A Machine Learning Framework for Improving Diagnosability of a Reconfigurable Manufacturing System

Author

MUHAMMAD FAISAL RANA

Regn Number

00000119601

Supervisor

DR SAJID ULLAH BUTT

DEPARTMENT OF MECHANICAL ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENINGEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

SEPTEMBER, 2019

A Machine Learning Framework for Improving Diagnosability of a Reconfigurable Manufacturing System

Author

MUHAMMAD FAISAL RANA

Regn Number

00000119601

A thesis submitted in partial fulfillment of the requirements for the degree of

MS Mechanical Engineering

Thesis Supervisor:

DR SAJID ULLAH BUTT

Thesis Supervisor's Signature

DEPARTMENT OF MECHANICAL ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENINGEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

SEPTEMBER, 2019

Declaration

I certify that this research work titled "*A Machine Learning Framework for improving Diagnosability of a Reconfigurable Manufacturing System*" is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

Signature of Student

MUHAMMAD FAISAL RANA

00000119601

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Signature of Student

MUHAMMAD FAISAL RANA

00000119601

Signature of Supervisor

DR SAJID ULLAH BUTT

Copyright Statement

Acknowledgements

I am thankful to my Creator Allah Subhana-Watala to have guided me throughout this work at every step and for every new thought which You setup in my mind to improve it. Indeed I could have done nothing without Your priceless help and guidance. Whosoever helped me throughout the course of my thesis, whether my parents or any other individual was Your will, so indeed none be worthy of praise but You.

I am profusely thankful to my beloved parents who raised me when I was not capable of walking and continued to support me throughout in every department of my life.

I would also like to express special thanks to my supervisor Dr Sajid Ullah Butt for his help throughout my thesis and also for Manufacturing System Design and Computer Integrated Manufacturing courses which he has taught me. I can safely say that I haven't learned any other engineering subject in such depth than the ones which he has taught.

I would also like to pay special thanks to Dr Aamer Ahmed Baqai for his tremendous support and cooperation. Each time I got stuck in something, he came up with the solution. Without his help I wouldn’t have been able to complete my thesis. I appreciate his patience and guidance throughout the whole thesis.

I would also like to thank Dr Hasan Aftab Saeed, and Dr Naveed Akmal Din for being on my thesis guidance and evaluation committee.

Finally, I would like to express my gratitude to all the individuals who have rendered valuable assistance to my study.

Dedicated to my exceptional parents and adored siblings whose tremendous support and cooperation led me to this wonderful accomplishment.

ABSTRACT

Reconfigurable Manufacturing Systems (RMS) effectively respond to fluctuating market needs and customer demands for finished product. Diagnosability is a supporting characteristic of RMS that has a say in the quality of finished product. Cost and time taken for manufacturing are also considerably affected if proper *diagnosability* measures are not taken. Previous studies on Diagnosability of RMS have been studied from Axiomatic System Theory as such Design For Diagnosability (DFD). Nevertheless Diagnosability remains to be the least studied characteristic of RMS. With the availability of digitized data, Machine Learning approaches to advance manufacturing have proven to be considerably effective. A research gap existed for the application of Machine Learning techniques in improving the Diagnosability of RMS. A framework of Machine Learning has been proposed to address this gap. The working of the framework has been illustrated by two demonstrations from the available datasets, one in identifying proper signals in semi-conductor manufacturing to predict excursions, and the second in predicting machine failures due to a variety of factors. The framework is rendered in a concurrent-engineering fashion. The framework is tested against two available manufacturing datasets. Increase in Diagnosability will decrease the cost and time taken to production.

Key Words: Reconfigurable Manufacturing Systems, Machine Learning, Artificial Intelligence, Preventive Maintenance, Intelligent Manufacturing

Table of Contents

List of Figures

List of Tables

List of Abbreviations

RMS	Reconfigurable Manufacturing Systems
DML	Dedicated Manufacturing Systems
FMS	Flexible Manufacturing Systems
GT	Group Technology
CM	Cellular Manufacturing
ML	Machine Learning
AI	Artificial Intelligence
DFD	Design For Diagnosability
GID3	Generalized Identification algorithm 3
CSV	Comma Separated Values file
WEKA	Waikato Environment for Knowledge Analysis

CHAPTER 1: INTRODUCTION

1.1 Manufacturing Systems

The term Manufacturing is used for anything hand-made by humans and by extension anything made by a system which was made by humans in the first place. In modern times, humans depend on machines for most of their tasks and manufacturing is no exception. It is in this realm that manufacturing is referred to a production process carried out mainly in industries in all countries across the world. Manufacturing is an activity to add value to the raw materials using forces of nature to make a product that solves some problem human(s) are facing. A manufacturing system can be termed as an arrangement of closely interacting agents called machines or tools working in time and space to meet a customer demand in an efficient manner. It is therefore necessary to come up with all the scenarios a manufacturing system will be facing for making products in its lifetime. Traditionally, manufacturing has always involved human intervention and oversight and there is no manufacturing system in practice to date which is totally autonomous.

1.2 Different types of Manufacturing Systems

1.2.1 Dedicated Manufacturing System

Dedicated Manufacturing Systems were the first to arrive and follow from the mechanized and standard production of unvarying and unchanging products such as gun barrels or wheel of vehicle or any component for which the demand will not change. It focuses on large production orders. If the system runs for a long time, production costs keep on decreasing resulting in the profit for manufacturing enterprise. However, the limitation of DMS is that the production of variety of products is not possible. In other words, DMS is a highly inflexible system.

1.2.2 Cellular Manufacturing

Cellular Manufacturing takes into account Group Technology (GT) which takes into account the similarities of parts to be produced and combine then into groups called Part

Family. This helps to achieve more production with lesser machines. The time taken to produce items in this way is also expected to decrease. Each cell consists of machines assigned to carry out Part Family operations. The concept of GT enables FMS and RMS.

1.2.3 Flexible Manufacturing System

Flexible manufacturing system (FMS) is a GT machine cell which is controlled by its software component and has high production rate. The fact that FMS are software controlled with CNC's at the operational end, the kinds of task FMS can perform are highly flexible. This enables FMS to respond to market need but a major drawback of FMS is that they are costly to operate.

1.2.4 Reconfigurable Manufacturing Systems

The Reconfigurable Manufacturing Systems (RMS) were designed in the 1990s to respond to the fluctuating product demands and changing market conditions. RMS is a compromise between Dedicated Manufacturing Lines and Flexible Manufacturing System – the former has a high throughput but a low variation in product features, the latter has low throughput but a high variation in product features.

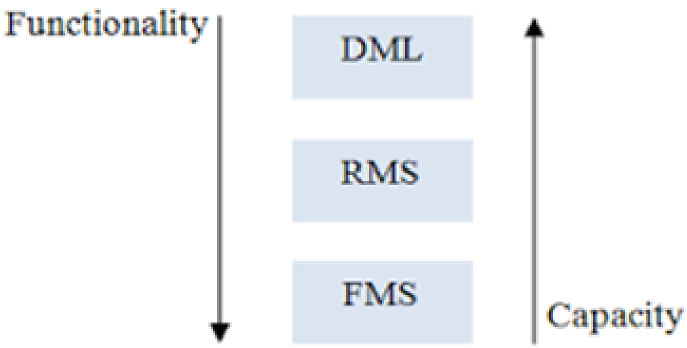

Figure 1.1 Comparison between the functionality and capacity of DML, RMS, FMS

Yorem Koren (1999) defines RMS as, [1]

"A Reconfigurable Manufacturing System (RMS) is designed at the outset for rapid change in structure, as well as in hardware and software components, in order to quickly adjust production capacity and functionality within a part family in response to sudden changes in market or in regulatory requirements."

The RMS consists of the following characteristics:

- Modularity

- Integrability
- Customized flexibility
- Scalability
- Convertibility
- Diagnosability

1.2.4.1 Diagnosability

Diagnosability remains to be the least studied characteristic of RMS despite having a say in the quality of products being manufactured in RMS and hence optimizing the cost and time of production.

The definition of the characteristic Diagnosability as defined by the pioneering author of RMS, Y. Koren [1],

> *"The ability to automatically read the current state of a system for detecting and **diagnosing** the root-cause of output product defects, and subsequently correct operational defects quickly."*

From the Principle 5 of RMS, also from the author of RMS, it flows that,

> *'The RMS possesses hardware and software capabilities to cost-effectively respond to unpredictable events — both external (market changes) and intrinsic events (machine failure)'*

It is important to mention here that RMS has yet to be implemented in its entirely in the industry but a few aspects of RMS have been applied in aerospace, automotive and semiconductor manufacturing.

The approaches to improve the diagnosability of a manufacturing system prior to RMS are summarized in the following table,

Approach	System	Description	Shortcoming
JIDOKA	Toyota Production System (TPS)	Automated and autonomous sensors to monitor manufacturing defects	Low margin for error
POKA-YOKE	Toyota Production System (TPS)	Avoidance of accidents that can lead to faulty manufacturing	Only prevents human errors
Six Sigma	General Electric, Motorola	Proactive approach to reduce error	Costly, low margin of error

NDT	Independent quality control	Involves a range of physics based techniques	Does not detect fault related to or arising in machine tools, is not real time, time consuming
Human inspection	Manual	Requires work force	Human inspection is a slow and inaccurate process

Table 1.1 Diagnosability along various manufacturing traditions

Owing to the deficiencies in the previous methodologies for detecting fault in manufacturing, RMS is advantageous because it addresses Diagnosability from two aspects,

- *detecting <u>machine failure,</u> and*
- *detecting <u>unacceptable part quality</u>*

Machines specialized in inspecting the defects of manufacturing has been proposed by Koren as,

> *The RMS must also be designed with product quality measurement systems as an integral part. <u>Reconfigurable Inspection Machine (RIM)</u> embedded in the RMS enables quick detection. These measurement systems are intended to help identify the sources of product quality problems in the production system rapidly, so they <u>can be corrected utilizing control methods, statistics, and signal processing techniques.</u>*

In order to make the holistic fault detection methodology of RMS real-time, it has to be equipped with the recent advances in artificial intelligence such as the machine learning techniques. It can now be postulated that Diagnosability in RMS ought to have three legs.

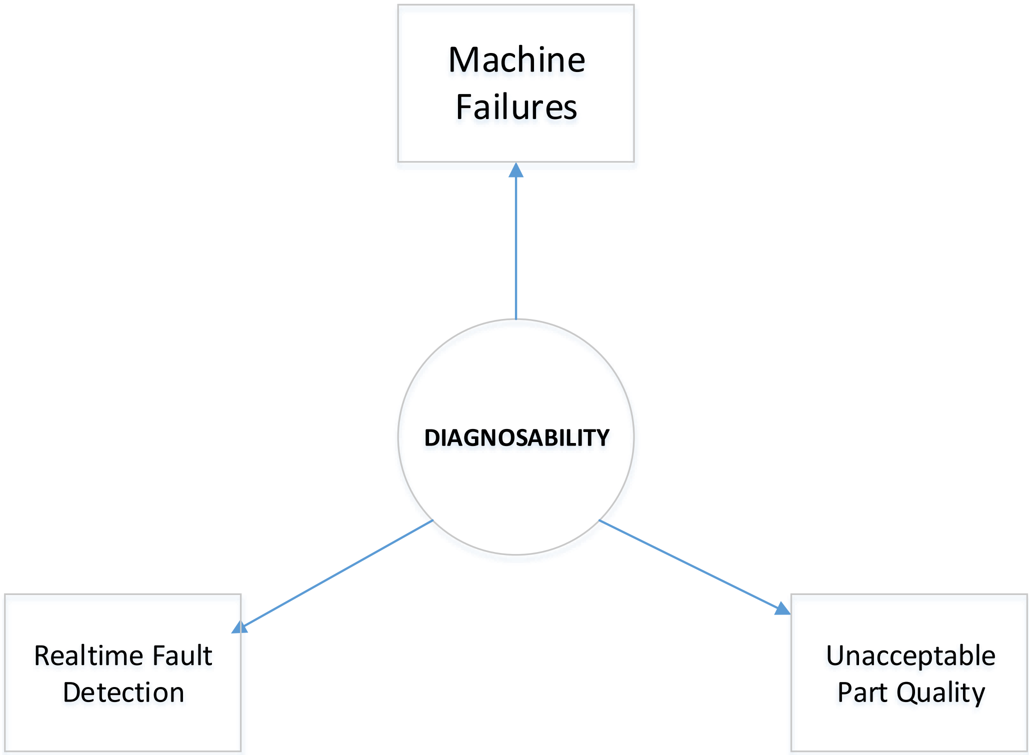

Figure 1.2 Three legs of Diagnosability in RMS

1.3 Industry 4.0

With the advent of Industry 4.0 and increased product varieties and fluctuating customer demands, manufacturing companies are finding ways to reduce cost and time needed for production while upping the quality of product. Due to the computerization of manufacturing systems, there is an increase in the digitized data. Industry 4.0 is being referred to by different terminologies in different part of the world: Advanced Manufacturing (USA), Smart Factory (South Korea), Vision China 2025 (China).

Big Data is the name given to the increase in data because it comprise of various formats, and as such, the availability of data on quality enables engineers to analyze the data for insights into process improvements, and diagnosing quality problems. This leads to better manufacturing outcomes.

Stages of industrial evolution is shown in the **Figure 1.3**. Industry 1.0 was the point in time when mankind made use of steam power to drive locomotives and engines in factories. The production of steam lead to electricity production. This conversion of energy made possible the second industrial revolution. Availability of electricity boosted research in electronics through which a degree of automation was achieved forming the third stage of industrial revolution. Most of the industry nowadays lie at Industry 2.0 and Industry 3.0.

However, miniature transistors enabled micro-electronics and nano-electronics developed and made possible the semiconductor manufacturing and the availability of computer systems that could be connected in the form of network.

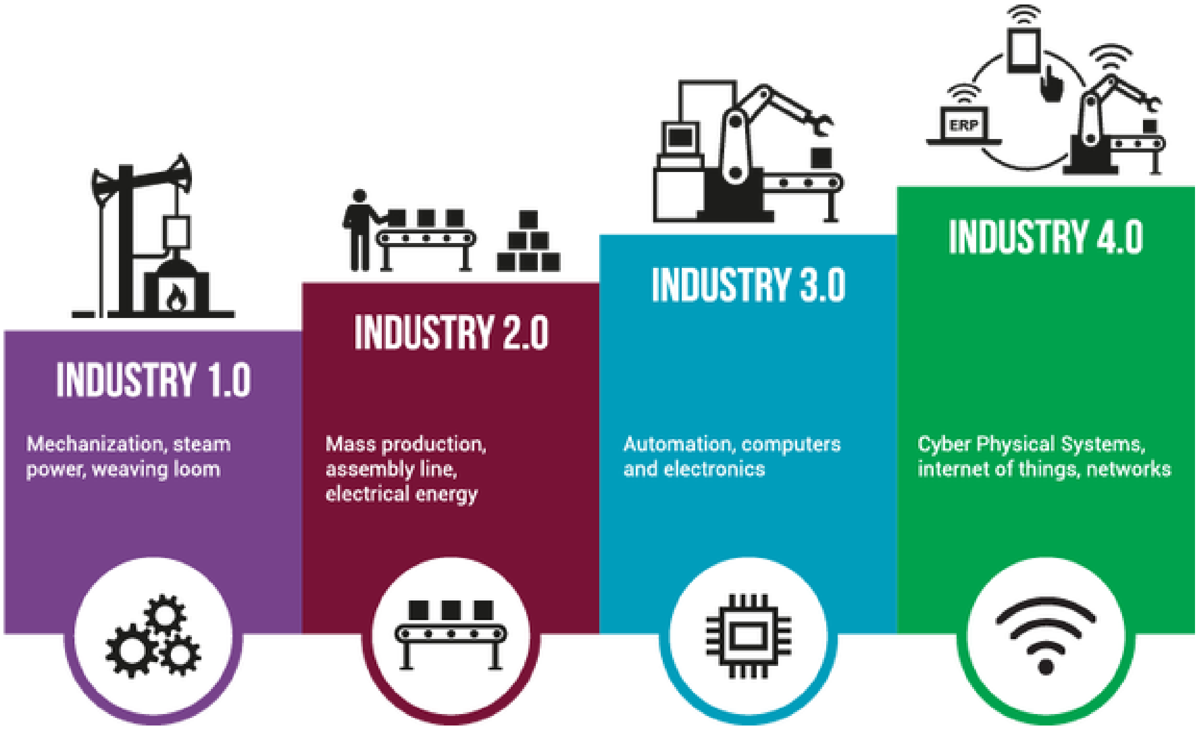

Figure 1.3 Evolutionary stages of industrial progress

It is very difficult to move from Industry 2.0 to Industry 3.0 due to the unavailability of good datasets that are clean because at this stage there are not AI systems available to magically improve the process, and as such industries cannot move from 2.0 to 4.0. Cisco Survey reveals that 75 % of IOT projects that transition from Industry 2.0 to Industry 4.0 fail [10].

The journey from Industry 2.0 to Industry 4.0 start from finding problems or issues that a manufacturing company thinks can be better solved by AI techniques. This process is very manual, and generally goes as follows,

a) Finding issues and problems

b) Having engineers to identify root cause of the problem (this takes a long time)

c) Figuring out the solution

d) Validating the solution and putting the solution to work

However, this process is very inefficient, and porous because the manufacturing enterprise is relying on humans to find the problem issues. So, there is an opportunity to try to automate this process which will complete the process ‘**a**’ to ‘**d**’ faster so that product can be dispatched to the customer faster and have a higher quality. To accelerate this, the organization needs detailed dataset which is everything about the process in a clean format that is able to be processed. Today, engineers collect this data and put it in the form of spreadsheet, and perform statistical processes and inferences on it and get results. This process can be automated but it relies on clean data. However, it’s very difficult for humans to perform statistics on these datasets. Algorithms are needed in order to derive insights from the datasets.

1.4 Machine Learning

Machine Learning is a technique used in Artificial Intelligence - statistical inferences with emphasis on learning from human nervous system. Machine Learning can handle tasks of NP-complete nature.

The difference between traditional programming approaches and machine learning based approaches is highlighted in the **Figure 1.4.**

The application of Machine Learning in improving production quality is under-explored and as such fraught with many opportunities as well as challenges.

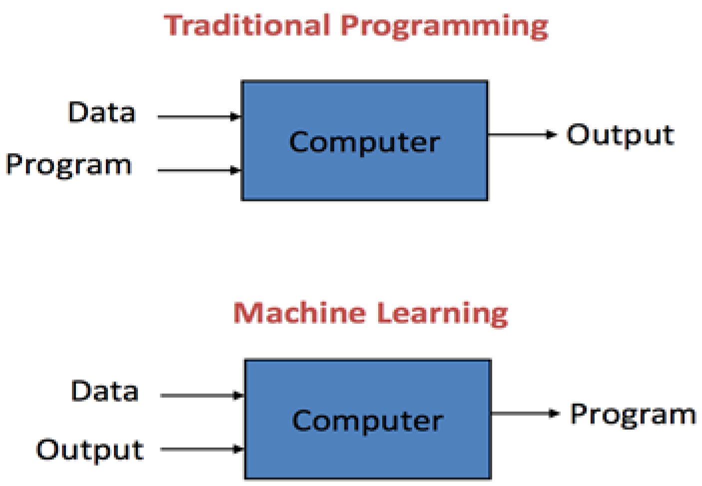

Figure 1.4 Difference between traditional programming and Machine Learning approaches

The Machine Learning is accomplished by performing statistical inferences on the available dataset in such a way that it results in an algorithm called the 'model' designed to optimize a specific output for that dataset as explained in the **Figure 1.5**.

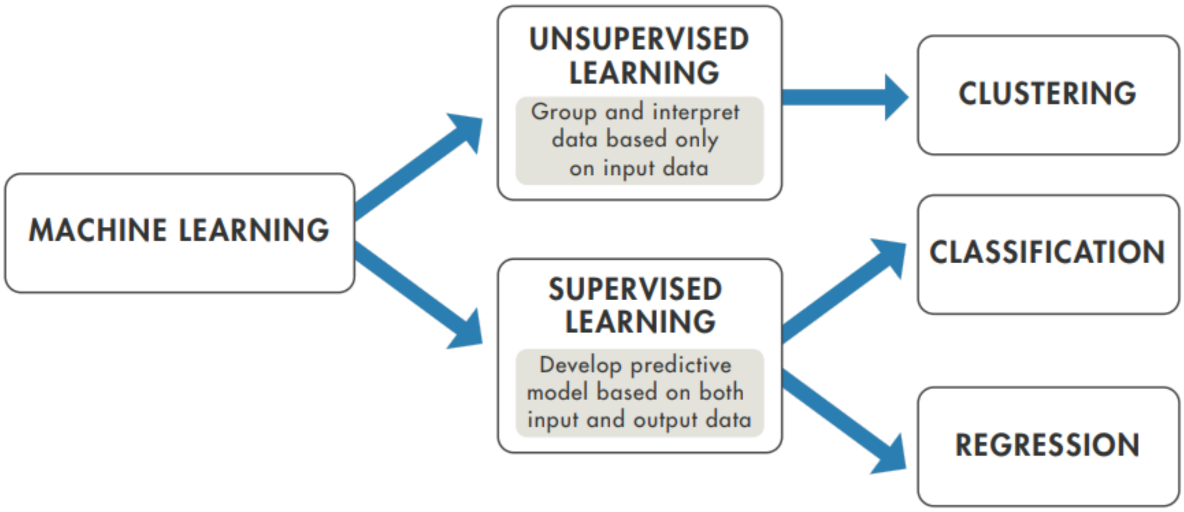

Figure 1.5 Machine Learning methods

Only the classification-based Supervised Machine Learning instances have been considered in this research due to unavailability of an unlabeled manufacturing dataset.

Artificial Intelligence and Machine Learning methods can be employed to model manufacturing processes quickly and easily as compared to any other available technique. Such modeled manufacturing processes can be further used to automate parameters related to machining processes for the optimization of process performance.

1.5 Motivation for research

Machine Learning methods have multi-disciplinary applications. In the wake of growing field of Artificial Intelligence whose major component is currently being practiced under the rubric of Machine Learning, developing the manufacturing systems to be more responsive to the recent advances in Machine Learning is the need of hour. This will ultimately lead to the rampant production of quality products thereby increasing the profit for manufacturing enterprises in a competitive world. An acceptance of latest trends in intelligent computing also prepares the manufacturing systems of today to brace up for the imminent advances of tomorrow.

Henceforth, my motivations for carrying out this research are enumerated as follows,

1. To keep manufacturing systems & practices up-to-date with the latest research in artificial intelligence with the effect to further the computerization of manufacturing systems.
2. To improve quality of life by increasing the efficiency and productivity manufacturing systems which has a direct effect on customer satisfaction, and profit maximization of manufacturing enterprises.
3. To prepare manufacturing systems welcome newer advancement in technology which is only possible if the degree of automation in manufacturing systems is increased, for example, the development of Autonomous Manufacturing Systems (Industry 5.0) which is in the offing.

1.6 Aim of Research

A detailed study of literature revealed that distributed approaches for solving problem of fault diagnosis are present.

1.7 Research Question

Could approaches in literature be used to develop the bigger picture - a consistent and structured framework which consolidates the existing solutions as well as provide a platform for the solution of unforeseeable problems of diagnosability in a manufacturing system, especially the RMS? In this research, a possibility for forming such a framework is investigated and proposed.

1.8 Thesis breakdown

Chapter 2 discusses the latest advancements in RMS and ML.

A survey of prevailing data-collection techniques in advanced manufacturing is done in Chapter 3.

The proposed framework is discussed in detail in Chapter 4.

The methodology and the workflow discussed in Chapter 4 is finally applied on two select datasets and the results discussed in Chapter 5.

Chapter 6 includes a summary and recommendations of findings along with future directions.

CHAPTER 2: RECONFIGURABLE MANUFACTURING SYSTEMS, MACHINE LEARNING – STATE OF THE ART

Anticipating the imminent uncertainties and violent fluctuations in demand-supply of twenty-first century for manufacturers, Yorem Koren in a keynote paper pioneered the idea of Reconfigurable Manufacturing Systems that help manufacturing companies to stay competitive. The goal of RMS is to manufacture products in a cost-effective manner while being sensitive to the market changes. The RMS was supported by its enabling components namely Reconfigurable Machine Tools, Reconfigurable Inspection Machines, Reconfigurable Software [1].

The Stream of Variation theory is proposed by Koren in support for his Diagnosability theory of RMS which takes into account Six-sigma quality control with the placement of sensors [1].

To deal with the internal and external uncertainties arising in manufacturing environments, the concept of Biological Manufacturing Systems (BMS) was proposed. The BMS was inspired from the ideas of evolution and adaptation of organisms. The authors propose the combination of evolutionary function and learning function to deal with the challenge of complex and real-time decision making during manufacturing [2].

Ding et al. (2002) took the concept of Stream of Variation theory forward and debated that sensors-placement is a costly affair. He came up with an algorithm that finds sensors the most suitable positions for a multistage manufacturing. This approach made data obtained through sensors more reliable [3]

Monotstori (2003) proposed that various AI techniques such as pattern recognition, artificial neural networks, fuzzy systems be integrated for problem as complex as manufacturing in which uncertainties arise. His approach is considered as a hybrid one but it does not talk about quality and diagnosability issues in manufacturing in general [4].

The most dedicated study to consider Diagnosability is done by Liu et al. (2004). He employed Axiomatic Theory approach to improve Diagnosability of manufacturing systems. Design For Diagnosability (DFD) approach assumes Diagnosability issues as part of the design of manufacturing system and recommends that measures that improve the quality issues should be incorporated at the outset of designing manufacturing systems. In his approach a diagnosability matrix is made to render clear the relationship between quality attributes and diagnosis. This helps to establish the conditions for optimal diagnosability. [5]

The ultimate purpose of machine learning is to give the software instructions responsible for operating the manufacturing system. Machine learning helps computer algorithms improve their performance over time as more and more training data becomes available for training of these algorithms. As such, behind these algorithms, is present, in the form of machine learning a ‘decision tree’ which is a mental framework existing in the mind of data-engineer and according to their experience and judgment as applied to the present activity of manufacturing system in achieving a desired product quality and production capacity.

The concept of Machine Learning is as old as 1950s. However, its application to the wider domain of knowledge has witnessed an exponential increase only recently. The application of Machine Learning methods to manufacturing is fraught with challenges and opportunities [6].

To improve and automate the semiconductor manufacturing, Irani et al. developed a Generalized ID3 machine learning algorithm as early as 1993. The GID3 trains itself with the help of training data and generates a decision tree capable of predicting the result of further experimentations taking into account general and varied conditions. The nodes in this tree correspond to the properties or specifications (design criteria) of the semiconducting manufacturing task .The tree can then be consolidated into an expert system governing the future production of semiconductor. This technique was found helpful in process diagnosis, and process optimization [7].

Pham (2005) evaluates several machine learning techniques and their successful applications in manufacturing. He uses inductive learning which is an old name for supervised machine learning is. He also review the advanced in machine learning approaches and presents challenging facing the machine learning in manufacturing area. He emphasizes the need to include more data formats like image and text and be made the part of data-mining for manufacturing [8].

Wuest et al. (2016) stressed that manufacturing systems are dynamic and the production is a non-linear processes that can even behave chaotic. Utilizing all means available to cater for any market need, machine learning is an area which gives promising results and usability. Because machine learning has the potential to answer newer and old challenges in manufacturing, it has the potential for becoming popular among researchers and practitioners. Machine Learning is not as fancy as it sounds since it involves many challenges and confusing approaches which come in the way of it being implemented globally [9].

Diagnosability remains to be least study characteristic of Reconfigurable Manufacturing Systems. A study gap was found in the application of machine learning methods for improving the *diagnosability* characteristic of RMS. The single largest hurdle witnessed in the way of is manufacturing data sharing and making data public. Machine Learning can solve NP-complete problems. In essence, ML techniques are a departure from equation based model of the problem, the algorithm in ML is called the model.

CHAPTER 3: AN OVERVIEW OF DATA-COLLECTION METHODS AND DATA-TYPES IN MANUFACTURING

The biggest challenge in using AI/ML to model manufacturing process behaviors is the facility to obtain training data for training the model. The primary way to collect data is through sensors. These sensors are properly placed to minimize errors arising due to environmental conditions.

All the manufacturing equipment used in assembly generate data. This data stays in manufacturer's shop floor system, or on the machines. If not saved, such data are usually overwritten because the working memory of manufacturing systems are not large.

3.1 Data Collection Techniques

3.1.1 High resolution imaging, 3D Laser imaging

This technique was initially developed to enhance computer vision. An high resolution image is a rich source of information. Different pixel densities can be chosen depending on the nature of manufacturing job being carried out.

3.1.2 Process measurement points

This technique was developed to validate six-sigma quality methodology. The data is not obtained in real-time fashion but in a interval fashion.

3.1.3 Reconfigurable Inspection Machines

This was proposed by Y Koren (1999), a RIM has all the requisites required to obtain data real-time. It primarily consists of sensors and the machine is trained to obtain information in varying time and space positions [1]. Figure 3.1 shows part of a RIM.

3.1.4 Human Inspection

It is an oldest method but still useful to obtain less critical data such as worker's logs.

Technique	Sensor
HD Imaging	HD Camera, Laser source

Process Measurement Points	Varied measurement techniques
Reconfigurable Inspection Machine	Sensors
Human inspection	Human senses

Table 3.1 Summary of data-collection techniques in manufacturing systems

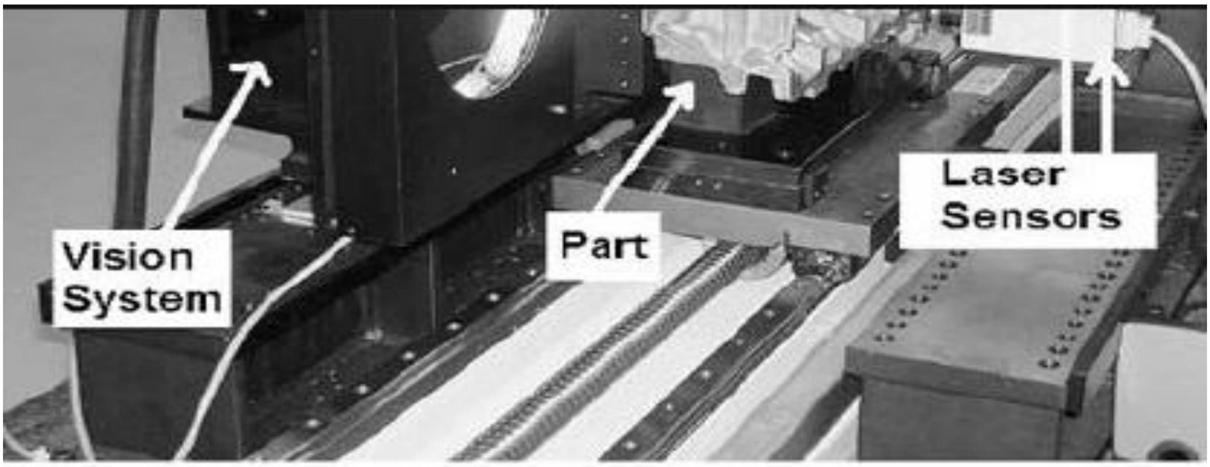

Figure 3.1: Image of an RIM

3.2 Data Types

The resulting types of data collected in manufacturing system environment are

- Images
- Audio: Acoustic emissions
- Text: Labels
- Numeric: Temperature, humidity, pressure
- String: Time-stamps, worker's log (text with numeric)
- Worker logs

Important here is to mention that computer always process data in numeric form. If an image is the source of data, computer has a way to identify the individual location and color of each pixel with an address for processing it further.

The feature based nature of dataset is such that each data can be qualified according to one or more of these,

i. Useless (noise)

ii. Nominal
iii. Binary
iv. Ordinal
v. Count
vi. Time
vii. Interval
viii. Image
ix. Video
x. Audio
xi. Text

Useless data bears no relationship with the output variable. **Nominal** data bears no relationship between different classes. **Ordinal** data can be ranked but they provide no distance between to discrete points. **Binary** data is either fail or pass, on or off. **Count** data means all the numbers are positive, the lowest possible value can be zero but not negative. Cyclic data is called **Time**, it repeats. Time data can be ignored doing classification problems, for example, hours before the previous machine break down happened. **Interval** data is continuous and useful for regression based classification techniques. **Text, Image, Audio, Video** are exceptionally helpful in doing clustering analysis (unsupervised machine learning).

3.3 Information Processing Techniques

The entire data collected through sensors might not be useful. The data-engineer needs to identify the mission-critical data and extract it in the form of features. As it happens, some of the signals carrying important data might be encoded into the least-critical data. In such a case, data-engineer is trained to extract the useful information from a lesser important signals using information processing techniques such as but not limited to the following in the process known in literature as '*data cleaning*',

- Signal Processing
- Data Augmentation
- Data Fusion

- Data Omission

In order to monitor parameters, data-engineer can also divide faults into hard-faults and soft-faults, with the mission-critical data corresponding to hard faults. Examples of hard-faults and soft-faults are given in the following table,

Criticality	Parameters
Hard Faults	Power outage, Tool breakage, Workpiece presence, Vibrations
Soft Faults	Temperature, Humidity, Surface texture, Acoustic emissions

Figure 3.2 shows how data is obtained through sensors and how it is processed.

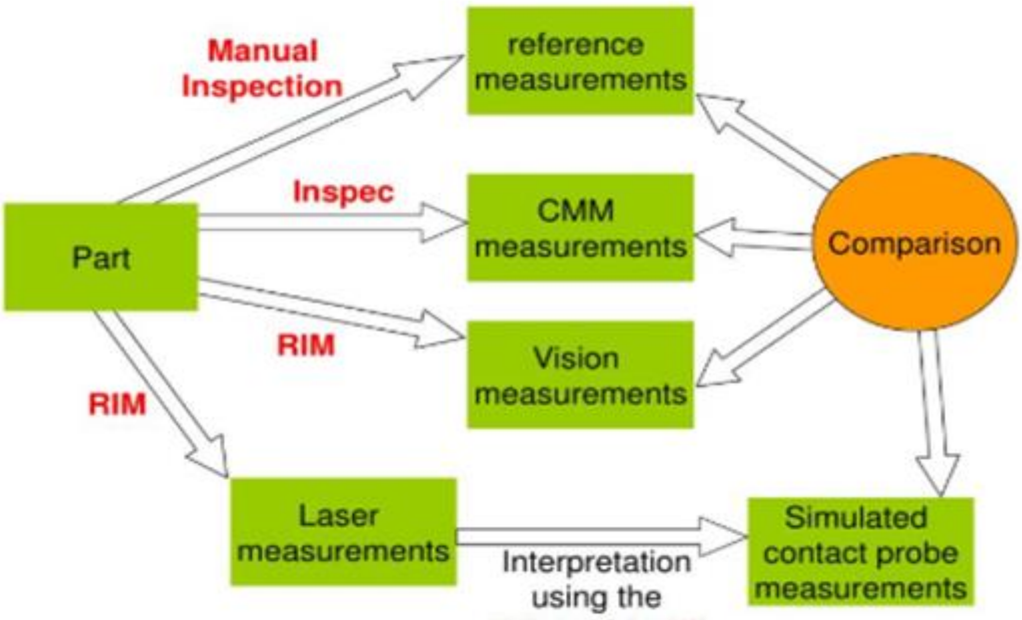

Figure 3.2: Summary of data collection methods

For example, in order to model tool life, the data acquired from shop floor it often imbalanced (all classes being unequally represented) and some points are more clustered than the rest. This needs augmenting of the input data which is achieved by mixing synthetic data to the already generated data. The synthetic data added can either be purely simulation based or the matter of an expert modeling the problem.

Due to commercial sensitivities, manufacturing organizations are reluctant to share their dataset.

3.4 Planning for the manufacturing data approach

The right of data is needed to keep productions high and not go behind manufacturing schedules because it helps do preventive maintenances both in actual manufacturing and process optimization. This can only be done if there is a system in place to collect reliable data.

There are some already existing patterns to use data.

3.4.1 Mission-critical data

Without this data, the primary goals and responsibilities of manufacturing organization can’t be met. Almost everybody in the manufacturing environment knows about the mission critical data.

3.4.2 Event-driven data

Some workforce in a manufacturing enterprise is on the look for data that can help them in the future such as researching failures in manufactured products and machine failures on the assembly line. The teams has a futuristic outlook and wants to know the event well in advance to prepare for a relevant remedy.

3.4.3 Background data

This data does not assume an emergency behind it. Temperature and humidity are the typical example of background data. This data typically gets ignored unless there is a dire need to evaluate the manufacturing problem from its angle.

Likewise, a new strategy can be devised by the manufacturing team depending on the task and information at hand.

Some product manufacturers make sure that they have done tests and data obtaining fitness programs before starting to do batch production. Tests are meant to make sure the facility is functionally performing according to the specified attributes of the product and data means the system is able to maintain a historic record of which can then be recalled to solve uncertain problems.

A manufacturing enterprise has shortlisted some risks in advance. Special emphasis can be paid around acquiring datasets from these potential risks or failures. The interval to collect data can also be minimized and data can be collected before and after the manufacture or assembly. This makes the functional performance data very flexible especially when dealing with the binary classes. Root-cause analysis can be enhanced by acquiring such a flexible data because it makes the virtual disassembly of product possible.

Structuring the data or information obtained from data becomes important if the manufacturing enterprise is concerned with a holistic maintenance of data for all aspects of the product so that it’s easy to work with later on. The complexity of data depends largely on the structure and organization of data.

Most companies spend on developing product systems rather than on data-collection techniques because it’s not an easy process to do so. Data security and managing the complexity of data, and deriving knowledge from it are complex tasks and need special expertise companies are just opening up to.

To get the value out of data, every company needs a special software with its own interface and machine learning models behind it. Presenting data in a way where engineer can understand is the most is crucial step in this process. Searching for a feature within the data demands for yet another expertise that must be embedded into the software user-interface of the company. Only a meticulous design of data strategy can generate correlations which will be of value to the manufacturing enterprise.

CHAPTER 4: THE FRAMEWORK

The framework of machine learning for diagnosability of a reconfigurable manufacturing system consists of two environments working together namely,

a) Machine Learning Environment

b) Reconfigurable Software Environment

The scheme of the framework is that as soon as the sensor data (training data) becomes available, the machine-learning environment fabricates an ensemble called the machine-learning model (algorithm) with the help of data-engineer (in the case of Supervised Machine Learning), and stores it in the memory. After the formation of machine –learning model is completed, the algorithm is trained enough to take on new data and operate the manufacturing environment on its own and deal with the uncertainties, errors, faults, inaccuracies, incursions arising with a reasonable confidence level. A Reconfigurable Manufacturing Systems when made autonomous with the incorporation of Machine Learning methods achieves a degree of autonomy and can register its own decision which are actuated through Reconfigurable Machine Tools via the control system. This scheme is outlined in **Figure 4.1**.

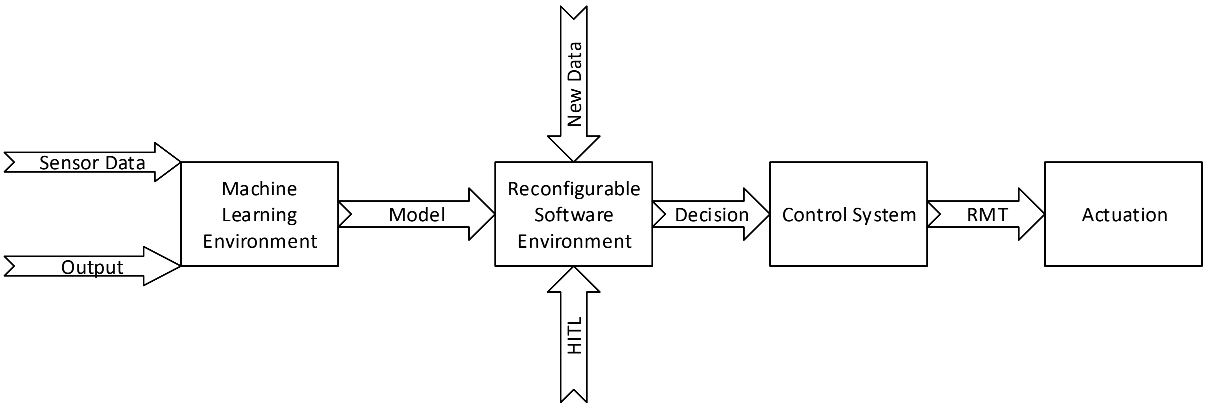

Figure 4.1: A scheme of proposed Machine Learning Framework for RMS Diagnosability

The elements, stages and phases of the Framework is discussed in detail as follows.

4.1 The Machine Learning Environment

4.1.1 Real World Data (Signals)

The Machine Learning environment consists of training the data and testing it against the new or split data. The data consists of signals. These can be obtained through Reconfigurable Inspection Machines or any set of sensors forming the monitoring part of the manufacturing system and optimally placed around the manufacturing process activity.

4.1.2 Information Processing and Signal Selection Phase

Not all signals are equally useful for monitoring a specific system. Often times a signals is a mixture of cardinal information, non-useful information and also noise. It can happen that the cardinal information is obstructed by and encoded in the latter two. Usually, there are a large number of signals which are collected, however, if every signal is called a feature, the most relevant signals should be identified. Approaches to feature selection and identification vary and are left on data-engineer’s choice. Information Processing Techniques have been described in Chapter 3. These shortlisted or selected signals can be used to factor-in the causes responsible for production errors and inaccuracies. Such an approach results in the increase in throughput, cuts the time the computer needs to learn and also decreases cost of production.

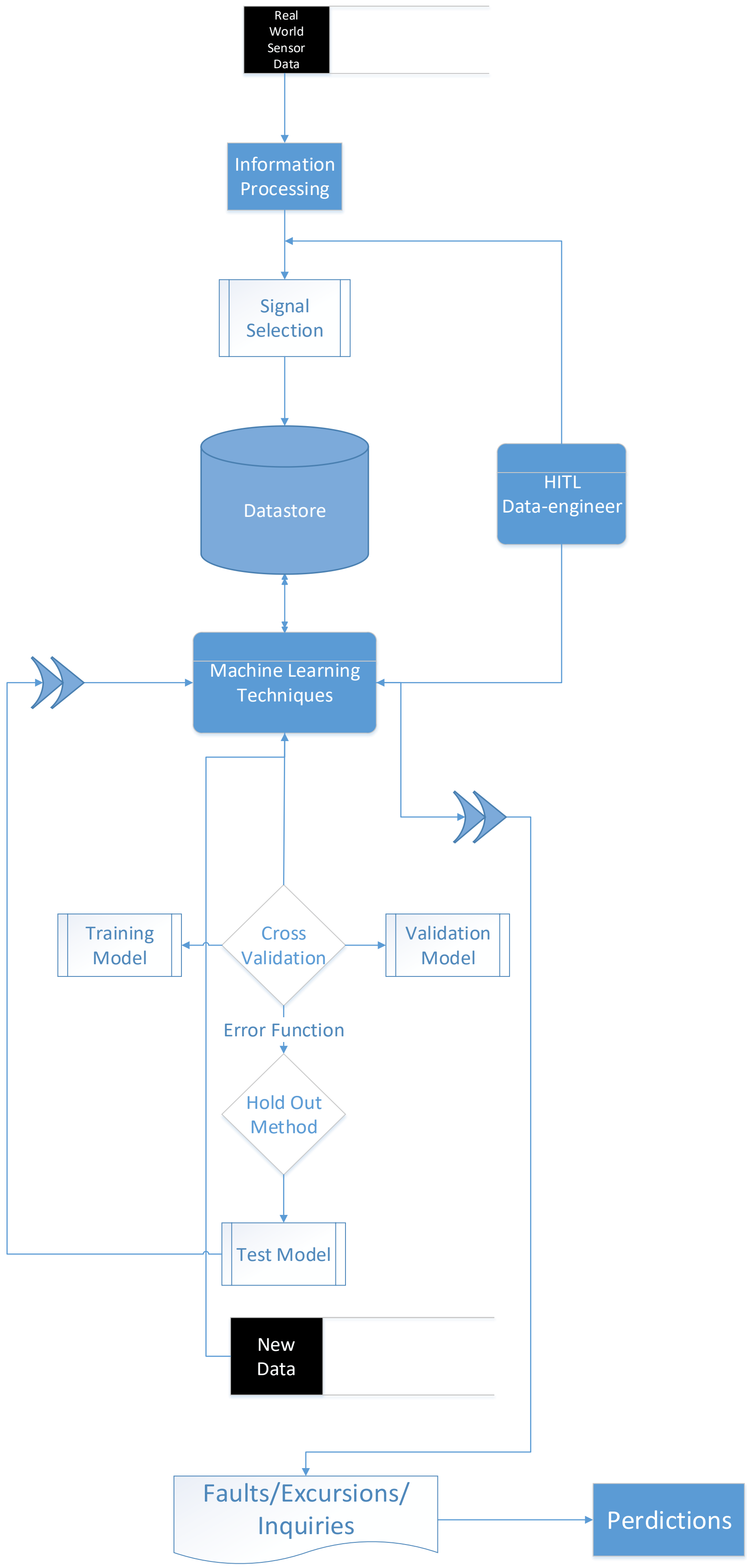

Figure 4.2 The Machine Learning Environment

4.1.3 Machine Learning Techniques

4.1.3.1 The Hold-out Method, Cross Validation

In this method, data is arbitrarily split into training and testing data. The training data is then split into training and validation data by continuous and random shuffling. The fundamental kind of cross validation is a method called the **holdout method**. A model is generated from the training data and the error-function is recorded. Using this error function a function is generated which then tried to predict the class values for test data (which serves as a new data for the training model). The error function tests itself against the new data and improves over again.

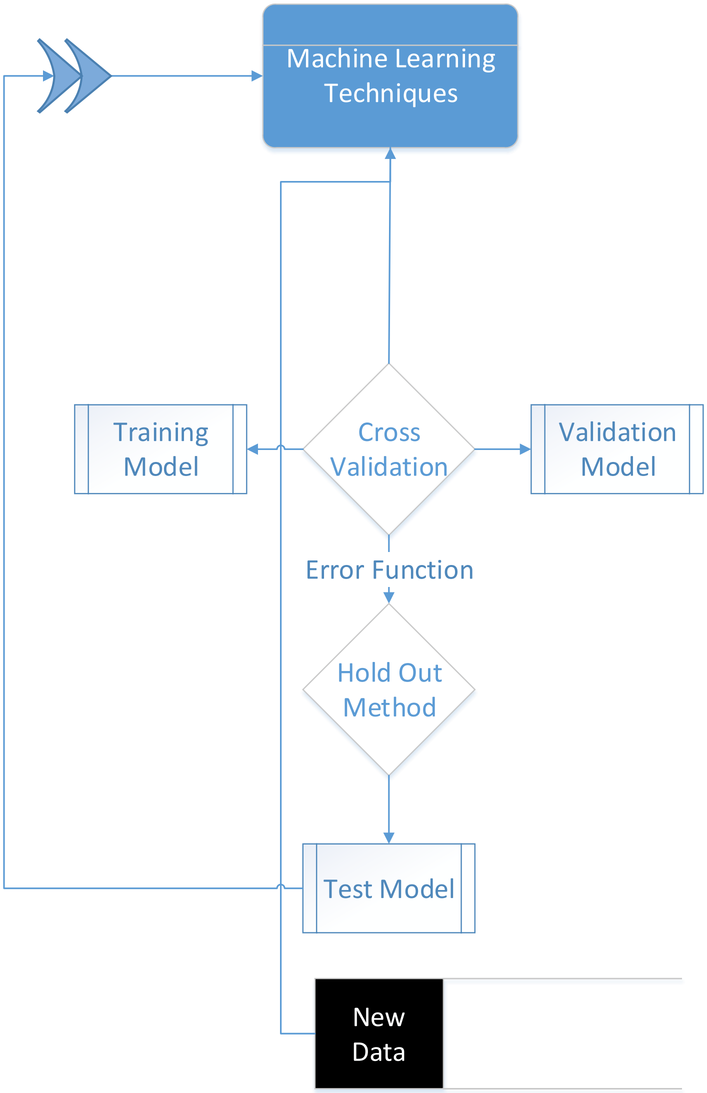

Figure 4.3 The hold-out method

The performance of algorithm is then tested on the new data. The result of new data is displayed in the form of a matrix called *confusion matrix*.

At this stage, data-engineer is able to get the answers to their ‘what if’ questions, and of predictions. An insight from the data results in the production of knowledge.

4.1.3.2 The Confusion Matrix:

The decision is represented in the form of Confusion Matrix. A detailed discussion on Confusion Matrix can be found in the section 4. .

4.2 The Reconfigurable Software Environment

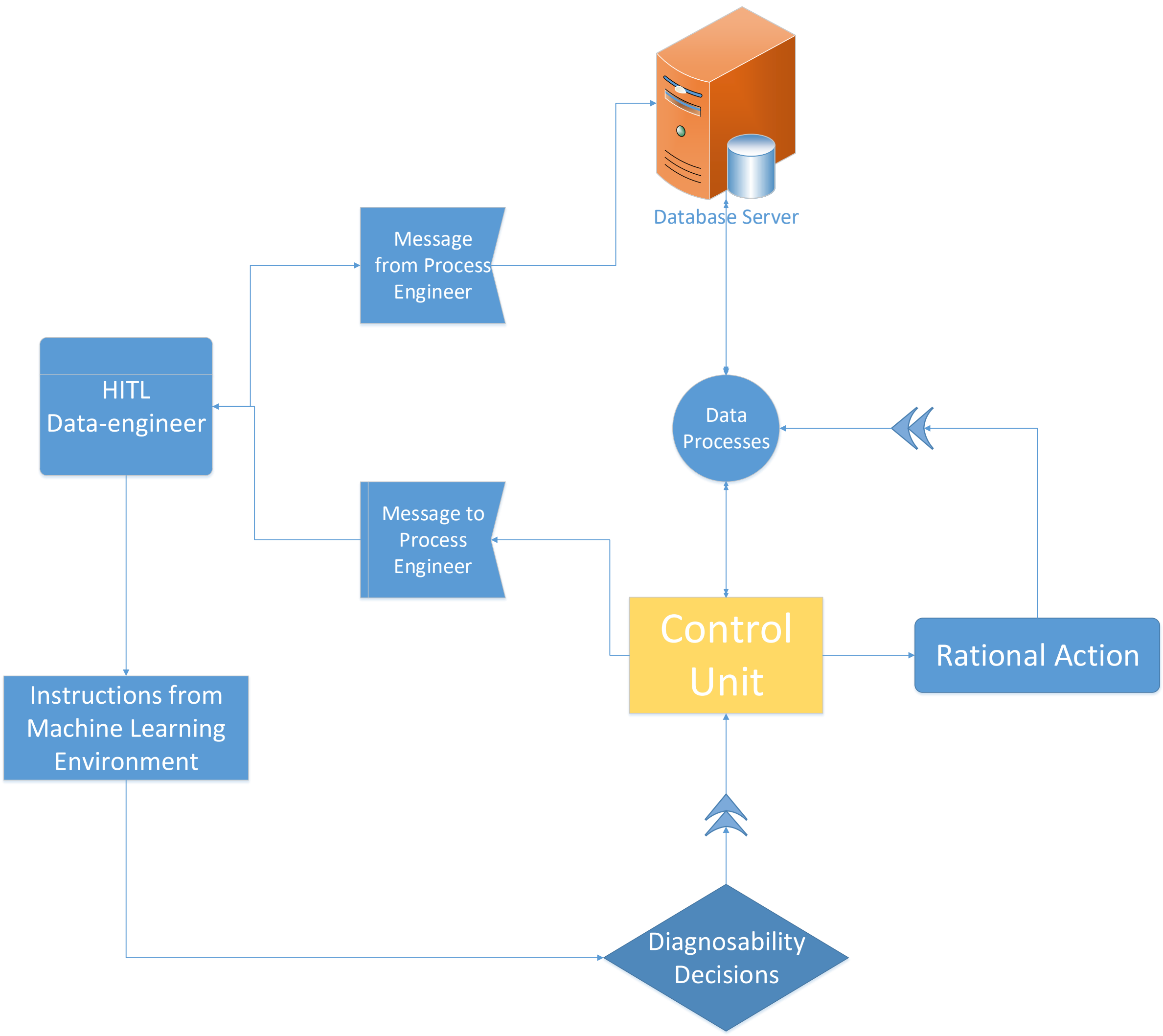

Figure 4.4 : The Reconfigurable Software Environment with Machine Learning modification for Reconfigurable Software

A Reconfigurable Software is what enables RMS perform its operation smoothly. A reconfigurable software is one with modular and open-architecture bases controllers so that the addition and assimilation of new data can be done real-time. Such a software has three components [1],

(i) an OS platform to run modules
(ii) a system to combine modules into a running system at boot-up of the control
(iii) a communication system to enable information interchange between modules on the basis of a standardized protocol

4.3 The Supervised Machine Learning Path to improving Diagnosability

4.3.1 Workflow

Reading from the framework, the workflow for supervised machine learning is as follows,

1. Preprocessing
2. HITL (data-engineer)
3. Hold out method {Cross validation [Train model. Validate model.] Test model.}
4. Using classification technique
5. Performance on new data.
6. Predictions (answers to question, situations).

4.3.2 Techniques: Logistic regression. Random tree. Random forest. Bayesian Network, so on

4.4 The Position of Data-Engineer or Process Engineer (Human-In-The-Loop)

The data-engineer is, however, only bound by their experience and, as a matter of fact, at a freedom to formulate his own model (algorithm) for the manufacturing problem at hand. Contrary to the extreme view of making manufacturing systems completely autonomous, there is going to be a need for a Human-In-The-Loop because a feedback to the newer instances can only be fed to the Reconfigurable Software after the intelligent approval of a human, for manufacturing systems are themselves dumb. There is a need for a human(s) who understands about the process going on behind the manufacturing because the manufacturing system itself can break, and as such to give inputs to the RMS at the outset of accepting a new product order.

4.5 For Unsupervised Machine Learning Path to improving Diagnosability

The workflow of Unsupervised Machine Learning becomes,

1. Preprocessing
2. Choosing Validity Index
3. Using clustering technique
4. Performance on new data.

4.6 The Feedback and Databases

Algorithm other than the Machine Learning models are at work to continuously monitor the real-time changes taking place in the software environment and register them in 'the databases' for future references. In this way learning rate in RMS takes place at exponential rates, and algorithm being constantly improved giving more choices to Data-engineer or Process-engineer over time.

4.7: The Confusion Matrix

Confusion Matrix is used to evaluate the quality of the output of a classifier. Always a square matrix. Results from supervised Machine Learning in improving the Diagnosability of RMS will hence be represented in the form of 'Confusion Matrix'.

A symbolic example of Confusion Matrix with 5 classes A, B, C, D, and E is,

	PREDICTED					
ACTUAL		A	B	C	D	E
	A	**TP_A**	E_{AB}	E_{AC}	E_{AD}	E_{AE}
	B	E_{BA}	**TP_B**	E_{BC}	E_{BD}	E_{BE}
	C	E_{CA}	E_{CB}	**TP_C**	E_{CD}	E_{CE}
	D	E_{DA}	E_{DB}	E_{DC}	**TP_D**	E_{DE}
	E	E_{EA}	E_{EB}	E_{EC}	E_{ED}	**TP_E**

Table 4.1: A symbolic representation of a 5-Classes Confusion Matrix

Columns for the Predicted vales for that class, and rows are Actual values of that class. Diagonal elements of the Confusion Matrix called True Positives or Classification because they are correctly identified. The non-diagonal entries are misclassifications or errors (E_{AB} reads A erroneously classified as B, E_{ED} as E erroneously classified as D, etc).

We are interested in the following,

The number of **True Positives** (the values in the diagonal),

True Positives, TP = $TP_A + TP_B + TP_C + TP_D + TP_E$

The number of **False Positives** (incorrectly identified),

False Positives, FP = Sum of all column elements except the TP

For Class D in the above illustration, FP = $E_{AD} + E_{BD} + E_{CD} + E_{ED}$

The number of **True Negatives** (correctly rejected),

True Negatives, TN = Sum of all columns and rows excluding that class's column and row

		PREDICTED				
		A	B	C	D	E
ACTUAL	A	**TP_A**	E_{AB}		E_{AD}	E_{AE}
	B	E_{BA}	**TP_B**		E_{BD}	E_{BE}
	C					
	D	E_{DA}	E_{DB}		**TP_D**	E_{DE}
	E	E_{EA}	E_{EB}		E_{ED}	**TP_E**

Table 4.2: The omission of row and column for calculating TN for class C.

For Class C in the above illustration, TN = **TP_A** + E_{AB} + E_{AD} + E_{AE} + E_{BA} + **TP_B** + E_{BD} + E_{BE} + E_{DA} + E_{DB} + **TP_D** + E_{DE} + E_{EA} + E_{EB} + E_{ED} + **TP_E**

The number of **False Negatives** (incorrectly rejected),

False Negatives, FN = Sum of all row entries except TP

For Class A in the above illustration, FN = $E_{AB} + E_{AC} + E_{AD} + E_{AE}$

And the total number of Test Examples of any class would be the sum of the corresponding row, that is, the sum of True Positives and False Negatives for that class.

Test Examples, TE = TP + FN

For Class E in the above illustration, TE = E_{EA} + E_{EB} + E_{EC} + E_{ED} + $\mathbf{TP_E}$

Interpreting the Confusion Matrix,

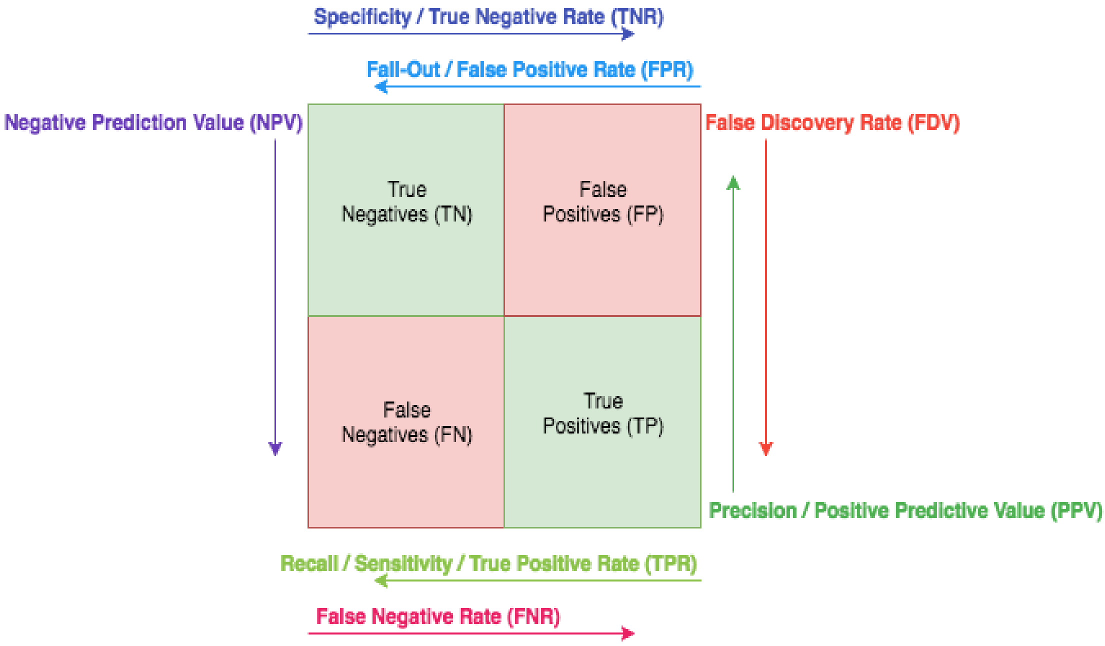

Figure 4.5 Explanation of Confusion Matrix

Calculations from Confusion Matrix are as follows,

a. Accuracy

Accuracy = Sum of correct classifications divided by the total number of classifications = TP / (Sum of all entries of Confusion Matrix)

b. Error (or Misclassification Rate)

Error = 100 – Accuracy

c. Precision (or Positive Predictive Value)

Precision = TP / (TP+FP)

For Class B in the above illustration, Precision B = $\mathbf{TP_B}$ / (E_{AB} + $\mathbf{TP_B}$ + E_{CB} + E_{DB} + E_{EB})

d. Prevalence

How often does ‘Yes’ happen in data?

Prevalence = Actual number of Yes / Total number of Yes

e. False Negative Rate

False Negative Rate = FN / (TP + FN)

f. False Positive Rate

How many times does it incorrectly predict a ‘No as a ‘Yes.’

FPR = FP / (FP + TN)

g. Sensitivity (or Recall or True Positive Rate)

How many times does it correctly predict a ‘Yes’ as a ‘Yes.’

Sensitivity = TP / (TP+FN)

For class E in the above illustration, Sensitivity E = $\mathbf{TP_E}$ / ($\mathbf{TP_E}$ + E_{EA} + E_{EB} + E_{EC} + E_{ED})

h. Specificity (or True Negative Rate)

How many times does it correctly predict a ‘No’ as a ‘No.’

Specificity = TN/(TN+FP)

For class D in the above illustration, Sensitivity D = ($\mathbf{TP_A}$ + E_{AB} + E_{AC} + E_{AE} + E_{BA} + $\mathbf{TP_B}$ + E_{BC} + E_{BE} + E_{CA} + E_{CB} + $\mathbf{TP_C}$ + E_{CE} + E_{EA} + E_{EB} + E_{EC} + $\mathbf{TP_E}$) / ($\mathbf{TP_A}$ + E_{AB} + E_{AC} + E_{AE} + E_{BA} + $\mathbf{TP_B}$ + E_{BC} + E_{BE} + E_{CA} + E_{CB} + $\mathbf{TP_C}$ + E_{CE} + E_{EA} + E_{EB} + E_{EC} + $\mathbf{TP_E}$ + $\underline{E_{AD} + E_{BD} + E_{CD} + E_{ED}}$)

The sum of FP in the above equation is underlined for Class D.

4.8 The Concurrent Engineering Perspective

Changes in the market conditions drive the modern manufacturing to decide on to produce a certain item or halt its production. The Framework has been rendered for Concurrent Engineering practices in the industry where the entire production team needs to communicate and be updated about the production process real-time. Costs of production and changes in design can then be implemented at an earlier stage saving cost and production times. This is one of the advantages of the proposed framework.

A massive capital can be saved by accelerating development and doing a better job on it increasing the quality and development before the manufacturing system gets to the production because in development the design is flexible and changes can be made. A similar scenario can be faced at the time of production as in how the system is used to prevent line down the process flow.

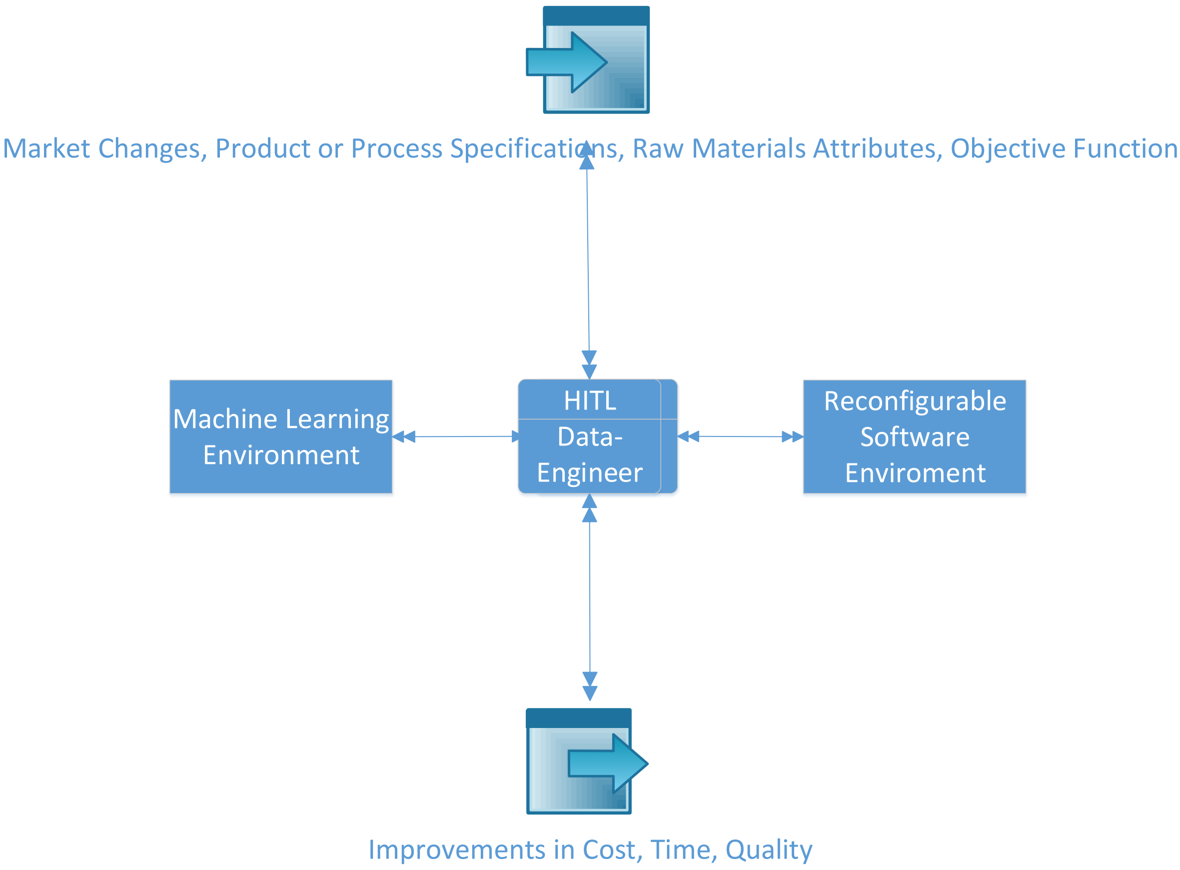

Figure 4.6: The Concurrent Engineering aspect to Reconfigurable Manufacturing Systems

4.9 The Framework

The final shape Framework assumes is as follow,

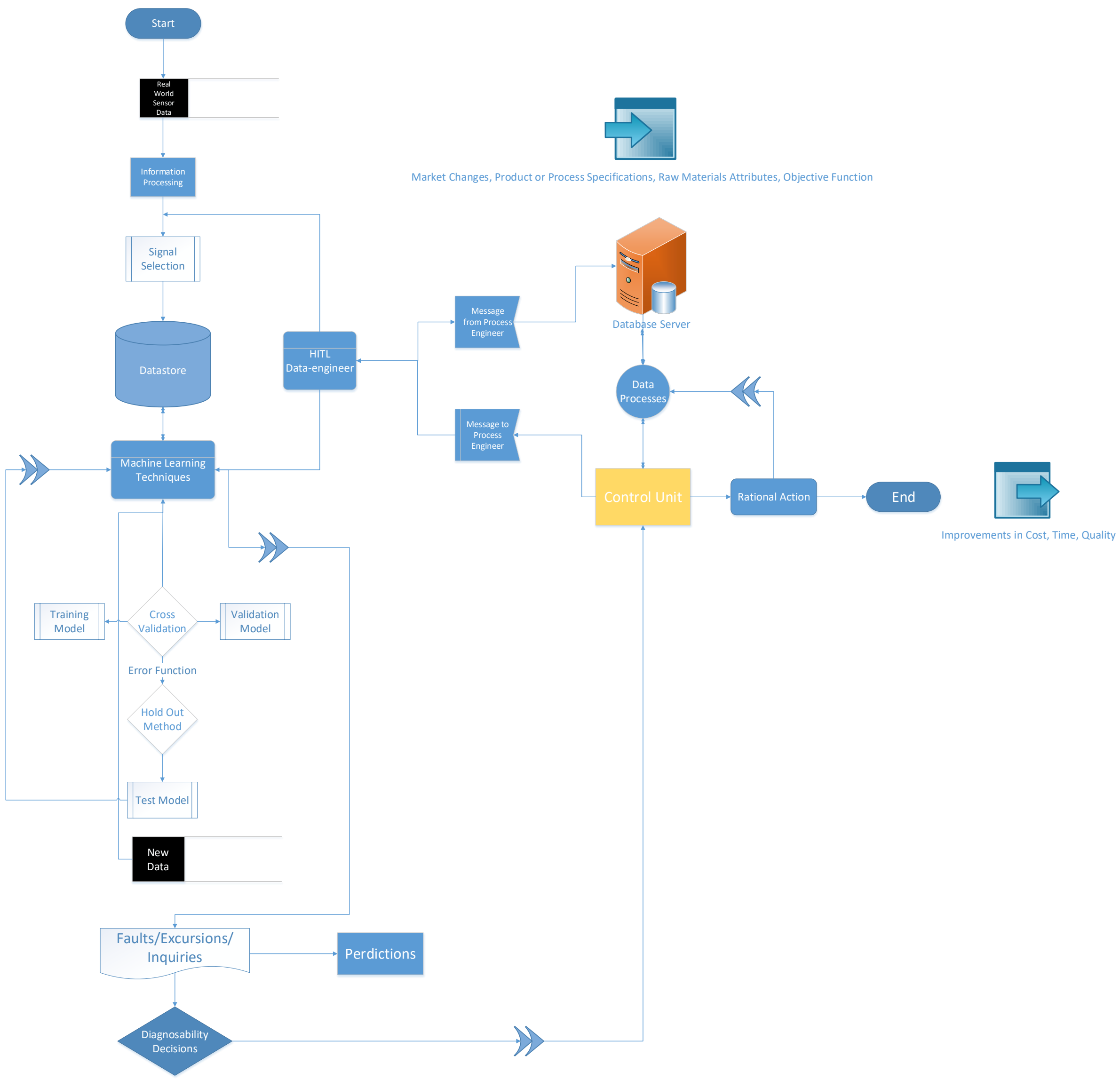

Figure 4.7: The Flowchart of Machine Learning Framework for improving Diagnosability of an RMS

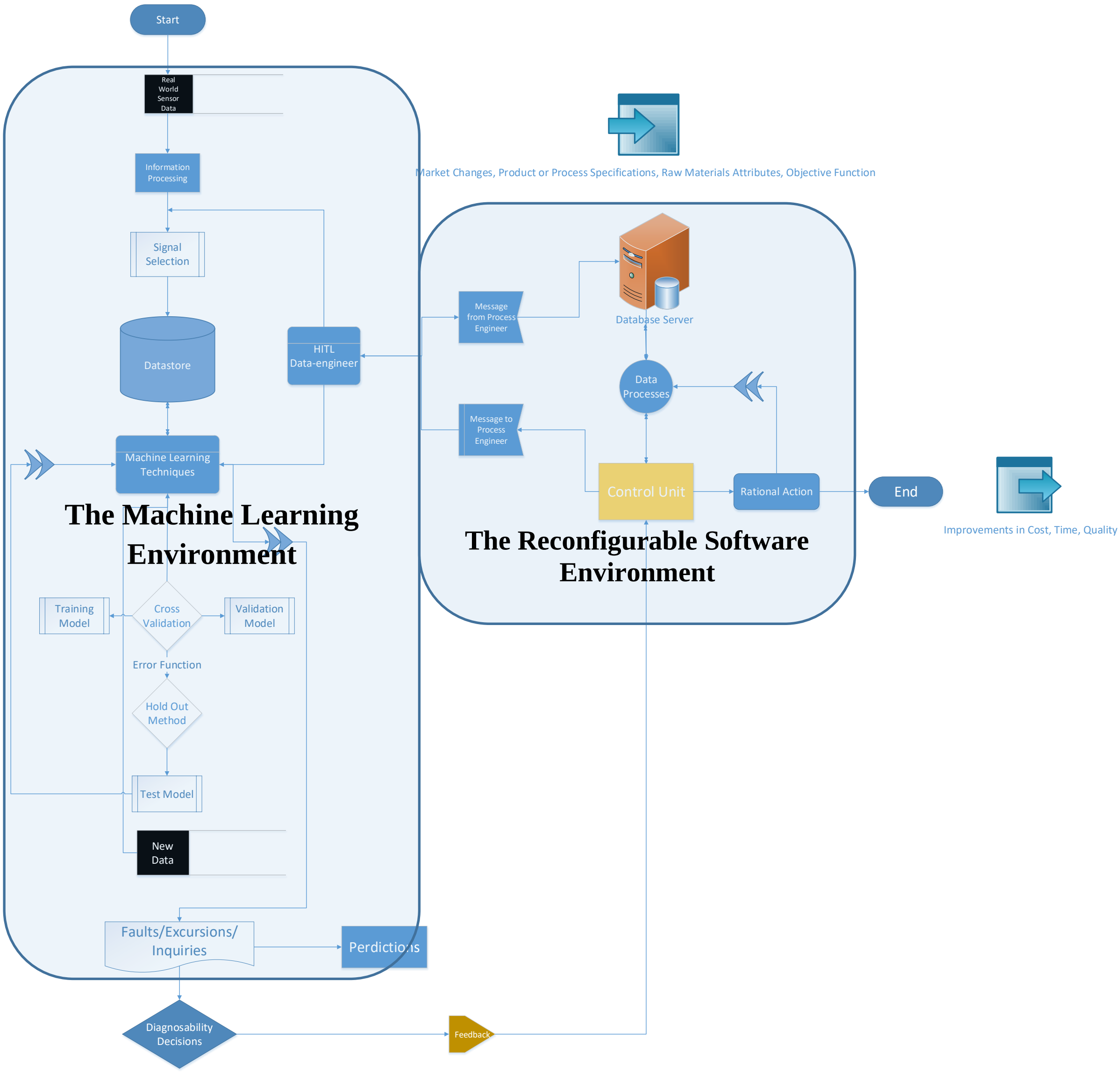

Figure 4.8 Delineation proper of Machine Learning environment, and Reconfigurable Software environment on the Framework.

4.10 Summary

The broader overview of the scheme of the proposed framework is depicted in **Figure 4.9**.

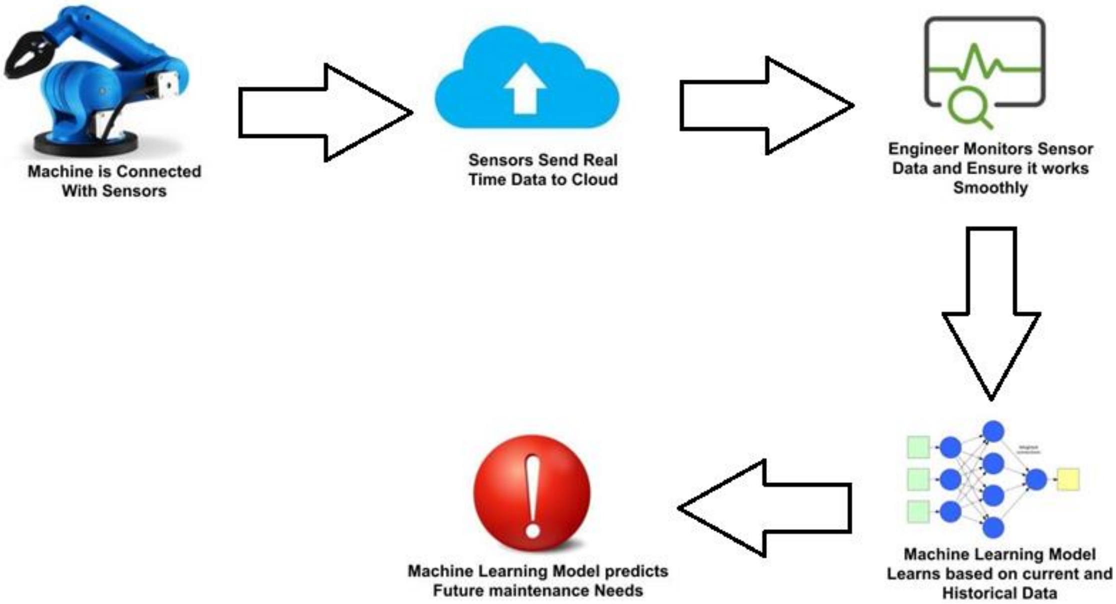

Figure 4.9 A broader understanding of the Framework

CHAPTER 5: CASE STUDIES

It was found that due to commercial sensitivities, manufacturing organizations were reluctant to share or donate their datasets. Subject to their availability, two datasets were chosen to implement the framework for the proposed methodology, the datasets were tested against five models producing the most accuracy, and results were reported in detail.

5.1 Case Study 1: Predicting Machine Failures

Machines can fail due to plethora of reasons among them can be power failure, tool breakage, temperature, humidity etc. The tested dataset for machine failures had two classes. Confusion matrix will be a 2x2 matrix.

Following are the models the dataset was tested against and at the splits mentioned. The actionable model for the case of

5.1.1 Logistic Function (Sigmoid Function)

Because Logistic Regression would work for continuous values, the model is modified to work for discrete values to get Logistic model.

If there are *k* classes for *n* instances with *m* attributes, the parameter matrix B to be calculated will be an *m*(k-1)* matrix.

The probability for class *j* with the exception of last class is

$$Pj(Xi) = \frac{e(XiBj)}{\sum(Xi * Bj) + 1}$$

The last class has probability,

$$1 - \left(\sum[j = 1 \ldots (k-1)]Pj(Xi)\right) = \frac{1}{\sum(Xi * Bj) + 1}$$

Pseudocode of the Logistic Function,

```
Repeat {

  1. Calculate gradient average
  2. Multiply by learning rate
  3. Subtract from weights

}
```

Following results are obtained from WEKA [11],

@ 60 % Split

The Confusion Matrix,

	YES	NO
YES	3482	0
NO	12	20

Accuracy = 99.6585 %

Error = 0.3415 %

Sensitivity = 0.9966

Specificity = 1.0000

@ 70 % Split

The Confusion Matrix,

	YES	NO
YES	2611	0
NO	8	16

Accuracy = 99.6964 %

Error = 0.3036 %

Sensitivity = 0.9969

Specificity = 1.0000

@ 80 % Split

The Confusion Matrix,

	YES	NO
YES	1740	0
NO	6	11

Accuracy = 99.6585 %

Error = 0.3415 %

Sensitivity = 0.9966

Specificity = 1.0000

5.1.2 Classification Via Regression

In this model, the class is binarized and one regression model is built for each class value.

Following results are obtained from Weka,

@ 60 % Split

The Confusion Matrix,

	YES	NO
YES	3482	0
NO	9	23

Accuracy = 99.7439 %

Error = 0.2561 %

Sensitivity = 0.9974

Specificity = 1.0000

@ 70 % Split

The Confusion Matrix,

	YES	NO
YES	2611	0
NO	6	18

Accuracy = 99.7723 %

Error = 0.2277 %

Sensitivity = 0.9977

Specificity = 1.0000

@ 80 % Split

The Confusion Matrix,

	YES	NO
YES	1740	0
NO	3	14

Accuracy = 99.8293 %

Error = 0.1707 %

Sensitivity = 0.9982

Specificity = 1.0000

5.1.3 Bayes Net

It is a base class for Bayesian Network classifier which is formed on algorithms like K2 and B.

@ 60 % Split

The Confusion Matrix,

	YES	NO
YES	3480	2
NO	8	24

Accuracy = 99.7154 %

Error = 0.2846 %

Sensitivity = 0.9977

Specificity = 0.9230

@ 70 % Split

The Confusion Matrix,

	YES	NO
YES	2607	4
NO	4	20

Accuracy = 99.6964 %

Error = 0.3036 %

Sensitivity = 0.9984

Specificity = 0.8333

@ 80 % Split

The Confusion Matrix,

	YES	NO
YES	1738	2
NO	2	15

Accuracy = 99.7723 %

Error = 0.2277 %

Sensitivity = 0.9988

Specificity = 0.8832

5.1.4 Random Forest

Random Forest builds many small decision-trees in parallel. All of these small decision-trees have few features which can be combined to get a strong learning model. This is a cost effective computational model.

Pseudocode of the Random Forest model,

1. Randomly select **“k”** features from total **“m”** features.
    1. Where **k << m**
2. Among the **“k”** features, calculate the node **“d”** using the best split point.
3. Split the node into **daughter nodes** using the **best split**.
4. Repeat **1 to 3** steps until “l” number of nodes has been reached.
5. Build forest by repeating steps **1 to 4** for “n” number times to create **“n” number of trees**.

@ 60 % Split

The Confusion Matrix,

	YES	NO
YES	3481	1
NO	8	24

Accuracy = 99.7439 %

Error = 0.2561 %

Sensitivity = 0.9977

Specificity = 0.9600

@ 70 % Split

The Confusion Matrix,

	YES	NO
YES	2611	0
NO	5	19

Accuracy = 99.8102 %

Error = 0.1898 %

Sensitivity = 0.9980

Specificity = 1.0000

@ 80 % Split

The Confusion Matrix,

	YES	NO
YES	1740	0
NO	2	15

Accuracy = 99.8862 %

Error = 0.1138 %

Sensitivity = 0.9988

Specificity = 1.0000

5.1.5 Comparison of the performance of the models

Graph-plots of results,

5.1.6 The Final choice of the model

Comparison of models and the final choice of model according to decision relevance is as follows

Model	Split	Accuracy	Sensitivity	Specificity	Sum
Logistic Regression	60	99.6585	0.9966	1.0000	101.6551
	70	99.6964	0.9969	1.0000	101.6933
	80	99.6585	0.9966	1.0000	101.6551
Classification via Regression	60	99.7439	0.9974	1.0000	101.7413
	70	99.7723	0.9977	1.0000	101.77
	80	99.8293	0.9982	1.0000	101.8275
Bayes Net	60	99.7154	0.9977	0.9230	101.6361
	70	99.6964	0.9984	0.8333	101.5281
	80	99.7723	0.9988	0.8832	101.6543
Random Forest	60	99.7439	0.9977	0.9600	101.7016
	70	99.8102	0.9980	1.0000	101.8082
	80	99.8862	0.9988	1.0000	101.885

Table 5.1: Comparison of models and the final choice of model

The above results shows that **Random Forest** model gives the optimum results a data-engineer can base his decisions on. Classification via Regression follows closely with Random Forest.

5.2 Case Study 2: Identifying Faults in Steel Plates

The dataset for this case has seven classes, a 7x7 matrix will result. Following are the details of classes (dependent variables),

a = Pastry

b = Z_Scratch

c = K_Scatch

d = Stains

e = Dirtiness

f = Bumps

g = Other_Faults

5.2.1 Logistic Function (Sigmoid Function)

@ 60 % Split

The Confusion Matrix,

	a	**b**	**c**	**d**	**e**	**f**	**g**
a	38	4	1	0	1	10	19
b	0	73	2	0	0	4	5
c	1	0	129	1	0	1	6
d	0	1	0	28	0	0	6
e	0	0	0	0	11	1	13
f	1	1	1	2	0	93	55
g	12	9	12	1	3	56	173

Accuracy = 70.6186 %

Error = 29.3814 %

	Sensitivity	Specificity
a = Pastry	0.521	0.731
b = Z_Scratch	0.869	0.830
c = K_Scatch	0.935	0.884
d = Stains	0.800	0.933
e = Dirtiness	0.440	0.733
f = Bumps	0.612	0.564
g = Other_Faults	0.654	0.629
Weighted Avg	**0.706**	**0.710**

@ 70 % Split

The Confusion Matrix,

	a	b	c	d	e	f	g
a	32	4	0	0	0	8	15
b	0	56	2	0	0	5	4
c	1	0	100	1	0	1	3
d	0	1	0	19	0	0	4
e	0	0	0	0	6	1	11
f	1	1	1	2	0	73	41
g	3	8	6	3	2	35	132

Accuracy = 71.8213 %

Error = 28.1787 %

	Sensitivity	Specificity
a = Pastry	0.542	0.865
b = Z_Scratch	0.836	0.800
c = K_Scatch	0.943	0.917
d = Stains	0.792	0.760
e = Dirtiness	0.333	0.750
f = Bumps	0.613	0.593

g = Other_Faults	0.698	0.629
Weighted Avg	**0.718**	**0.727**

@ 80 % Split

The Confusion Matrix,

	a	**b**	**c**	**d**	**e**	**f**	**g**
a	25	2	0	0	1	4	5
b	0	44	1	0	0	2	4
c	1	0	71	0	0	0	1
d	0	1	0	14	0	0	1
e	1	0	0	0	3	0	7
f	1	2	0	1	0	48	27
g	2	2	4	2	2	26	83

Accuracy = 74.2268 %

Error = 25.7732 %

	Sensitivity	**Specificity**
a = Pastry	0.676	0.833
b = Z_Scratch	0.863	0.863
c = K_Scatch	0.973	0.934
d = Stains	0.875	0.824
e = Dirtiness	0.273	0.500
f = Bumps	0.608	0.600
g = Other_Faults	0.686	0.648
Weighted Avg	**0.742**	**0.741**

5.2.2 Classification Via Regression

@ 60 % Split

The Confusion Matrix,

	a	b	c	d	e	f	g
a	36	7	0	0	1	9	20
b	0	81	2	0	0	0	1
c	0	0	132	1	1	2	2
d	0	0	0	33	0	1	1
e	0	0	0	0	17	4	4
f	1	4	1	1	0	95	50
g	7	11	6	0	3	69	173

Accuracy = 73.067 %

Error = 26.933 %

	Sensitivity	Specificity
a = Pastry	0.493	0.818
b = Z_Scratch	0.964	0.786
c = K_Scatch	0.957	0.936
d = Stains	0.943	0.943
e = Dirtiness	0.680	0.773
f = Bumps	0.625	0.528
g = Other_Faults	0.643	0.689
Weighted Avg	**0.731**	**0.738**

@ 70 % Split

The Confusion Matrix,

	a	b	c	d	e	f	g
a	25	7	0	0	1	5	21
b	0	63	2	0	0	0	2
c	0	0	101	0	0	2	3
d	0	0	0	23	0	0	1
e	0	0	0	0	11	0	7
f	2	1	1	1	0	75	39
g	3	9	2	0	3	40	132

Accuracy = 73.8832 %

Error = 26.1168 %

	Sensitivity	Specificity
a = Pastry	0.424	0.833
b = Z_Scratch	0.940	0.788
c = K_Scatch	0.953	0.953
d = Stains	0.958	0.958
e = Dirtiness	0.611	0.733
f = Bumps	0.630	0.615
g = Other_Faults	0.698	0.644
Weighted Avg	**0.739**	**0.746**

@ 80 % Split

The Confusion Matrix,

	a	b	c	d	e	f	g
a	21	2	0	0	0	2	12
b	0	47	1	0	0	1	2
c	0	0	70	0	0	1	2
d	0	0	0	15	0	0	1
e	0	0	0	0	6	1	4
f	1	0	0	1	0	59	18
g	1	5	0	0	0	29	86

Accuracy = 78.3505 %

Error = 21.6495 %

	Sensitivity	Specificity
a = Pastry	0.568	0.913
b = Z_Scratch	0.922	0.870
c = K_Scatch	0.959	0.986
d = Stains	0.938	0.938
e = Dirtiness	0.545	1.000

f = Bumps	0.747	0.634
g = Other_Faults	0.711	0.688
Weighted Avg	**0.784**	**0.798**

5.2.3 Bayes Net

@ 60 % Split

The Confusion Matrix,

	a	**b**	**c**	**d**	**e**	**f**	**g**
a	43	7	0	0	6	11	6
b	1	71	0	0	0	7	5
c	0	0	125	1	0	1	11
d	0	0	0	33	0	1	1
e	3	0	0	0	17	2	3
f	10	1	0	1	0	102	38
g	29	8	4	5	1	72	150

Accuracy = 69.7165 %

Error = 30.2835 %

	Sensitivity	**Specificity**
a = Pastry	0.589	0.500
b = Z_Scratch	0.845	0.816
c = K_Scatch	0.906	0.969
d = Stains	0.943	0.825
e = Dirtiness	0.680	0.708
f = Bumps	0.671	0.520
g = Other_Faults	0.558	0.701
Weighted Avg	**0.697**	**0.713**

@ 70 % Split

The Confusion Matrix,

	a	b	c	d	e	f	g
a	36	7	0	0	1	10	5
b	1	58	0	0	0	4	4
c	0	0	96	0	0	1	9
d	0	0	0	22	0	0	2
e	2	0	0	0	11	2	3
f	3	0	0	1	1	86	28
g	18	7	3	2	0	45	114

Accuracy = 72.6804 %

Error = 27.3196 %

	Sensitivity	Specificity
a = Pastry	0.610	0.600
b = Z_Scratch	0.866	0.806
c = K_Scatch	0.906	0.970
d = Stains	0.917	0.880
e = Dirtiness	0.611	0.846
f = Bumps	0.723	0.581
g = Other_Faults	0.603	0.691
Weighted Avg	**0.727**	**0.736**

@ 80 % Split

The Confusion Matrix,

	a	b	c	d	e	f	g
a	25	3	0	0	1	4	4
b	0	44	0	0	0	3	4
c	0	0	66	0	0	0	7
d	0	0	0	15	0	0	1
e	0	0	0	0	7	0	4
f	3	0	0	1	0	62	13
g	15	1	2	1	0	32	70

Accuracy = 74.4845 %

Error = 25.5155 %

	Sensitivity	Specificity
a = Pastry	0.676	0.581
b = Z_Scratch	0.863	0.917
c = K_Scatch	0.904	0.971
d = Stains	0.938	0.882
e = Dirtiness	0.636	0.875
f = Bumps	0.785	0.614
g = Other_Faults	0.579	0.680
Weighted avg	**0.745**	**0.757**

5.2.4 Random Forest

@ 60 % Split

The Confusion Matrix,

	a	b	c	d	e	f	g
a	32	4	0	0	0	9	28
b	0	74	1	0	0	1	8
c	0	0	135	0	0	0	3
d	0	0	0	32	0	1	2
e	0	0	0	0	21	2	2
f	1	1	0	1	0	107	42
g	4	9	4	3	0	48	201

Accuracy = 77.5773 %

Error = 22.4227 %

	Sensitivity	Specificity
a = Pastry	0.438	0.865
b = Z_Scratch	0.881	0.841

c = K_Scatch	0.978	0.964
d = Stains	0.914	0.889
e = Dirtiness	0.840	1.000
f = Bumps	0.704	0.637
g = Other_Faults	0.747	0.703
Weighted Avg	**0.776**	**0.785**

@ 70 % Split

The Confusion Matrix,

	a	**b**	**c**	**d**	**e**	**f**	**g**
a	26	2	0	0	0	8	23
b	0	58	0	0	0	1	8
c	0	0	103	0	0	0	3
d	0	0	0	22	0	0	2
e	0	0	0	0	15	1	2
f	2	0	0	1	0	91	25
g	4	6	4	2	0	31	142

Accuracy = 78.5223 %

Error = 21.4777 %

	Sensitivity	**Specificity**
a = Pastry	0.441	0.813
b = Z_Scratch	0.866	0.879
c = K_Scatch	0.972	0.963
d = Stains	0.917	0.880
e = Dirtiness	0.833	1.000
f = Bumps	0.765	0.689
g = Other_Faults	0.751	0.693
Weighted Avg	**0.785**	**0.792**

@ 80 % Split

The Confusion Matrix,

	a	b	c	d	e	f	g
a	23	0	0	0	0	1	13
b	0	45	1	0	0	1	4
c	0	0	70	0	0	0	3
d	0	0	0	14	0	0	2
e	0	0	0	0	11	0	0
f	1	0	0	0	0	59	19
g	1	2	3	0	0	20	95

Accuracy = 81.701 %

Error = 18.299 %

	Sensitivity	Specificity
a = Pastry	0.622	0.920
b = Z_Scratch	0.882	0.957
c = K_Scatch	0.959	0.946
d = Stains	0.875	1.000
e = Dirtiness	1.000	1.000
f = Bumps	0.747	0.728
g = Other_Faults	0.785	0.699
Weighted Avg	**0.817**	**0.827**

5.2.5 Comparison of the performance of the models

Graph-plots of results,

5.2.6 The Final choice of the model

Comparison of models and the final choice of model according to decision relevance is as follows

Model	Split	Accuracy	Sensitivity	Specificity	Sum
	60	70.6186	0.706	0.710	72.0346

Logistic Regression	70	71.8213	0.718	0.727	73.2663
	80	74.2268	0.742	0.741	75.7098
Classification via Regression	60	73.067	0.731	0.738	74.536
	70	73.8832	0.739	0.746	75.3682
	80	78.3505	0.784	0.798	79.9325
Bayes Net	60	69.7165	0.697	0.713	71.1265
	70	72.6804	0.727	0.736	74.1434
	80	74.4845	0.745	0.757	75.9865
Random Forest	60	77.5773	0.776	0.785	79.1383
	70	78.5223	0.785	0.792	80.0993
	80	81.701	0.817	0.827	83.345

Table 5.2 Comparison of models and the final choice of model

The above results shows that **Random Forest** model gives the optimum results a data-engineer can base his decisions on. Classification via Regression follows closely with Random Forest.

CHAPTER 6: CONCLUSION

6.1 Advantages of research

1. The solution of NP-complete problems is possible through Machine Learning approaches.
2. Diagnosability has been rendered a three leg system.
3. Error prediction which makes the predictive maintenance of manufacturing systems possible.
4. Adds the Concurrent Engineering aspect to Reconfigurable Manufacturing System.
5. A better rested workforce.

The hallmark of research carried out in this thesis is the introduction of multi-data decision making through a machine learning algorithm that constantly improves itself as it learns more and more from its production experience. The research carried out in this introductory in the field of AI and Diagnosability, and as such can be carried forward along many lines. It is hoped that the technique presented in this research will prove to be an important step in the globalization of manufacturing practices.

Defines the position and responsibilities of a data-scientist/engineer in a Reconfigurable Manufacturing System.

6.2 Future work

By obtaining suitable datasets, and according to the strategies as delineated in the framework, the results for following cases should be demonstrated as part of the future work on the said framework.

i) Regression-based Supervised Learning

ii) Deep Learning

Should the datasets not become available, simulation based manufacturing software be used to generate a random but realistic datasets to validate the study of Framework. Following areas have a promising research potential,

1. Under the RMS lifecycle-considerations

2. Optimizing Machine Learning for RMT Control System

6.3 Inter-departmental Collaboration

A setting up of a Machine Learning laboratory in the Department of Mechanical Engineering of College of Electrical & Mechanical Engineering is suggested along with the collaboration with concerned departments to generate more research interests in the promising area of Machine Learning as applied to the latest manufacturing systems. The flow of funds is invariably related to the setting up, and working for such a laboratory.

APPENDIX-I

Python actionable model for Machine Failure dataset

```
import LogisticRegression

# To predict probabilities fill the desired input_data
# in next line. Numeric fields are compulsory if the model was not
# trained with missing numerics.
input_data = {
    "Measure14": 1,
    "Measure13": 1,
    "Hours Since Previous Failure": 1,
    "Measure15": 1,
    "Measure12": 1,
    "Measure10": 1,
    "Measure11": 1,
    "Measure8": 1,
    "Measure9": 1,
    "Measure7": 1,
    "Date.hour": 1,
    "Date.day-of-week": 1,
    "Date.day-of-month": 1,
    "Date.month": 1,
    "Measure3": 0,
    "Measure4": 1,
    "Measure1": 1,
    "Measure2": 2,
    "Humidity": 1,
    "Operator": Operator2,
    "Date": 1,
    "Temperature": 1,
    "Measure5": 1,
    "Measure6": 1
}
logisticregression.predict(input_data, full=True)

#
# input_data: dict for the input values
# (e.g. {"petal length": 1, "sepal length": 3})
# full: if set to True, the output will be a dictionary that includes the
# distribution of each class in the objective field, the predicted class
and
```

Python actionable model for Steel Plate Faults dataset

```
import LogisticRegression

# Downloads and generates a local version of the logistic regression,
# if it hasn't been downloaded previously.


# To predict probabilities fill the desired input_data
# in next line. Numeric fields are compulsory if the model was not
# trained with missing numerics.
input_data = {
    "LogOfAreas": 1,
    "Outside_Global_Index": 1,
    "Log_Y_Index": 1,
```

```
    "Log_X_Index": 1,
    "Outside_X_Index": 1,
    "Square_Index": 1,
    "Edges_Y_Index": 1,
    "Edges_X_Index": 1,
    "Luminosity_Index": 1,
    "Orientation_Index": 1,
    "Empty_Index": 1,
    "Steel_Plate_Thickness": 1,
    "Edges_Index": 1,
    "TypeOfSteel_A300": False,
    "TypeOfSteel_A400": True,
    "Length_of_Conveyer": 1,
    "Y_Perimeter": 1,
    "Sum_of_Luminosity": 1,
    "Pixels_Areas": 1,
    "X_Perimeter": 1,
    "Y_Minimum": 1,
    "Y_Maximum": 1,
    "X_Minimum": 1,
    "X_Maximum": 1,
    "Minimum_of_Luminosity": 1,
    "Maximum_of_Luminosity": 1,
    "SigmoidOfAreas": 1
}
logisticregression.predict(input_data, full=True)

#
# input_data: dict for the input values
# (e.g. {"petal length": 1, "sepal length": 3})
# full: if set to True, the output will be a dictionary that includes the
# distribution of each class in the objective field, the predicted class
and
# its probability.
```

REFERENCES

[1] Koren Y, Heisel U, Jovane F, Moriwaki T, Pritschow G, Ulsoy G, Van Brussel H. Reconfigurable manufacturing systems. CIRP Annals - Manufacturing Technology 1999.

[2] Uedal K, Itsuo H, Nobutada F, Jari V. Reinforcement Learning Approaches to Biological Manufacturing Systems - CIRP Annals - Manufacturing Technology 2000.

[3] Ding Y, Jianjun S, Dariusz C. Diagnosability Analysis of Multi-Station Manufacturing Processes. Journal of Dynamic Systems, Measurement, and Control. DOI: 10.1115/1.1435645

[4] Monostori, L. AI and machine learning techniques for managing complexity,changes and uncertainties in manufacturing. Engineering Applications of Artificial Intelligence 16 (2003) 277–291.

[5] J. P. Liu , Z. B. Luo , L. K. Chu & Y. L. Chen. Manufacturing system design with optimal diagnosability. International Journal of Production Research. DOI: 10.1080/00207540310001645147.

[6] Wuest T, Christopher I ,Klaus-Dieter T. An approach to monitoring quality in manufacturing using supervised machine learning on product state data. Journal of Intelligent Manufacturing DOI 10.1007/s10845-013-0761-y

[7] Irani, K B, Jie Cheng, Usama M. Fayyad, Zhaogang Chiang. Applying machine learning to semiconductor manufacturing. 0885/9000/93/0200-0041 IEEE 1993.

[8] Pham, D. T, A A Afify. Machine-learning techniques and their applications in manufacturing. Journal of Eningeering Manufacture. DOI: 10.1243/095440505X32274

[9] Wuest, T, Daniel W, Christopher I, Klaus-Dieter T. Machine learning in manufacturing: advantages, challenges, and applications. Production & Manufacturing research: an open access Journal, 2016 Vol. 4, no. 1, 23–45. http://dx.doi.org/10.1080/21693277.2016.1192517

[10] https://newsroom.cisco.com/press-release-content?articleId=1847422, 2019.

[11] Weka tool: Available from: http://www.cs.waikato.ac.nz/ml/weka/downloading.html, 2019.