An Internship Report on

ME4276 Industry Internship / Capstone Project

With

Trisim Technologies Pvt. Ltd., Pune

Submitted in partial fulfillment of requirements for

Bachelor of Technology

In

Mechanical Engineering

Of

Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology, Pune - 37

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

2020-21

Submitted By

Prathmesh Lonkar

ME-B-41

1710771

Institute guide

Prof. Santosh Joshi

Industry guide

Mr. Indrajit Jagtap

Trisim Technologies Pvt. Ltd.

Table of Contents

Table of Figures

Abstract

This project report describes in detail, the work undertaken by me at Trisim Technologies, Pune. I mainly worked on 3 projects during my internship tenure.

The first project, Study of different algorithms for Supervised Machine Learning problems. A research-oriented project which required me to study these various algorithms, their parameters, particular applications, advantages, disadvantages, etc.

Second project, Development of an UI based Data Science and Machine Learning Toolkit application. A project which automates and eases the process of ML model building and testing for Data Science applications.

Third project, Study of Motor Current Signature Analysis (MCSA). Again, a research-oriented project to study the concepts and application of MCSA which is useful in the condition monitoring of Induction motors.

Keywords: Supervised ML, algorithms, Data science, UI, Toolkit, API, Flask, Postman, MCSA, condition monitoring.

Work Description

During my internship, I was assigned with 3 main projects. The problem statement, objective, execution, purpose and conclusion of these projects are provided below.

Project 1

Problem Statement

To study the important algorithms that are used in solving Supervised Machine Learning problems and learn their implementation in python. Also, to prepare a detailed document of this research.

Objectives met

- To learn what supervised Machine Learning is.
- Identifying the main algorithms used for it.
- Study and compare these algorithms.
- Particular applications of these algorithms.
- Learning their implementation in Python.
- Prepare a sample code to illustrate the implementation of these algorithms.
- Prepare a documentation stating the entire study and learnings.

Project Execution

To tackle this project, first I had to learn about the very basics of Machine Learning. Machine learning is nothing but a method of data analysis that automates analytical model building. It is a branch of artificial intelligence (AI) based on the idea that systems (or say, machines) can learn from data, identify patterns and make decisions with minimal human intervention. I referred to online platforms like Kaggle to support the learning.

I learned about the types of Machine Learning problems, their classifications, terminologies related to them, etc. My concerned type was Supervised ML. In Supervised Machine Learning, models are trained using labelled dataset, where the model learns about each type of data. Once the training process is completed, the model is tested on the basis of test data (a subset of the training set), and then it predicts the output. Supervised ML is again of 2 types: Regression and Classification.

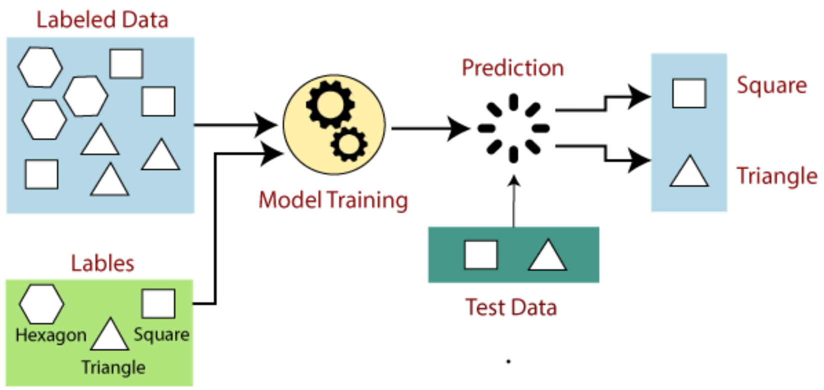

Fig1. Supervised Machine Learning

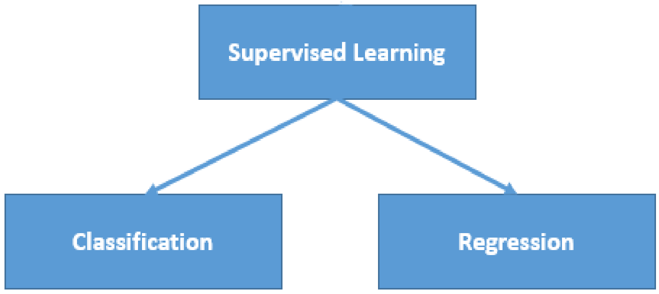

Fig2. Types of supervised ML

The main difference between Regression and Classification algorithms is that Regression algorithms are used to predict the continuous values such as price, salary, age, number of calories, etc. and Classification algorithms are used to predict/Classify the discrete values such as Male or Female, True or False, Spam or Not Spam, 1 or 0, etc.

Meanwhile, I also got my hands on implementing this Supervised ML model building process (for both, regression and classification) in Python. This required me to learn more about the data science side of modelling. Like, how to import datasets (.csv) files in python, how to perform the numerous EDA (Exploratory Data Analysis) operations on those files, how to create interactive visualizations from datasets, how to deal with different datatypes in those datasets, and many more. To do this, I studied more about python libraries like Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, Plotly, etc.

Simultaneously, I also worked on preparing the formal documentation of this study. This documentation contains information about the types of Supervised ML problems (regression and classification), detailed information of main algorithms used for both regression and classification, their explanation, parameters, comparisons, pros-cons, preferred applications, etc. This document upon completion was successfully submitted serving the project completion.

Project Purpose

The submitted project documentation will help Trisim Technologies to identify or learn about the required algorithms. Also, it will be convenient to understand which algorithm to use where depending upon the factors mentioned in the document.

The works of this project will act as a go-to guide for the company while understanding or dealing with Supervised Machine Learning problems.

Conclusion

Learned about the Machine Learning implementation in python. Undertook various online courses to meet the learning requirements.

The detailed study of different algorithms used for classification and regression (both of which are the types of supervised machine learning) was accomplished, maintained and successfully submitted in form of a descriptive document.

Project 2

Problem Statement

To develop a User Interface (UI) based Data Science and Machine Learning Toolkit application. The toolkit should perform the appropriate Machine Learning procedures based on the actions taken by the user.

Objectives met

- To learn the steps and procedure of a general ML process.
- Understanding the concepts of application development.
- Learning to create interactive UIs using a python library.
- Implementing the created UI for toolkit and verifying the code.
- Finish developing a desktop version for the toolkit application.
- Studying about APIs in order to create backend code for the web version.
- Designing the visual layout and workflow for frontend UI of the web version.
- Learning Python-Flask in order to implement API functions.
- Learning Postman in order to verify these APIs.
- Finish developing the frontend design and backend API code for web version of the toolkit application.
- Verifying the workings of the toolkit.

Project Execution

This was the central task or the focal project that I worked on. To begin with this project, I was already well equipped with the knowledge of Data-Science and Machine Learning processes, its implementation in Python, and the necessary requirements of the Machine Learning workflow (as I had grasped all these concepts while working on Project 1).

The further task required me to build a toolkit which would act as a ready UI to perform all the major Data-science and Machine Learning operations, like, viewing the datasets statistics, its information, performing EDA, data modification and cleaning operations, visualization using different charts (say, line, bar, scatter, heatmap, box, etc.), model training, tuning, validation, evaluation, prediction, etc.

The idea behind this project was to accompany an industry client (say a user who may not know programming or details of ML implementation using a program) to perform Supervised ML process in a fast, supported, and easy manner. The user of this toolkit is able to perform all the operations (right from browsing & importing the dataset to making predictions on the test set) just by clicking the buttons on the toolkit UI. In-short, the user is able to perform all the main required operations of a ML model-building process, without having much knowledge about programming (because all he/she will have to do is to click on the buttons in that interactive UI).

To progress with the development, firstly I decided to start developing a local desktop version of the toolkit. To give a slim idea to the reader, below is a cropped screenshot figure representing only one of the numerous features from the old version of the desktop toolkit.

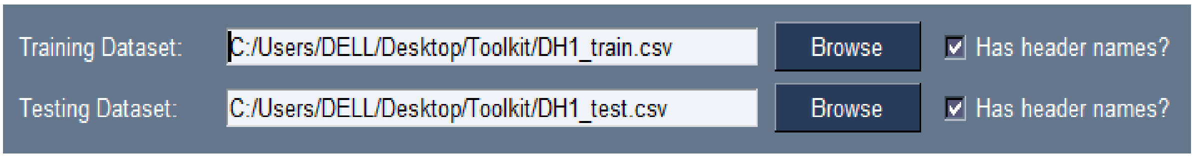

Fig3. One feature from the old desktop version

For this, I learned a new python library called PySimpleGUI. It is a Python package that enables Python programmers to create interactive GUIs. As I learned more about this library and its implementation, I started developing the desktop frontend UI. I would cross-check the UI functioning along with the backend code to verify the workings.

After completing the development of desktop version, I tested all its features on many datasets to verify proper functioning. As a result, I had finished the development of the local (or offline) desktop version of the UI based Machine Learning toolkit application.

Now, the next task of this project was to work on the Web version of the toolkit. Though my primary responsibility of this task was to develop the back-end part for this web version of toolkit, I understood that deciding how the front-end should appear was an important factor (because I had earlier worked for the front-end UI of the desktop version and was aware about the basic requirements and workflow needed). Hence for this part, I worked closely with my Industry guide to decide the modifications in the workflow for this web version of toolkit.

For this, I made the prototypes for UI designs and workflow. We screened the designs many times (during our daily online meetings). Naturally, I went through various iterations and modifications for the workflow designs. Upon reaching an acceptable UI workflow from end-user point of view, it was finalized. It should be noted that, for the front-end part of this web version, I was only involved in the designing of its visual layout and workflow i.e., how it should look, how the work-flow should happen, etc. and not in actual programming for the front-end.

After we finalized the design and workflow, I submitted my design layouts, and it was now my responsibility to start developing the back-end part for this toolkit. The front-end programming as per the design layouts was then further taken care by another employee at Trisim.

For the backend part, I was required to develop the APIs for the Toolkit. For this matter, I learned "Python-Flask". Flask is a powerful micro web framework for python. It is widely used for developing web applications. It took me a few days to completely get a grip on flask.

To verify the proper functioning of these APIs, I used another application called, "Postman". Postman is a popular API client platform that makes it easier for developers to create, share, document and test the APIs. I used postman as my apparent frontend, to test and check whether my APIs are working correctly or not.

The testing part proved very important. As, I went on making the required modifications in the APIs and then testing them again to remove errors. After the entire backend API code was tested and verified using Postman, the project was completed. As a result, I had finished development of the frontend design layout and backend API code for web version of the Machine Learning Toolkit application serving the project completion.

Project Purpose

This Toolkit will be used as a direct application by Trisim Technologies and its clients. The toolkit will help the clients to perform Machine Learning operations (without needing them to have any programming knowledge). Also, the users of this toolkit will be able to build and compare various models in lesser time.

Conclusion

Learned about numerous new concepts, platforms and libraries. Implemented the concepts of backed web development. Also applied the creative skills to design the frontend visual layouts.

Developed the entire Desktop version of UI based Machine Learning Toolkit. Designed the new front-end layout and workflow, and developed the backend APIs for web version of the toolkit.

Project 3

Problem Statement

Study the concept of Motor Current Signature Analysis (MCSA). Research on how Machine Learning techniques can be implemented with MCSA to detect faults in induction motor. Produce a well summarized documentation for future reference.

Objectives met

- Study the concept of MCSA.
- Find out the way to implement ML algorithms with MCSA to detect failure.
- Prepare a document with all the study and learnings.

Project Execution

This was a descriptive study project but lasted for a short duration compared to the previous ones. The subject of study (Motor Current Signature Analysis or MCSA) is well known technique worldwide but was completely new to me. Hence, I started with the basics.

MCSA is nothing but a condition monitoring technique used to diagnose problems occurring mainly in induction motors. The concept originates from the early 1970s and was first proposed for use in nuclear power plants for inaccessible motors and motors placed in hazardous areas. It is rapidly gaining acceptance in industry today. The related tests are performed online without interrupting production with the motor running under the load at normal operating conditions.

MCSA can be used as a predictive maintenance tool for detecting common motor faults at an early stage and as such prevent expensive catastrophic failures, production outages and extend motor lifetime. It can be used as a diagnostic tool and powerful addition to vibration and thermal monitoring (verifying a fault with more than one technology).

MCSA is a method from wider field of Electrical Signature Analysis (ESA), useful for analyzing not only electrical induction motors, but also generators, power transformers as well as other electric equipment.

MCSA monitors the current (more precisely, stator current or supply current) of the motor. Typical stator current monitoring system is illustrated in the figure below.

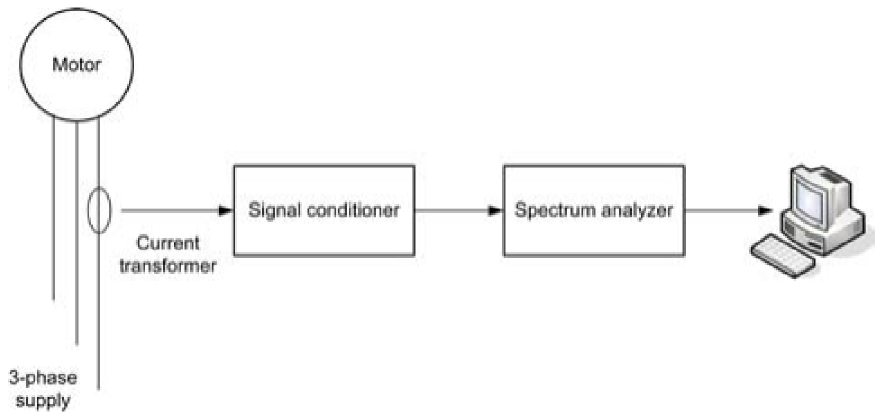

Fig4. Basic Current monitoring system

Single stator-current monitoring system is commonly used in MCSA (I.e., monitoring only one of the three phases of the motor supply current). This is because, if the motor is a 3-phase induction motor, the required supply current frequency in all the 3 phases is the same, which suffices the need.

But, current from all the 3-phases can also be used for monitoring while performing MCSA. Motor stator windings are used as a transducer in MCSA, which picks the signals (or induced currents) from the rotor (but also reveals information about the state of the stator).

Now, the first step is that the motor current is sensed by a Current Sensor (say a clamp probe or a current transformer) with a resistive shunt across its output, and recorded in time domain.

Second step, these recorded current signals are then led to a spectrum analyzer or a specialized MCSA instrument which gives us the recorded current signal information in frequency domain.

The third step, the obtained result (current signal in frequency domain) is studied and the fault is identified. This obtained result is called the current spectrum (also known as frequency spectrum or current signature). Current spectrum of a typical induction motor (which may have a fault) is illustrated in the figure 5. It is nothing but a magnitude (dB) vs frequency (Hz) plot for that signal.

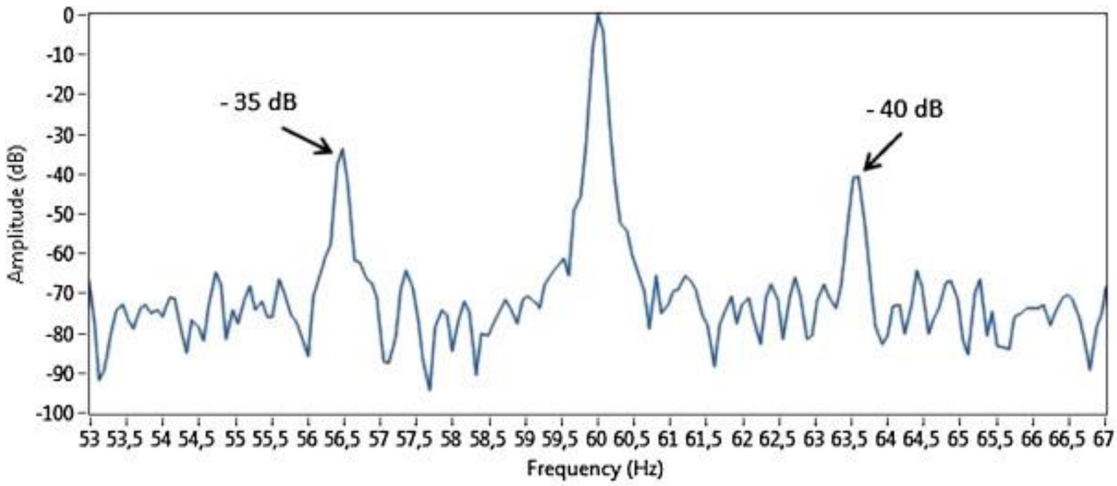

Fig5. Current spectrum of a typical induction motor

In the ideal case, motor current should be a pure sinusoidal wave (in time domain). This would lead to a current spectrum with a high amplitude peak only at the supply frequency which is usually 50 Hz or 60 Hz (high amplitude peak at 60 Hz shown in the figure above).

But in reality, many harmonics are present in the motor current. These harmonics are nothing but the result of different frequencies that arise due to faults (electrical and mechanical) in the motor. When such a signal is led to an analyzer, these fault conditions present in the motor further modulate the motor current signal and contribute to additional sideband harmonics (with a seemingly high amplitude) which then appears in the current spectrum.

Why does this happen? Faults in motor components produce corresponding anomalies in the magnetic field and change the mutual and self-inductance of the motor that appear in the motor supply current spectrum as sidebands around line (or supply or grid) frequency.

Based on the obtained fault signatures, motor faults can be identified and its severity can be accessed. Frequency range of interest in MCSA is typically 0-5 kHz. This, according to Nyquist theorem, requires a sample rate of at least 10,000 samples per second (for 5 kHz, the maximum frequency in range). During the test, the motor should be run at loading greater than 70%.

When a sufficient number of current signatures that correspond to various motor fault conditions are available, statistical methods may be applied for the task of data classification. General overview of a system that uses statistical pattern recognition is illustrated below.

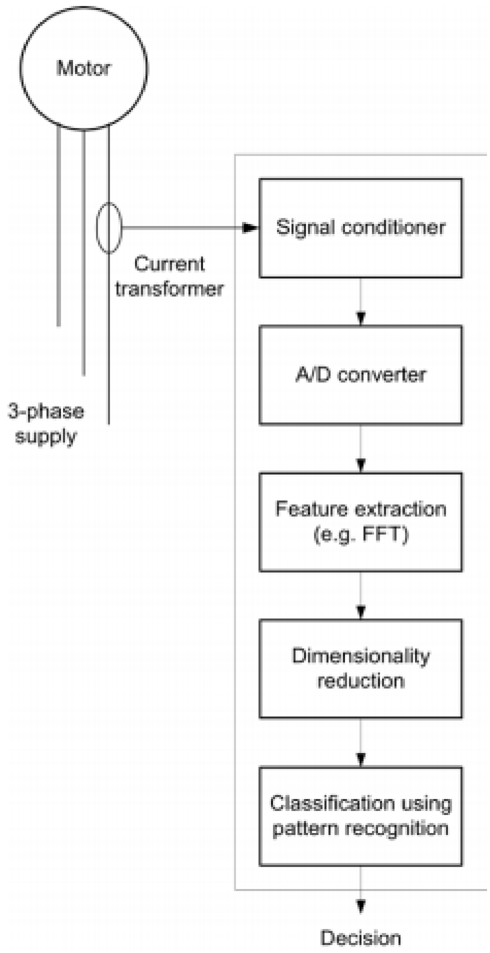

Fig6. Implementing Pattern recognition algorithm with MCSA

After the signal conditioning, analog-to-digital conversion is performed. Some pre-processing techniques (e.g., FFT) are used for feature extraction. Dimensionality reduction follows in the next step. Classification of unknown current signatures is performed by statistical pattern recognition (algorithm) with recognizers previously trained on a large number of correct and faulty signatures (basically, a model previously trained on a large number of data points).

Support Vector Machine (SVM) is a machine learning method successfully applied to a wide range classification and pattern recognition problems. The most important benefit is efficiency of SVM in high dimensional classification problems. Classification can also be performed using decision trees like Classification and Regression Tree (CART).

The prediction system (ML model building procedure) consists of four steps: data acquisition, feature extraction, feature selection and fault classification as shown below.

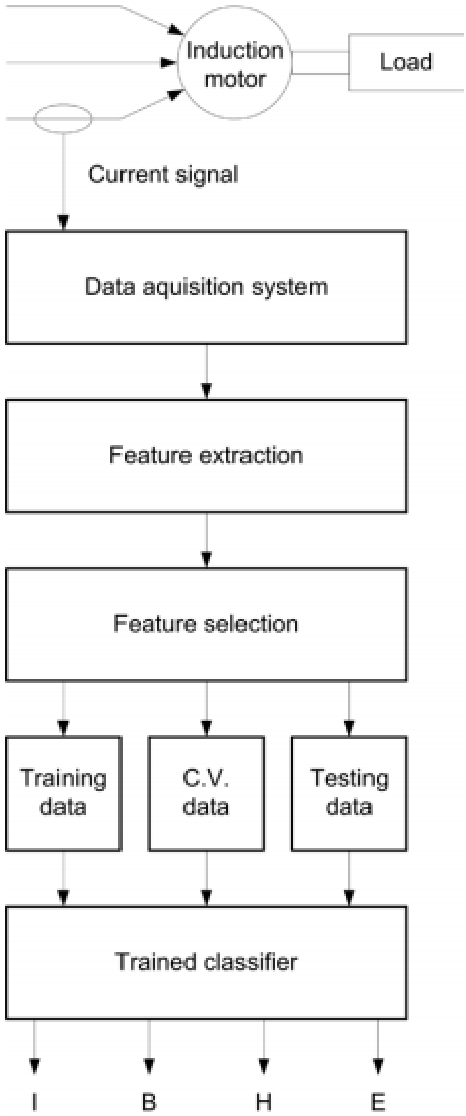

Fig7. ML process implemented with MCSA

During feature extraction, statistical parameters are calculated from the acquired current signal (FFT process). Feature selection and dimensionality reduction is performed using Principal Component Analysis (PCA). The trained classifier can then detect most common faults: stator winding inter-turn short (I), rotor dynamic eccentricity (E) as well as both of these faults (B). Output H corresponds to a healthy motor state. Or the output faults can be different from the ones specified above. They will be the same as specified in the training data set which will be used to train the model.

Project Purpose

The submitted documentation will be used by the company for future references.

Conclusion

Learned about the condition monitoring system used for induction motors and Machine Learning implementation on MCSA to detect failure patterns.

Upon project completion, Produced a detailed document containing all the subject matter study and learnings.

Experience and Learnings

This internship was majorly based on programming logic, Data Science, and Machine Learning concepts. In particular, it was about grasping the knowledge of Data science and Machine Learning and making a product on it using programming skills so that the users of this product can apply these concepts without needing to get into the very depths of these concepts. This product would further help its users to easily apply these concepts.

Not only the technical concepts, but this internship also encouraged my research and learning ability. During the initial period, I completed around 12+ online courses related to Data Science and Machine Learning. It helped me to gain tremendous knowledge and confidence in the concepts. Not only learning, but I also implemented this grasped knowledge by taking parts into online competitions to develop a grip.

Finally, I got an opportunity to use my programming logic and skills to implement what I had learned so far. Also, I got to learn many new concepts from my industry guide.

These were all the technical aspects of learning. Other than this, I also got to see how people who were on seemingly parallel paths converged in a wider combined goal of building something bigger. It introduced me to the corporate world and helped me gain the practical lens of looking at things. It questioned every design choice I made, going down to why I made those choices and if those really were the optimal way of doing things.

It encouraged me to find not just a good enough solution but a near optimal one for any problem at hand. It helped organize my work, in term of time and space, thereby making it an efficient process overall. It is one thing to have the knowledge and another to implement it. This is what this internship helped me in achieving.

Acknowledgements

I wish to express my gratitude to Prof. (Dr.) R. Jalnekar, Director, VIT Pune for providing the facilities of the Institute and for his encouragement during the course of this work.

I also express my deep gratitude to Prof. (Dr.) Mangesh Chaudhari, Head of the Department, Mechanical Engineering, VIT Pune for his guidance and support.

I would like to thank my industry guide Mr. Indrajit Jagtap and the employees of Trisim Technologies Pvt. Ltd. for providing me with guidance and help on every step of the way during the course of this internship and for imparting me with invaluable knowledge and teaching me the etiquettes of a professional employee. His knowledge, support and sense of direction throughout this internship are very much appreciated.

I would also like to gratefully acknowledge the enthusiastic supervision of my internship guide, Prof. Santosh Joshi for his valuable guidance, patience, and kind encouragement throughout the internship work that made me present this internship report in an efficient manner.

Finally, I wish to deeply thank my parents who have always provided me with constant care and have been very supportive and encouraging through all these many years.

Prathmesh Lonkar

June 2021

******