overestimate the responses by showing where the residuals cluster around

Analysis of the plot:

The residuals are distributed around the zero line, indicating a balanced performance where the model neither consistently overestimates nor underestimates the true values. However, there is a noticeable variability in the residuals as the true responses increase, suggesting potential heteroscedasticity – uneven variance in the data spread around the line.

Outliers: Some points notably diverge from the main cluster, indicating outliers where the model predictions significantly differ from the actual values.

The residuals exhibit a similar distribution around the zero line, suggesting consistent model behaviour across different data selections.

Heteroscedasticity:

The spread of residuals is not consistent across all predicted values. There's more variability in residuals for middle-range predictions (around 3-6) compared to the extremes.

Differences:

Residuals show tighter clustering around the zero line compared to previous observations, indicating potentially improved model precision.

This plot is crucial for refining linear regression models as it helps gauge prediction accuracy and identifies areas for model enhancement.

c) validation predicted vs actual plot:

Linear Relationship:

The plot clearly demonstrates a strong linear relationship between the predicted and true response values. The data points closely align with the diagonal line, indicating that the model's predictions are highly accurate and closely match the actual values.

Mathematical representation:

The ideal relationship in this plot can be represented by the equation $y = x$, where the predicted values perfectly match the true values. When data points fall directly on this line, it signifies perfect prediction accuracy. Any deviation from this diagonal line represents prediction errors, highlighting the discrepancies between the model's predictions and the actual observed values.

Data Distribution:

The data points are predominantly clustered around the line, especially for lower values of true response (from 1 to 4). However, as the true response values increase (from 5 to 8), the data points show more variance from the line, indicating less accuracy in the predictions for higher values. - There are noticeable deviations where multiple predicted values for a single true response value are either underestimating or overestimating the true response, particularly noticeable in the middle range of the true response values (around 4 to 6).

Model Performance:

The concentration of points along the diagonal suggests that the model generally predicts values close to the actual values. The deviation from the line, especially in the middle range, indicates potential outliers in the data.

Comparison of training and test data:

The similarities are indicated by the consistent pattern of points around the line across different segments of the plot

d) confusion matrix:

There are two shapes of confusion matrix the first one is the confusion matrix number of observations and the second one is the validation confusion matrix TPR FNR. But the matrix being commented on is the second one.

This matrix is organized to showcase the model's performance across 8 different classes. Each cell contains the percentage of predictions made by the model, with the rows corresponding to the actual classes and the columns corresponding to the predicted classes. This format allows for a detailed analysis of the model's classification accuracy, highlighting its strengths and areas where it may need improvement.

The diagonal cells (from top left to bottom right) show the percentage of correct predictions for each class (True Positive Rate, TPR). These values are highlighted in blue, indicating higher accuracy.

The off-diagonal cells in each row show the percentage of instances that were misclassified, where the model predicted a different class than the true class.

The last two columns on the right side of the matrix provide the True Positive Rate (TPR) and False Negative Rate (FNR) for each class:

TPR is the percentage of actual positives that are correctly identified (also known as sensitivity). FNR is the percentage of actual positives that are not identified (also known as miss rate).

This suggests the classification model has very poor performance, with a high rate of misclassifications across all 8 classes. Further investigation into the model architecture, training data, and hyperparameters would be needed to improve the classification accuracy.

e) Shapley:

The horizontal bar graph shows the Shapley values for all variables, sorted by their absolute values. Each Shapley value explains the deviation of the score for the query point from the average score of the predicted class, due to the corresponding variable. For regression models, predictions are response values. For a query point, the sum of the Shapley values for all predictors corresponds to the total deviation of the prediction from the average.[44]

These values are numerical representations used to quantify the contribution of each feature in a predictive model. The sum of Shapley values yields the difference between actual and average prediction.

The Shapley values in the regression range between -2.5 and 0. We have three bars; each bar represents a column in the input. The length and the direction of these bars indicate the magnitude and the direction of the contribution of each feature within this predictor to the model output. The first column had the largest bar, its direction was negative, this indicates that he is the most influential column on the negative side. the second and third columns are less influential on the negative side. Therefore, it is recommended that all of the data lines be optimized in order to achieve a more favorable outcome.

The Shapley values in the classification range between -0.5 and 0.5 each bar defines a column. but this one has sixteen bars because each bar has eight classes and here there is one Shapley for each class. the observed thing is that almost all the classes This has the potential to negatively impact the accuracy of the output. Still, some classes have a positive impact on the output of the classifier like the effect of the eight classes on the first column. This Shapley plot shows that the accuracy of the classification is low and we need to improve it.

f) ROC curve:

Blue Curve (AUC = 0.9473): This curve rises sharply towards the top-left corner, indicating a high true positive rate with a low false positive rate, suggesting excellent model performance for this class.

Red Curve (AUC = 0.9973): This curve is very close to the top-left corner, almost ideal, showing that the model has an excellent performance with very high sensitivity and specificity.

Orange Curve (AUC = 0.9833): Also performs well, with a steep ascent and reaching close to the top-left, indicating high effectiveness in classification.

Green Curve (AUC = 0.911): This curve is somewhat lower, indicating a slightly lesser ability to distinguish this class compared to others.

Purple Curve (AUC = 0.9225): Similar to the green curve but with a slightly higher AUC, indicating a better performance.

Light Blue Curve (AUC = 0.9277): Shows good performance, with a high true positive rate.

Light Green Curve (AUC = 0.9374): This curve is higher than the light blue and purple curves, indicating better performance.

Dark Red Curve (AUC = 0.982): Very close to the ideal top-left corner, indicating excellent model performance for this class.

Analysis and Comparison:

The curves generally show high AUC values, suggesting that the model performs well across multiple classes. The red and dark red curves, with AUC values close to 1, are particularly indicative of excellent model performance.

Compared to typical ROC curves in similar classification tasks, these curves suggest that this is highly effective, especially for the classes represented by the red and dark red curves. In many classification tasks, achieving AUC values above 0.9 is considered excellent, and several curves here meet or exceed this threshold.

The presence of multiple curves with high AUC values also suggests that the model is robust across various classes, not just optimized for a single output.

Overall, this appears to be a highly effective model for classifying multiple classes, as indicated by the high AUC values across its ROC curves.

Overall, this model \appears to be a highly effective model for classifying multiple classes, as indicated by the high AUC values across its ROC curves.

g) LIME:

The chart displays three vertical bars each corresponding to a different predictor variable labeled as column-1, column-2, and column-3 some lime plots have two bars.

The y-axis is labeled as the predictor.

The x-axis represents the lime value.

Analyze the performance of regression LIME:

Column 1 has a positive LIME value of approximately 5, suggesting a strong positive influence on the model's prediction. Column 3 has a LIME value of around -1, indicating a slight negative impact. Column 2 has a LIME value of about -3, showing a more significant negative influence compared to column 3.

column1 appears to be the most influential predictor with its high positive value, making it a critical factor in the model. In contrast, column 2 and column 3 negatively impact the model's output, with column2 being more detrimental.

h) Partial dependence :

The plot shows that as the feature value increases, the target variable also increases in a roughly linear fashion. This implies that the feature in column 3 has a positive relationship with the target variable.

Overall, the similar training and test curves suggest that the model is performing well and not overfitting to the training data.

We can see that the training and test curves are quite similar, indicating that the model is not overfitting to the training data. The curves have a similar shape and magnitude, suggesting that the model is generalizing well to the test set.

The plot shows that as the feature value increases, the target variable also increases in a roughly linear fashion. This implies that the feature in column 3 has a positive relationship with the target variable.

Overall, the similar training and test curves suggest that the model is performing well and not overfitting to the training data.

II.5 Comparison between neural network and decision tree:

II.5.1 Methodology:

Both can be used for classification and regression tasks. Appropriate algorithms should be used for each task, such as classification trees for categorizing faults and regression trees for predicting continuous outcomes.

II.5.2 Performance:

Neural Networks are known for their ability to model complex relationships in data. In regression mode, they can capture non-linear patterns effectively. The response plot for the neural network in regression mode shows how well the model fits the data. Additionally, LIME and SHAP values help in understanding the contribution of each feature to the prediction.

Decision Trees, on the other hand, are simpler models that partition the data into subsets based on feature values. In regression mode, they create piecewise constant approximations. The response plot for the decision tree in regression mode highlights the step-like nature of the model's predictions. LIME and SHAP values for decision trees provide insights into the importance of each feature in the decision-making process.

Local Interpretability (LIME): Both models can be interpreted locally using LIME. The LIME plots for both the Neural Network and Decision Tree classifiers provide insights into the contribution of each feature for specific instances. This is particularly useful for understanding individual predictions.

Global Interpretability (SHAP): SHAP values offer a global perspective on feature importance. The SHAP summary plots for both classifiers show the overall impact of each feature on the model's predictions. This helps in identifying the most influential features across the entire dataset.

Model Complexity: Neural Networks are generally more complex and can capture intricate patterns in the data, while Decision Trees are simpler and easier to interpret. This is reflected in the

SHAP values, where the Neural Network may show more nuanced feature contributions compared to the Decision Tree.

Overall, both LIME and SHAP provide valuable insights into the behavior of the classifiers, helping us understand and compare the models' decision-making processes.

II.5.3 Interpretation:

II.5.3.1 Neural Networks:

Generally perform better with complex patterns and large datasets.

II.5.3.2 Decision Trees:

Simpler, more interpretable, and faster to train but may suffer from overfitting.

II.5.4 Advantages and disadvantages:

II.5.4.1 Advantages of Decision Trees:

a) Interpretability:

Decision trees are highly interpretable and easy to understand. Each decision node represents a feature, and the branches represent the decision rules, making it straightforward to follow the logic of the model.

b) Simplicity:

They are simple to implement and require less computational power compared to neural networks. This makes them suitable for real-time applications where quick decisions are needed.

c) Handling Non-linear Relationships:

Decision trees can handle non-linear relationships between features without requiring any transformation of the data.

d) Feature Importance:

They provide a clear indication of which features are most important for prediction, which can be useful for understanding the factors affecting motor performance.

II.5.4.2 Disadvantages of Decision Trees:

a) Overfitting:

Decision trees are prone to overfitting, especially when they are deep. This can lead to poor generalization to new data.

b) Instability:

Small changes in the data can result in a completely different tree being generated, making them less stable compared to neural networks.

c) Bias:

They can be biased if some classes dominate. Techniques like balancing the dataset or using ensemble methods (e.g., Random Forests) can mitigate this issue.

II.5.4.3 Advantages of Neural Networks:

a) High Accuracy:

Neural networks, especially deep learning models, can achieve high accuracy by learning complex patterns in the data.

b) Scalability:

They can handle large datasets and a high number of features, making them suitable for complex motor performance analysis.

c) Flexibility:

Neural networks can be adapted to various types of data and tasks, including regression, classification, and time-series forecasting.

II.5.4.4 Disadvantages of Neural Networks:

a) Complexity:

Neural networks are complex and require significant computational resources for training and inference. This can be a limitation for real-time applications.

b) Interpretability:

They are often considered "black boxes" because it is difficult to interpret how they make decisions. This can be a drawback when understanding the factors affecting motor performance is crucial.

c) Training Time:

Training neural networks can be time-consuming, especially for deep learning models with many layers and parameters.

II.5.5 Suggestions for improvement:

Ensemble methods like Random Forests and Gradient Boosting Trees can improve performance and robustness.

II.6 Conclusion:

In this chapter, we conducted a mathematical modelling and comparative analysis of two AI-based diagnostic methods, namely neural networks and decision trees, applied to induction motors. A robust mathematical foundation was established by detailing the critical equations governing motor dynamics and implementing these models in Simulink to facilitate dynamic simulation.

The comparative analysis demonstrated that both neural networks and decision trees offer significant advantages in diagnosing motor faults and predicting performance. Each approach has unique strengths. Neural networks are particularly adept at handling complex, non-linear relationships, whereas decision trees offer interpretability and efficiency in decision-making.

In conclusion, this chapter demonstrates the viability and benefits of integrating AI techniques into electromechanical diagnostics, highlighting their potential to enhance system reliability and performance. These findings provide a foundation for further research and development in intelligent diagnostic systems for electromechanical applications.

Chapter III:

Implementation of a fault diagnosis system based on a neural network on an FPGA circuit

III.1 Introduction:

Monitoring asynchronous motors is critical for guaranteeing continuous manufacturing operations and avoiding production chain interruptions. Given that asynchronous machines are critical components in a variety of sectors, they must operate continuously. Early diagnosis of defects reduces not just their severity but also their influence on the system. Rapid fault localization accelerates maintenance operations, allowing for timely remedial measures after diagnosis. As a result, constant monitoring and real-time diagnosis emerge as essential activities.

This chapter focuses on the field of autonomous defect diagnosis using artificial intelligence, specifically neural networks. Our primary goal is to use neural networks to monitor asynchronous motors and detect and fix any defects as soon as possible. These include phase failure, phase imbalance, and short-circuits in the motor system. Our goal is to create a neural network on an FPGA circuit by using the stator phase current's RMS signal as a defect indication.

The major goal of this implementation is to investigate the effectiveness of hardware integration solutions, namely FPGA circuits, in diagnosing asynchronous motor problems, with a special emphasis on phase faults. Our strategy begins with adapting the neural network to guarantee optimal implementation, with a focus on efficiency, execution speed, and low space use on the FPGA circuit. The neural system is then programmed using the System Generator, a tool used to generate VHDL code. The produced code is verified and implemented on a VIERTEX FPGA circuit via Xilinx's ISE Foundation software.

Through this comprehensive study, we aim to ascertain the effectiveness of FPGA-based neural network implementations in diagnosing asynchronous motor faults, thereby advancing the field of fault diagnosis in electromechanical systems.

III.2 FPGA Hardware in the Loop:

FPGAs are integrated circuits made up of an array of identical logic cells interconnected by programming (linked by configurable communication buses). The principle basic is simple: in order to implement a logic function in an FPGA, it suffices to configure the logic cells and to link them correctly using the internal buses. Indeed, FPGAs are the descendants of CPLD, they are simply much more complex and powerful. There are all of them sizes, from a few thousand logic gates to a few million, which makes it possible to implement complex circuits without having to design an ASIC. In addition to configurable cells, many FPGAs have more complex memories and modules like multipliers. Some manufacturers even go so far as to implement microprocessor cores [47]. In addition, they are fully tested after production, so the designs do not require the generation of component test programs, no the automatic generation of test vectors (or design for testability). They only require functional type testing. As far as speed is concerned, FPGAs offer units that operate at speeds exceeding 400MHz in many applications. The speed of FPGAs is adequate for most applications. In critical cases, applications can be speeded up simply by using faster units, often without changing the circuit design. With dedicated circuits, the situation is completely different. Indeed, new manufacturing processes require the production of masks and increase the overall cost and production time. The development of FPGAs is accompanied by a constant evolution of design tools. These high-level tools remain affordable even by small design companies. The development time is made up solely of the time for simulation and production of the prototype, while the time for the other phases necessary for the dedicated circuits: generation of test vectors / production of masks, manufacture of waffles, packaging and testing final in manufacture, are avoided. This leads to a development time for FPGAs measured in days or weeks. While for dedicated circuits the durations are calculated in months FPGAs have several advantages compared to other technologies, mainly [48]:

• Given their structure, all functions can be implemented, even the most complex, including including microprocessors or DSPs

• Can be reconfigured indefinitely.

• Parallel operation.

The legs of an FPGA are generally configurable and can adapt to several protocols communication (TTL, PC, etc.).

• Their programming is done by advanced languages, such as VHDL, Verilog, System generator. The FPGA has some disadvantages (which are not very important) such as:

• Programming these circuits requires learning the programming language, although that several studies have contributed to generate the VHDL language by more advanced languages, tonamely: the C or MATLAB language;

• Since FPGAs are not dedicated circuits, the functions implemented are more slow and take up more space compared to a dedicated circuit, such as the DSP;

• Floating point representation is nearly impossible.

III.3 Application of neural network and decision tree:

To effectively address the diagnosis of failures in electromechanical systems using both neural networks and decision trees, a structured approach must be adopted [42,49,50,51,52,53]. This approach entails two primary steps:

- **Problem Examination and Validation:**
  The first step involves thoroughly analyzing the problem to be addressed, assessing its compatibility with neural network and decision tree methodologies, and defining clear objectives for the solution. This phase is critical for ensuring the quality and effectiveness of the chosen approach.

- **Neural Network and Decision Tree Technology Implementation:**
  The second step focuses on the technical aspects of implementing neural networks and decision trees. This includes selecting appropriate network types, configuring parameters such as the number of hidden layers and learning algorithms, and tailoring the implementation to suit the specific characteristics of the problem at hand and the desired objectives.

For the proposed fault diagnosis system targeting induction motor faults, the neural network architecture comprises three layers: an input layer with three neurons transmitting values corresponding to input variables (Veff1, Veff2, and Veff3), a hidden layer, and an output layer with three neurons producing binary outputs (0 or 1). The RMS voltages Veff1, Veff2, and Veff3 are computed using the RMS block available in the Simulink library.

The second stage of designing the artificial neural network (ANN) involves the learning process, which necessitates a comprehensive database defining the input-output mapping. This database typically adopts a matrix format, clarifying inputs and desired outputs. In this application, the matrix consists of three inputs corresponding to RMS voltages and three digits representing fault codes. Optimal learning requires a rich database encompassing various fault scenarios. To achieve this, the following tasks are undertaken:

a. Simulation of the machine in a normal (healthy) state.
b. Simulation of the machine under abnormal conditions (e.g., single-phase, two-phase faults).
c. Acquisition of RMS values in each scenario, including the healthy state.

This process results in a comprehensive classification of different machine states, facilitating effective fault diagnosis as outlined in Table 1.

Table.III.1 Classification of different faults of IM

TYPES OF FAULTS	SYMBOLES	CODES		
		S1	S2	S3
HEALTHY STATE	HS	0	0	0
SINGL_PHASE CUT **A** OFF	SPDA	1	0	0
SINGL_PHASE CUT **B** OFF	SPDB	0	1	0
SINGL_PHASE CUT **A** OFF	SPDC	0	0	1
TOW_PHASE CUT **AB** OFF	DPDAB	1	1	0
TOW_PHASE CUT **AC** OFF	DPDAC	1	0	1
TOW_PHASE CUT **BC** OFF	DPDBC	0	1	1

III.3.1 Network and tree test:

For both artificial neural network (ANN) and decision tree building blocks, a structured approach is paramount to ensure effectiveness in classification tasks.

III.3.1.1 Artificial Neural Network and tree Building Blocks:

Multilayer perceptrons and decision tree, within the domain of ANNs, have demonstrated efficacy in shape classification and regression tasks. In our testing, we employed a multilayer network utilizing the backpropagation algorithm for learning. This algorithm iteratively adjusts synaptic weights to minimize the average quadratic mean error, a fundamental aspect of learning in neural networks.

The network architecture consists of multiple layers, including an input layer representing the input variables (e.g., Va, Vb, Vc), an output layer corresponding to decision outcomes, and one or more hidden layers encapsulating internal representations of the problem variables.

Key steps in building of the training method include:

a. Choice of Inputs:

The number of the inputs is determined by the number of relevant variables. In our case, the cash values of variables (Va, Vb, Vc) dictate the number of input neurons, which corresponds to three variables.

b. Choice of Outputs:

Output determination involves defining the number of outputs and their representation. To facilitate interpretation, we opted for binary outputs (0, 1), aligning with the binary nature of the problem. The output function is linear, and the activation function is typically a sigmoid function.

c. Determination of Hidden Neurons and Layers:

The number of hidden neurons and layers is determined through iterative testing and error analysis of the learning algorithm.

d. Decision Tree Building Blocks:

Decision trees offer a transparent and interpretable framework for classification tasks. Key elements of building decision trees include:

e. Feature Selection:

The decision tree algorithm selects the most relevant features (variables) for decision-making. In our context, features such as voltage values (Va, Vb, Vc) serve as inputs.

f. Node Splitting:

Nodes in the decision tree are split based on selected features to maximize information gain or minimize impurity. This process continues recursively until a stopping criterion is met.

g. Tree Pruning:

Pruning techniques may be applied to reduce the size of the tree and improve generalization performance.

h. Output Interpretation:

Decision tree outputs are easily interpretable, often representing class labels or decisions based on feature thresholds.

By following these steps, both neural network and decision tree models can effectively classify and interpret data, contributing to the diagnosis of electromechanical system failures.

III.3.2 Signal processing by RMS:

The RMS transform is applied for the “abc” stator current signals where the lack of phase or three unbalanced phases will therefore be visible the current signals which are signal’s fault indicators.

III.3.3 Interpretation of the results:

To effectively detect errors in a system, a robust diagnosis method is necessary. One approach involves using neural networks an decision tree, which require a substantial number of examples during training. This includes both healthy operational cases and instances with defects. During training, the learning function of the neural network is exposed to these examples, allowing the network to learn to map input data to the corresponding diagnostic output. After training, the network can not only recognize the specific examples it has learned but also generalize to new, similar instances. This generalization capability provides a certain degree of robustness against standard signal distortions when detecting defects.

At time t=1st = 1st=1s, consider the neural network detecting the first error with outputs S1, S2, S3 having values (1,0,0). This output configuration might indicate a specific type of defect, such as a single-phase cutoff, and can be distinguished from other defect cases based on the pattern of the outputs.

In addition to neural networks, decision trees can also be employed for system diagnosis. Decision trees work by recursively splitting the data based on feature values, creating a tree structure where each leaf node represents a diagnostic decision. They are particularly useful because they provide a clear and interpretable model for decision-making.

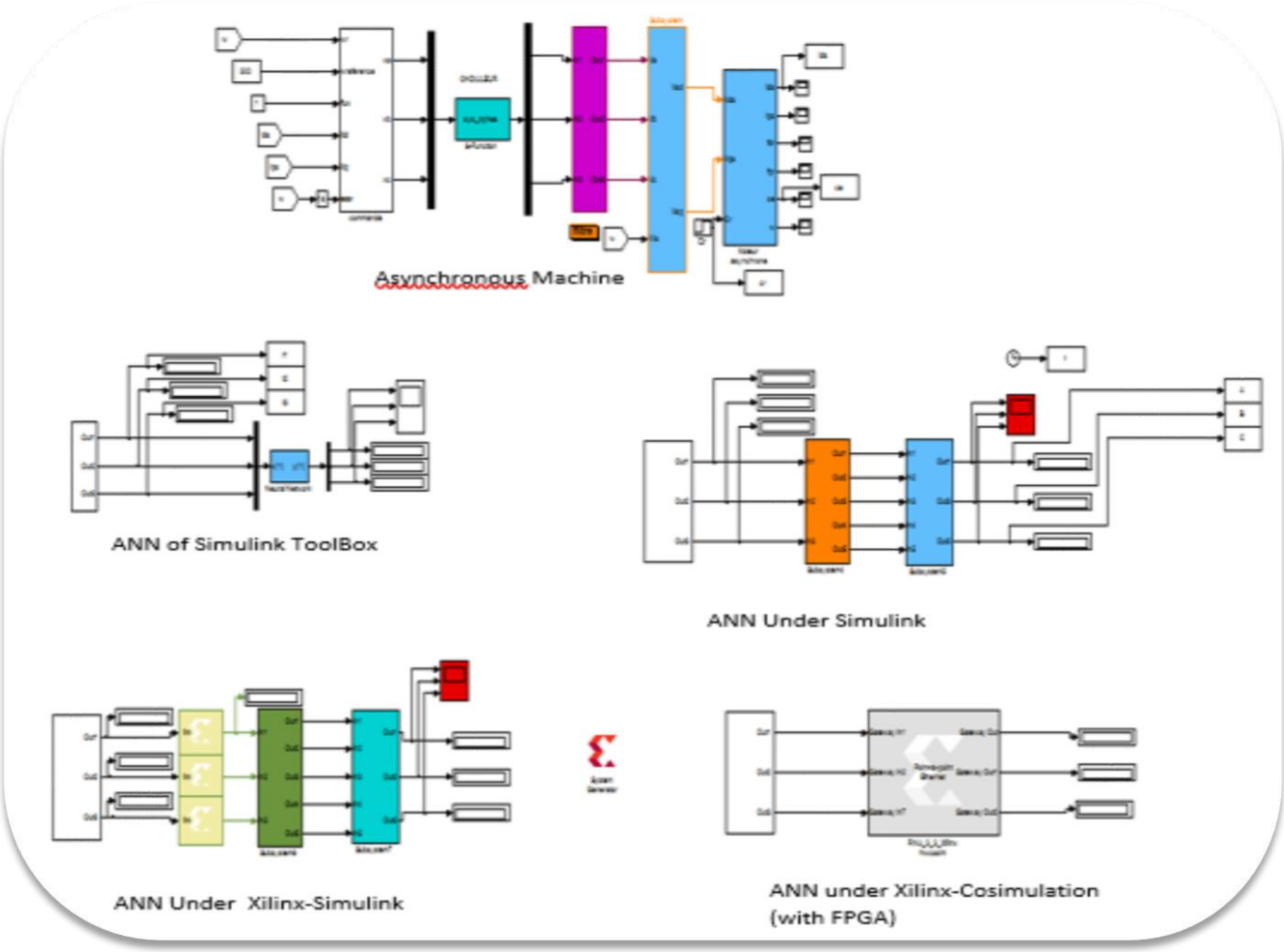


Figure.III.1 ANN and decision tree based simulation with several tools (Simulink, Xilinx, Cosimulation).

III.4 ANN and decision tree testing by co-simulation:

Once the ANN and decision tree code is deployed on the FPGA board, the entire system can be simulated as depicted in Figure III.3. To evaluate the performance of the proposed ANN, including the approximated activation sigmoid function, the same validation scenario used previously for the ANN has been employed in this test. The results, illustrated in Figure III.4, show the instantaneous stator current, the RMS stator current, and the three binary outputs from the ANN that identify the fault type.

It is evident from the results that the implemented ANN and the decision tree correctly generates the binary code corresponding to the specific fault applied to the motor. This indicates the accuracy and reliability of the method in fault detection. During motor startup, the ANN and decision tree output can be easily filtered or ignored, as the transient signals do not last long enough to be mistaken for consequential faults.

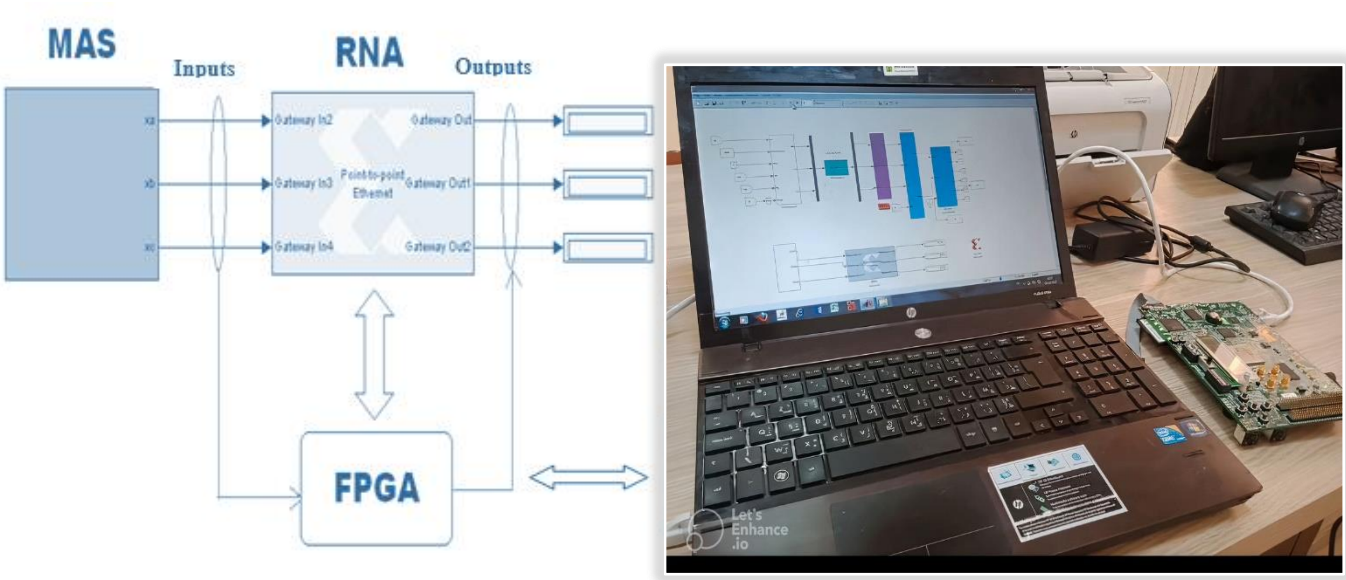


Figure.III.2 Principle of implementation of ANN block on FPGA using Co-simulation (Co-simulation of ANN block with Virtex4 Device)

Finally, Figure III.4 illustrates the integration of the induction motor Simulink model with the artificial neural network (ANN) and the decision tree system modeled in Xilinx. The System Generator is utilized to convert the Xilinx model (ANN) and the decision tree, generate the corresponding VHDL code, and download it onto the FPGA processor board (Virtex4 XC4VSX35-1011668). The entire system setup is depicted in Figure 12. The FPGA board is connected to MATLAB/Simulink on the PC via an Ethernet cable, enabling seamless communication and control.

Table.III.2 Times of faults application

Time of application	Fault type	Code
t =1s	Single-phase cut A	SPDA
t = 2s	Healthy State	HS
t = 3s	Single-phase cut B	SPDB
t = 4s	Healthy State	HS
t = 5s	Single-phase cut C	SPDC
t = 6s	Healthy State	HS
t =7s	Unbalance single-phase2	DPDAB
t = 8s	Healthy State	HS
t =9s	Unbalance single-phase2	DPDAC

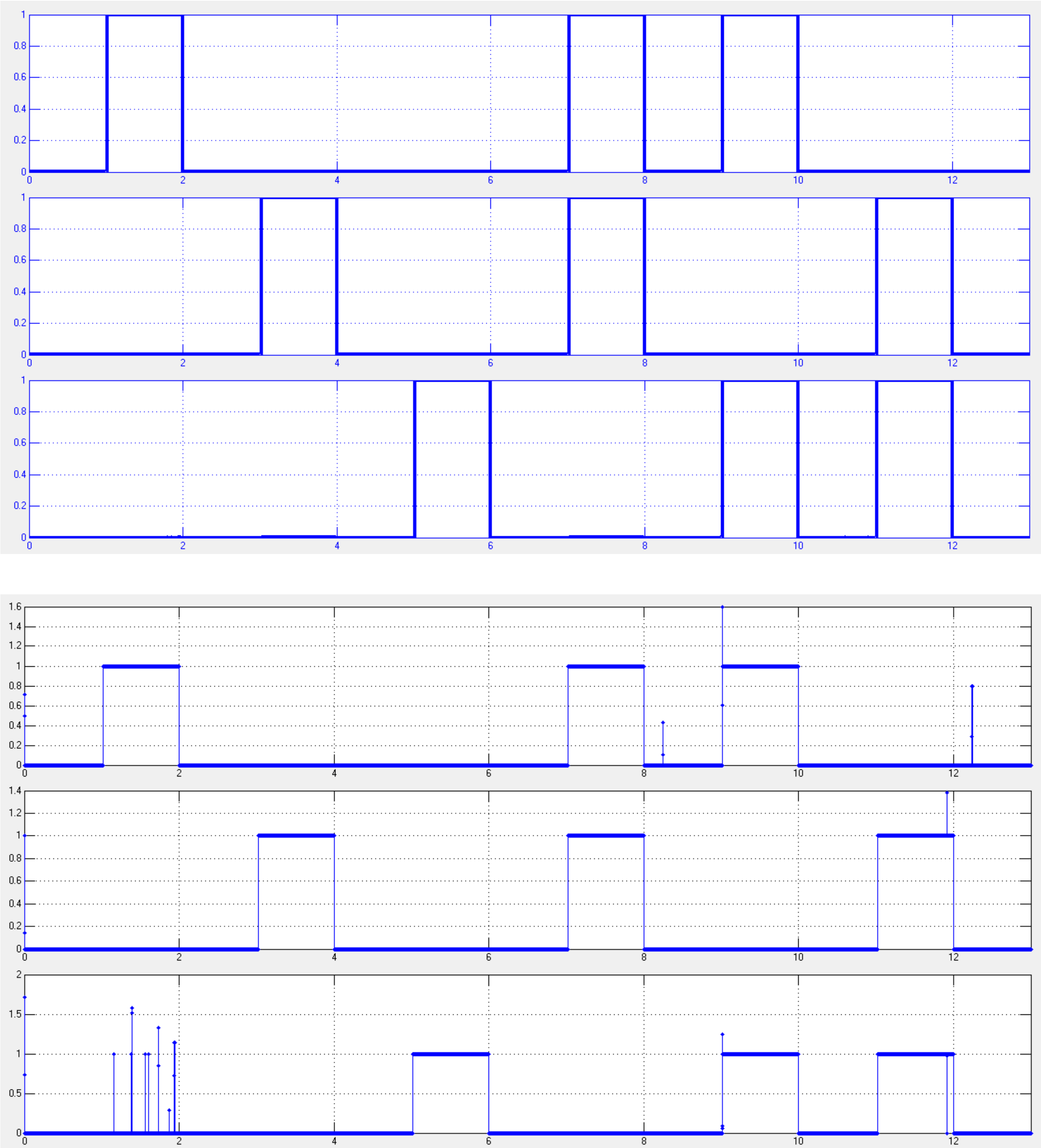


Figure.III.3 The stator voltages with different faults and the ANN outputs

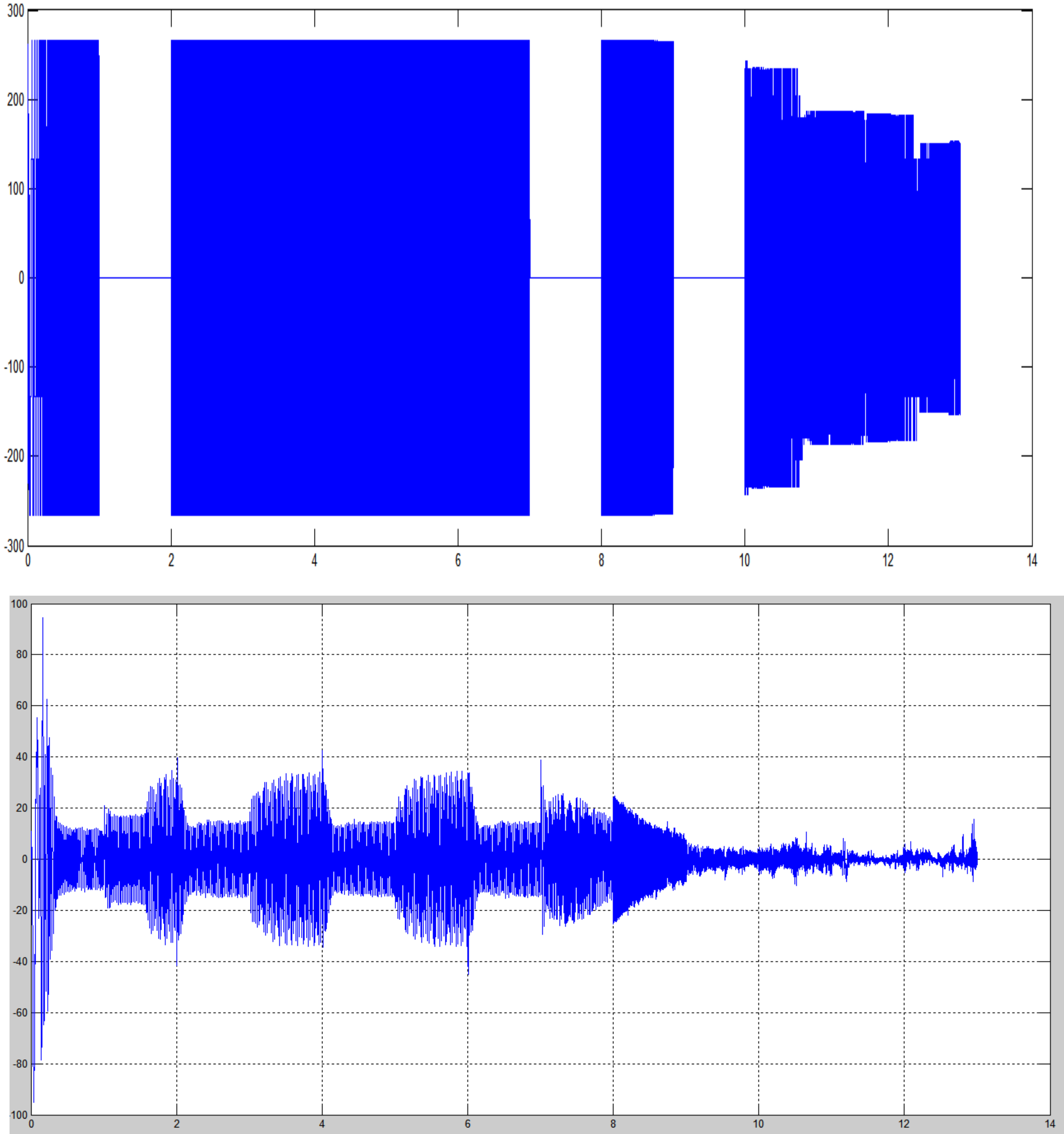


Figure.III.4 The stator voltages with different faults application

Table.III.3: Results of synthesis of obtained ANN Block under Xilinx.

Device utilization summary: Selected Device : 4vsx35ff668-10		
Number of Slices	113 out of 15360	0%
Number of Slices Flip Flops	113 out of 30720	0%
Number of 4 Inputs LUTs	206 out of 30720	0%
Number of Bonded IOBs	60 out of 448	13%
Number of GCLKs	6 out of 32	18%

From the results obtained on the table.3 reveals that the VHDL code occupies an area of less than 1% on FPGA, means that the FPGA type Virtex4 XC4VSX35- 1011668 supported widely this VHDL code [3]

III.5 Conclusion:

In this chapter, we have studied a neural network that uses simple inputs such as The RMS values for the three stator voltages. Moreover, in order to implement the neural network, we first went through several parametric studies (input selection, output selection, etc.). In this chapter we also implemented a neural network based diagnostic system on FPGA.

In this work, an ANN has been developed and implemented for purpose of diagnosis of induction machine faults. The proposed ANN has one five neurons hidden layer, three binary outputs and three inputs which are the RMS values of the three motor voltages. As the input – targets data for motor diagnosis are available, Levenberg-Marquardt backpropagation algorithm of Neural Network Toolbox has been used to train he ANN. The results have shown through the validation test that the obtained synaptic weights of the ANN are the optimal values by obtaining a very small mean squared error.

One of the most important parameters in the implementation procedure is the modeling of the activation sigmoid function which is very commonly used in ANNs. In this work a method has been applied to approximate the sigmoid function by 2nd order polynomial equation resulting in less computation time and memory requirements.

The system generator has been used to generate the VHDL code corresponding to the ANN model and downloaded into the FPGA Vertex 4 board. Co-Simulation of the induction motor modeled using Simulink blocks and the ANN running in the FPGA has been successfully done. The obtained ANN outputs show the effectiveness of the proposed topology and the approximation of the activation sigmoid function.

The use of high-level design tool such as system generator is very beneficial for the verification and design of any complex diagnosis or control algorithm without using a real system which can be damaged by the algorithm itself.

General conclusion

General Conclusion:

In this thesis, we have developed an intelligent diagnostic system for electromechanical systems, specifically targeting induction motors. Our study began with an exploration of various types of faults in these machines, their causes, and the diagnostic methods applicable to each type. We highlighted the flexibility and effectiveness of using artificial neural networks (ANNs) and decision tree techniques for monitoring and controlling production lines.

We conducted modeling and simulation of both the asynchronous motor and the diagnostic system, employing a trilayered neural network and a fine decision tree. The neural network was designed with a hidden layer of five neurons, three binary outputs, and three inputs, corresponding to the RMS values of the motor voltages. For training the ANN, we used the Levenberg-Marquardt backpropagation algorithm from the Neural Network Toolbox. Validation tests indicated that the ANN achieved optimal synaptic weights, with a very small mean squared error, demonstrating high accuracy.

In parallel, we developed a fine decision tree model, which provided an alternative method for fault diagnosis. We compared the performance of the ANN and the decision tree in terms of accuracy and speed. Both methods were then implemented on an FPGA board, chosen for its superior performance relative to DSPs and microprocessors.

To implement the ANN model, we used the system generator to produce VHDL code, which was subsequently downloaded onto an FPGA Vertex 4 board. We successfully co-simulated the induction motor model created in Simulink with the ANN running on the FPGA. The outputs confirmed the effectiveness of our ANN topology and the accuracy of the activation sigmoid function. Similarly, we implemented and validated the decision tree on the FPGA, further confirming the robustness of our diagnostic system.

The use of high-level design tools such as the system generator proved invaluable, enabling the verification and design of complex diagnostic and control algorithms without risking damage to actual systems. This approach streamlined the development process and ensured the safety and reliability of the systems being diagnosed.

In conclusion, our intelligent diagnostic system, which incorporates both a trilayered neural network and a fine decision tree, demonstrates significant potential for enhancing the maintenance and reliability of electromechanical systems. By enabling early fault detection and prevention, this system offers a powerful tool for improving the operational efficiency and longevity of such systems.

Biblioghraphy

Bibliography :

[1] T. Wildi, Electrical Machines,Drives, and Power Systems, new jersey: Pearson education, 2002, p. 236.

[2] R. T. b. P. G. a, "Signal based condition monitoring techniques for fault detection and diagnosis of induction motors: A state-of-the-art review," *mechanical systems and signal processing,* no. 114, p. 106912, 2020.

[3] K. S. V. K. M. Siddiqui, "Health Monitoring and Fault Diagnosis in Induction Motor-A Review," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering,* vol. 3, p. 6552, 2014.

[4] M. M. S. S. H. M.-F. E. M, "A nove indicator of stator winding inter-turn fault induction motor using infrared thermal imaging," *Infrared Phys Technol,* no. 161, p. 330–336, 2013.

[5] V. A. Z. A. M. SA, "Detection of stator winding inter-turn short circuit in induction motor using vibration specified harmonic amplitude," in *chez 2nd International Conf. on Acoustic & Vibration, Sharif University of Technology*, Tehran, 2012.

[6] V. F. P. J. F. M. A. J. P. a. M. M. C. T. G. Amaral, "Image Processing to a Neuro-Fuzzy Classifier for Detection and Diagnosis of Induction Motor Stator Fault," in *33rd Annual Conference of the IEEE Industrial Electronics*, 2007.

[7] A. &. G. D. &. S. L. S. &. A. A. Choudhary, "Condition Monitoring and Fault Diagnosis of Induction Motors: A Review," *Archives of Computational Methods in Engineering,* vol. 26, p. 1226, september 2018.

[8] S. C. ,. M. M. ,. S. S. Subrata Karmakar, Induction Motor Fault diagnosis, Approach through Current Signature Analysis, Singapore: Springer, 2018, pp. 3,30.

[9] G. e. a. Gentile, "CURRENT PATTERN ANALYSIS TO DETECT INDUCTION MACHINE NON ROTATIONAL ANOMALIES," in *Proc. of International Conference on Electrical Machines (ICEM),* 1998.

[10] F. F. a. M. Martelli, "Development of Expert System Knowledge Base to On- Line Diagnosis of Rotor Electrical Faults of Induction Motors," in *IEEE Ind. Appl. Society Annual Meeting*, 1992.

[11] v. 3. n. %. p. 4.-4. 1. B. Y. a. G. B. Kliman, "An adaptive statistical time-frequency method for detection of broken bars and bearing faults in motors using stator current," *IEEE Transactions on Industry Applications,* vol. 35, no. 12, pp. 442-452, 1999.

[12] N. 2. 1. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery,* vol. 2, no. 2, June 1998.

[13] M. H. J. B. a. I. Y. H. G. E. Styvaktakisvol, "Expert system for classification and analysis of power system events," *IEEE Transactions on Power Delivery,* vol. 17, no. 12, pp. 423-428, 2001.

[14] M. Chow, Methodologies of Using Neural Network and Fuzzy Logic Technologies for Motor Incipient Fault Detection, Singapore: World Scientific, 1997.

[15] A. T. a. E. V. M. Kaviarasan, "Fault diagnosis of three phase squirrel cage induction motor due to bearing by using artificial intelligence," in *International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)*, 2016.

[16] J. S. N. I. T.I. Liu, "Detection of roller bearing defects using expert system and fuzzy logic," *Mech. Syst. Sig. Process,* vol. 10, no. 5, p. 595– 614, 1996.

[17] H. C. X. C. Z. H. Z. S. Z. Liu, "Multi-fault classification based on wavelet SVM with PSO algorithm to analyze vibration signals from rolling element bearings," *Neurocomputing,* p. 399–410, 2013.

[18] P. a. G. N. K. a. A. A. Jaiswal, "Comparative study of various training algorithms of artificial neural network," in *International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, 2018.

[19] I. N. D. Silva, D. H. Spatti, R. Andrade Flauzino, L. H. B. Liboni and S. F. d. Reis Alves, Artificial Neural Networks, A Practical Course, Switzerland: Springer, 2017.

[20] M.-C. e. a. Popescu, «Multilayer perceptron and neural networks,» *WSEAS Transactions on Circuits and Systems,* vol. 8, n° %17, pp. 579-588., 1 jul 2009.

[21] N. R. O'shea K, "An introduction to convolutional neural networks," *arXiv,* 26 nivember 2015.

[22] M. H. C. Sazlı, *A Brief Review of Feed-Forward Neural Networks*, Faculty of Sciences University of Ankara , 2006.

[23] P. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE,* vol. 78, no. 10, pp. 1550-1560, 1990.

[24] A. T. A. i. i. e. Goh, "Back-propagation neural networks for modeling complex systems," *Artificial intelligence in engineering,* vol. 9, no. 3, pp. 143-151, 1995.

[25] F. Schilling, *The effect of batch normalization on deep convolutional neural networks,* Stockholm, School of Computer Science and Communication, 2016.

[26] S. Demyanov, *Regularization methods for neural networks and related models,* University of Melbourne, Department of Computing and Information Systems, 2015.

[27] S. &. S. C. Ioffe, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International conference on machine learning*, 2015.

[28] R. a. M. M. Ribani, "A survey of transfer learning for convolutional neural networks," in *32nd SIBGRAPI conference on graphics, patterns and images tutorials (SIBGRAPI-T),* 2019.

[29] P. A. R. D. D. N. B. E. K. B. M.-L. L. N. M. T. e. a. Harrison, «Selecting methods for ecosystem service assessment: A decision tree approach,» *Ecosystem services,* vol. 29, pp. 481-498., 2018.

[30] W. D. K. A. M. a. D. M. A. Baswardono, «Comparative analysis of decision tree algorithms: Random forest and C4. 5 for airlines customer satisfaction classification,» *Journal of Physics: Conference Series,* vol. 1402, n° %16, 2019.

[31] O. C. Njoku, *Decision Trees and Their Application for Classification and Regression Problems,* Mathematics, 2019.

[32] A. S. a. V. C. VSRD-IJCSIT, «Comparison between ID3 and C4.5 in Contrast to IDS Surbhi Hardikar,» *VSRD-IJCSIT,* vol. 2, n° %17, pp. 659-667, 2012.

[33] B. M. A. E. H. &. E. M. Hssina, "A comparative study of decision tree ID3 and C4.5," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 12, pp. 13-19, 2014.

[34] H. B. &. G. Y. J. Wang, "Research on C4. 5 algorithm improvement strategy based on MapReduce," *Procedia Computer Science,* vol. 183, pp. 160-165, 2021.

[35] S. a. P. G. Singh, "Comparative study ID3, cart and C4. 5 decision tree algorithm: a survey," *International Journal of Advanced Information Science and Technology (IJAIST),* vol. 27, no. 27, pp. 97-103, 2014.

[36] P. J. D. a. M. K. R. Patidar., "Decision tree C4. 5 algorithm and its enhanced approach for educational data mining," *Engineering Universe for Scientific Research and Management,* vol. 7, no. 12, pp. 1-14, 2015.

[37] X. a. C. X. a. L. X. Zheng, "Visible-Infrared Person Re-Identification via Partially Interactive Collaboration," *IEEE Transactions on Image Processing,* vol. 31, pp. 6951-6963, 2022.

[38] K. L. S. &. V. G. Coussement, "A comparative analysis of data preparation algorithms for customer churn prediction: A case study in the telecommunication industry," *Decision Support Systems*, vol. 95, pp. 27-36, 2017.

[39] B. W. O. J. D. T. P. ,. D. D. P. Septia Devi Prihastuti Yasmirullah, "Modification of Multivariate Adaptive Regression," *Journal of Physics: conference series,* vol. 1863, no. 1, p. 012078, 2021.

[40] J. Friedman, "Multivariate Adaptive Regression Splines," *The Annals of Statistics*, vol. 19, pp. 1-141, 1991.

[41] P. D. E. a. L. W. Geurts, "Extremely randomized trees," *Machine learning,* vol. 63, pp. 3-42, 2006.

[42] M. K. D. Eddine, *Elaboration d'un Système Intelligent de Surveillance et de,* Boumerdes, Génie Electrique, 2007, pp. 61-65;155.

[43] A. P. P. ,. S. Ampulembang, *Development of Continuous Birespon Nonparametric Regression Model Using MARS,* Surabaya: PhD Thesis Institut Teknologi Sepuluh Nopember, 2017.

[44] M. S. Team, "Mathworks," 2023. [Online]. Available: https://www.mathworks.com..

[45] T. M. L. Alvarez, "Don't Lie to Me: Avoiding Malicious Explanations With STEALTH," *IEEE Software,* vol. 40, no. 3, pp. 43-53, 2023.

[46] F. M. R. R. K. M. H. A. Y. a. A. H. S. Javed Mehedi Shamrat, "Performance evaluation among id3, c4. 5, and cart decision tree algorithm," in *Pervasive Computing and Social Networking: Proceedings of ICPCSN,* Singapore, 2021.

[47] Q. &. D. D. M. He, "Fault diagnosis of induction motor using neural networks," in *International Conference on Machine Learning and Cybernetics*, 2007.

[48] N. R. M. &. H. H. Yassa, «Motor current signature analysis for the air gap eccentricity detection in the squirrel cage induction machines,» *Energy Procedia*, vol. 162, pp. 251-262, 2019.

[49] I. K. D. &. C. S. Chouidira, « Induction Machine Faults Detection and Localization by Neural Networks Methods,» *Revue d'Intelligence Artificielle,* vol. 33, n° %16, pp. 427-434., 2019.

[50] S. S. R. B. D.E.Khodja, «Implementation of Optimized Approximate Sigmoïd Function on FPGA Circuit to use in ANN for Control and Monitoring,» *Control Engineering and Applied Informatics Journal*, vol. 17, n° %12, pp. 64-72, 2015.

[51] A.E.Aib, *Implémentation des techniques d'intelligence artificielles FPGA en vue de contrôle des systèmes d'entrainements électriques,* Université Mohamed Boudiaf d'M'sila, 2022.

[52] D. E. K. S. C. Abdelghani Aib, «Field programmable gate array hardware in the loop validation of fuzzy direct torque control for induction machine drive,» *Journal of Electrical Engineering & Electromechanics,* pp. 28-35, 2023.

[53] D. E. K. S. C. Abdelghani Aib, «FPGA Hardware in the Loop Validation of Torque and Flux Estimators for Direct Torque Control (DTC) of an Induction Motor (IM),» *Advances in Modelling and Analysis,* vol. 64, n° %11-4, pp. 9-16, 2021.