REPUBLIC OF TURKEY
ADANA ALPARSLAN TÜRKEŞ SCIENCE AND TECHNOLOGY UNIVERSITY

GRADUATE SCHOOL OF
ELECTRICAL AND ELECTRONIC ENGINEERING DEPARTMENT

IOT BASED PUMP PERFORMANCE MONITORING FOR AGRICULTURAL IRRIGATION AND PREDICTIVE MAINTENANCE WITH MACHINE LEARNING MODELS

İRFAN ÖKTEM

M.Sc.

ADANA 2024

REPUBLIC OF TURKEY
ADANA ALPARSLAN TÜRKEŞ SCIENCE AND TECHNOLOGY UNIVERSITY

GRADUATE SCHOOL OF
ELECTRICAL AND ELECTRONIC ENGINEERING DEPARTMENT

IOT BASED PUMP PERFORMANCE MONITORING FOR AGRICULTURAL IRRIGATION AND PREDICTIVE MAINTENANCE WITH MACHINE LEARNING MODELS

İRFAN ÖKTEM
M.Sc.

THESIS ADVISOR
Assoc. Prof. Dr. TUĞÇE DEMİRDELEN

ADANA, 2024

DECLARATION OF CONFORMITY

In this thesis study, which was prepared following the thesis writing rules of Adana Alparslan Türkeş Science and Technology University Institute of Graduate School, I declare that I provide all the information, documents, evaluations and results in accordance with scientific ethics and moral codes without resorting to any means or assistance that would be contrary to scientific ethics and traditions. I also declare that I refer to all of the articles I used in this study with appropriate references and accept all moral and legal consequences if a situation is found contrary to my statement regarding my work.

22/04/2024

[Signature]

İrfan ÖKTEM

ABSTRACT

IOT BASED PUMP PERFORMANCE MONITORING FOR AGRICULTURAL IRRIGATION AND PREDICTIVE MAINTENANCE WITH MACHINE LEARNING MODELS

İrfan ÖKTEM
M.Sc., Department of Electrical and Electronic Engineering
Supervisor: Assoc. Prof. Dr. Tuğçe DEMİRDELEN
April 2024, 71 pages

Irrigation pumps are used for agricultural irrigation in rural areas. Periodic maintenance of irrigation pumps in rural agricultural areas is costly and time-consuming; early detection of problems with real-time information avoids cost and time loss.

The main objective of this study is to monitor the performance of the pump installed for agricultural irrigation using the Internet of Things (IoT) and to apply regression models to the system data and to provide the optimum parameter values for the models with genetic algorithm (GA). Firstly, the automation system was installed and the temperature, humidity, current, voltage and pump vibration data were transferred to an IoT platform the named "Thingspeak". Then, an application was created with the Qt user interface creation toolkit. Thanks to this application, malfunctions occurring in the system were instantly notified to the user. Predictive maintenance of the system consists of a two-stage process in the field of machine learning. In the first stage, four different method models were evaluated. These models are linear regression, polynomial regression (PR), random forest regression (RFR) and support vector regression (SVR). In the second stage, the optimal parameters of PR, SVR and RFR models were determined using GA. As a result, the random forest regression model was found to be the most appropriate model for the dataset. It is hoped that this thesis will contribute to researchers interested in IoT-based pump performance monitoring for agricultural irrigation and predictive maintenance estimation using machine learning models.

Keywords: Genetic algorithm, irrigation pumps in rural areas, predictive maintenance

ÖZET

TARIMSAL SULAMA İÇİN IOT TABANLI POMPA PERFORMANSININ İZLENMESİ VE MAKİNE ÖĞRENİMİ MODELLERİ İLE KESTİRİMCİ BAKIMIN TAHMİNLENMESİ

İrfan ÖKTEM
Yüksek Lisans, Elektrik-Elektronik Mühendisliği Anabilim Dalı
Danışman: Doç. Dr. Tuğçe DEMİRDELEN
Nisan 2024, 71 sayfa

Kırsal alanlarda tarım sulamaları için sulama pompaları kullanılmaktadır. Kırsal tarım alanında bulunan sulama pompalarının periyodik bakımı maliyetli ve zaman alıcıdır; gerçek zamanlı bilgi alınmasıyla sorunların önceden tespit edilmesi, maliyet ve zaman kayıplarını önler.
Bu çalışmanın temel amacı, Nesnelerin İnternetini (IoT) kullanarak tarımsal sulama için kurulan pompanın performansını anlık olarak izlemek ve sistem verilerine regresyon modelleri uygulayıp genetik algoritma (GA) ile modeller için optimum parametre değerlerini sağlamaktır. İlk olarak otomasyon sistemi kurularak sıcaklık, nem, akım, gerilim ve pompa titreşim bilgileri "Thingspeak" adlı bir IoT platformuna aktarılmıştır. Ardından Qt kullanıcı arayüzü oluşturma araç seti ile bir uygulama oluşturulmuştur. Bu uygulama sayesinde sistemde meydana gelen arızalar anlık olarak kullanıcıya bildirilmiştir. Sistemin kestirimci bakımı, makine öğrenimi alanında iki aşamalı bir süreçten oluşmaktadır. İlk aşamada, dört farklı yöntem modeli değerlendirildi. Bu modeller doğrusal regresyon, polinom regresyon (PR), rastgele orman regresyonu (RFR) ve destek vektör regresyonudur (SVR). İkinci aşamada, GA kullanılarak PR, SVR ve RFR modellerinin en uygun parametreleri belirlendi. Sonuç olarak, veri seti için en uygun modelin rastgele orman regresyon modeli olduğu bulunmuştur. Bu tezin, tarımsal sulama için IoT tabanlı pompa performansı izleme ve makine öğrenimi modelleri kullanarak kestirimci bakım tahmini ile ilgilenen araştırmacılara katkı sağlayacağı umulmaktadır.

Anahtar Kelimeler: Genetik algoritma, kırsal alanlardaki sulama pompaları, kestirimci bakım

Dedication

I would like to dedicate my thesis to my family and Gazi Mustafa Kemal ATATÜRK…

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to the following individuals who have played a significant role in the completion of this project:

First and foremost, I would like to thank my supervisor, Assoc. Prof. Dr. Professor Tuğçe DEMİRDELEN, for her invaluable guidance, continuous support, and insightful feedback throughout the entire research process. Her expertise and dedication have been instrumental in shaping this work.

I am indebted to my esteemed Asst. Prof. Dr. İnayet Özge AKSU, Asst. Prof. Dr. Burak ESENBOĞA and Research Assistant Abdurrahman YAVUZDEĞER, for his critical insights, constructive criticism, and valuable suggestions that have greatly enriched the content and quality of my thesis. His contributions have been invaluable in refining my research and strengthening its overall impact.

I would like to extend my appreciation to my colleagues who have generously shared their knowledge and expertise, answering my questions and providing support during the preparation of both the theoretical framework and the practical implementation of this study. Their collaboration and willingness to help have been truly invaluable.

Finally, I would like to express my deepest gratitude to my family. Your unwavering support, love, and understanding have been the pillars that have carried me through this journey. The precious moments we have shared together as a family have provided me with the inspiration and strength to persevere. Thank you for your endless love, sacrifices and support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

5G	: 5th Generation Mobile Network
AC	: Alternating Current
ADC	: Analog to Digital Converter
AMR	: Automatic Meter Reading
ANN	: Artificial Neural Networks
API	: Application Programming Interface
AT	: Attention
CRC	: Cyclic Redundancy Check
DAC	: Digital-to-Analog Converter
DC	: Direct Current
DSA	: Differential Search Algorithm
DT	: Decision Tree
ELM	: Extreme Learning Machine
FTP	: File Transfer Protocol
GA	: Genetic Algorithm
GND	: Ground
GSM	: Global System for Mobile Communications
GPIO	: General Purpose Input/Output
GPRS	: General Packet Radio Service
GUI	: Graphical User Interface
HTTP	: Hypertext Transfer Protocol
I^2C	: Inter-Integrated Circuit
IMU	: Inertial Measurement Unit
IoT	: Internet of Things
IP	: Internet Protocol
JSON	: JavaScript Object Notation
KB	: Kilobyte
LiNCA	: Lithium Nickel Cobalt Aluminum Oxide
LiNMC	: Lithium Nickel Manganese Cobalt Oxide
LSTM	: Long Short-Term Memory
LTE-M	: Long-Term Evolution for Machines

MATLAB	: Matrix Laboratory
ML	: Machine Learning
MPPT	: Maximum Power Point Tracking
MSE	: Mean Square Error
MQTT	: Message Queuing Telemetry Transport
OLED	: Organic Light-Emitting Diode
PIR	: Passive Infrared
PWM	: Pulse Width Modulation
QoS	: Quality of Service
RAM	: Random Access Memory
RBF	: Radial Basis Function
RF	: Random Forest
RFR	: Random Forest Regression
RMSE	: Root Mean Square Error
R-squared	: Coefficient of Determination
SAIDI	: System Average Interruption Duration Index
SIM	: Subscriber Identity Module
SMA	: SubMiniature version A
SMBO	: Sequential Model-Based Optimization
SMS	: Short Message Service
SOC	: State of Charge
SVM	: Support Vector Machine
SVR	: Support Vector Regression
TCP	: Transmission Control Protocol
UART	: Universal Asynchronous Receiver Transmitter
VCC	: Voltage Common Collector
A	: Ampere
V	: Volt
F	: Farad
µ	: Micro
m	: Meter

1. INTRODUCTION

While agriculture is the foundation of food production worldwide, the sustainability and efficiency of modern agricultural practices depend on the effective use of irrigation systems. Irrigation systems are one of the most important components of modern agriculture. The efficiency and longevity of irrigation systems depend on regular maintenance of the pumps used. However, traditional maintenance methods can often be time consuming, costly and inefficient.

Predictive maintenance aims to predict a system's failure time using experience, physical laws, and machine learning techniques in order to replace faulty components before they fail, thereby reducing system downtime, lowering maintenance costs, and improving product quality.(Voronov et al., 2020)

Usually, faults in a machine result in higher levels of vibration and temperature. Therefore, the identification and early detection of bearing failures allows the maintenance team to fix the problem before it causes a serious breakdown. Thus, machine downtime is avoided or minimized.(Khorsheed and Beyca, 2020)

This thesis investigates the applicability of machine learning algorithms and IoT systems to ensure efficient maintenance of irrigation pumps. The maintenance of irrigation pumps is critical for the continuity of agricultural activities and crop productivity. Traditional maintenance methods are usually based on fixed periodic maintenance schedules and often fail to detect potential problems in a pump. For this reason, this thesis investigates how machine learning algorithms can be used to analyze data such as vibration, current, temperature, voltage, and humidity in the maintenance of irrigation pumps.

Machine learning algorithms are powerful tools for data analysis and prediction. By analyzing irrigation pump operating data, these algorithms can help identify critical factors such as maintenance scheduling, parts replacement needs, and performance prediction.

Machine learning (ML) can be divided into three categories: supervised, unsupervised, and reinforced learning. Unlike the first type, in which the predictors and response variables are known before developing the model, the second type only has observations. In contrast, reinforcement learning involves the agent engaging with the environment to learn actions and consequences.(Amruthnath and Gupta, 2018) Supervised machine learning aims to build a predictive model, which is used to make value predictions on a test data set where the predictive features are known but the value of the target variable is unknown.

This paper discusses the use of basic machine learning algorithms such as linear regression, polynomial regression, support vector regression, and random forest regression in irrigation pump maintenance. We will explain how these algorithms can be used in areas related to irrigation pump maintenance, such as data analysis, failure prediction, and maintenance scheduling. We will also discuss the role of genetic algorithms that combine these algorithms to extract meaningful information from complex data sets.

In addition, this thesis examines how genetic algorithms can be used to optimize the maintenance parameters of irrigation pumps. Genetic algorithms provide a population-based optimization approach to determine the optimal among different maintenance strategies. This approach can reduce maintenance costs and extend the life of the pumps with the right maintenance timing.

IoT employs sensors and communication technology to sense and communicate real-time data, allowing for faster calculations and better decision-making. IoT-based systems automate operations using sensors and communication technologies, while data analysis controls energy use. Fossil fuels, which account for over 80% of total energy consumption, can be monitored using IoT systems to improve energy efficiency, enhance knowledge about energy performance, and help to reduce the issues associated with managing power plants.(Motlagh et al., 2020)

IoT systems can be an important tool for real-time monitoring of irrigation pumps and data analysis. Data collected by sensors can be used to monitor the operating status of pumps, detect performance problems and determine maintenance requirements. The focus is on how IoT sensors can be integrated with irrigation pumps and how these systems can optimize maintenance processes.

Consequently, this thesis will investigate the availability of new approaches and technologies to optimize irrigation pump maintenance and increase agricultural productivity. This research is expected to make a significant contribution to the agricultural irrigation pumps sector.

1.1. The Objective Of The Thesis

In recent years, IoT technologies have revolutionized energy systems. IoT systems combine components such as sensors, network connectivity, and data analytics to monitor, control, and efficiently manage the status of irrigation pumps in real time. In this way, the status of the irrigation pumps can be continuously monitored, energy efficiency can be increased, and fault detection and maintenance processes can be optimized. Agricultural irrigation pumps, especially in rural areas, are inadequate in terms of maintenance; sometimes they are checked even though they are not needed, causing financial and time loss. The aim of the thesis:

- to predict the appropriate regression model for the next maintenance using IoT systems and machine learning algorithms to inspect and control the status of the irrigation pumps for agricultural irrigation in rural areas.
- to predict the next maintenance linear, polynomial, support vector and random forest regression methods are used.
- to detect situations such as overload, failure and upcoming maintenance, the study aims to develop a user interface using Python consisting of sensors and data acquisition devices to monitor the system in real time.
- to develop the regression models genetic algorithms were used.

It also highlights the growing importance of using machine learning to perform their periodic maintenance. The results of this work will be a valuable contribution to professionals and researchers in the energy sector and will guide the development of future predictive maintenance.

2. LITERATURE REVIEW

Irrigation pumps are constantly evolving to meet and sustainability goals. In this context, the integration of IoT systems and machine learning into irrigation pumps has significant potential for energy management and efficiency improvement. The sensors, data analytics, and remote access capabilities provided by IoT can make irrigation pump networks smarter, more reliable, and more flexible. A irrigation pumps in a rural area requires regular maintenance. Sometimes, an unforeseen situation may occur in the irrigation pumps, and the pre-scheduled maintenance may need to be done earlier. To eliminate this problem, there are studies in the literature that focus on issues such as the health information of the systems using IoT systems. Recent studies in the literature using IoT and machine learning are presented in detail below.

Motlagh et al., (2020) mentioned that modern technologies such as IoT offer numerous applications in the energy sector, namely in energy supply, transmission, distribution, and demand, increasing energy efficiency and the share of renewable energy. IoT uses sensors and communication technologies to sense and transmit real-time data, which enables fast calculations and optimal decision making. IoT-based systems automate processes through sensors and communication technologies and control energy consumption through data analysis. Fossil fuels, which account for about 80% of energy, can be monitored with IoT systems to increase energy efficiency, raise awareness about energy performance, and contribute to reducing the difficulties in managing power plants. It is used at different points in the energy sector, from buildings to smart cities, in terms of efficiency, reliability, and fast results. Among wireless communication sensors, long-term evolution for machines (LTE-M) makes sense to be used after global system for mobile communications (GSM) in terms of range.

Liu et al., (2020) In this literature study, the sensors, network technologies, storage layer structure, and user interface interaction used in the range from electricity generation to distribution and even small home appliances within the scope of smart home technology and the difficulties that may occur are mentioned. For weather maintenance personnel, underground vibration, robustness of towers, voltage sensor measurement quality, smart bolts against theft, and fire hazards due to the heating of cables are perception layered problems. The high number of sensors, large data transfers, and slow transmission of data are network layer problems.

Voltages are very high in transmission lines, finding the appropriate voltage for these sensors and ensuring that the signals do not pick up noise, corona events, or magnetic interference can make the connection problematic. It was mentioned that in case of malfunction and maintenance of transmission lines and towers, they can be videoed in 4K quality via drones with 5G connections and delivered to the relevant places.

Peter et al., (2022) focused on the fact that it takes a long time to find and resolve faults in power distribution. It is aimed at reducing the time of fault resolution over the Internet, i.e., the system average interruption duration index (SAIDI), using the Microsoft Azure cloud. In terms of visualizing the problem, it is concluded that a system such as Scada reacts later than a cloud controlled system.

Shahinzadeh et al., (2019) mentioned the role of IoT in electricity generation, transmission, and distribution. In the production of electricity, maximum power point tracking (MPPT) points in solar panels, gearboxes, actuators in wind turbines, etc. It contains sensors in many parts, and these sensors can report the health status of products, production, and intrinsic performance. In the transmission of electricity, demand supply, losses, phase shifts, and distribution tower problems are improved with IoT systems. In power distribution, IoT systems can improve the detection of problems in low-voltage transmission lines, the ability to implement emergency demand response, power loss management, remote monitoring and control during unplanned disasters, charging times for electric cars (at night or when the grid is not loaded), etc.

Nath and Thakur, (2018) stated that the fuse of a phase may blow on the low voltage side of the distribution transformer from time to time due to faults or overload. This leads to a loss of time in determining the cause of the fuse blowing and troubleshooting the problem, as well as a reduction in the service provided to the consumer. Therefore, in this study, it is aimed to determine which fuse has blown and to transmit this information to the control room using IoT systems. For this purpose, Zigbee (a low-power Bluetooth version with a distance of 10-75 m) and GSM technologies were used.

Onibonoje et al., (2019) aimed to minimize the power outage time for a house and to monitor the power consumption of the house instantly. In case of emergency (fire, overload), a warning system was created via short message service (SMS) or general packet radio service (GPRS), and protecting the house was brought to the fore.

Shaban and Alsharekh, (2022) focused on predicting future load demand by modeling residential consumption using machine learning, taking into account weather conditions. The goal is to make appropriate economic investments according to load demand. Energy savings can be achieved by allowing households to see the amount of consumption. Esp32 (Wifi module) voltage and current sensors were used, and the data was visualized by Thingspeak.

Jain et al., (2012) developed a GSM-based electricity meter for electricity distribution companies. Electricity meters are read continuously with a microcontroller via automatic meter reading (AMR) and billing information is sent to the user via SMS and email at the end of the month. The aim is to save manpower. Consumption in the past months can be analyzed so that it can be useful for the user to save energy.

Papakitsos et al., (2019) mentioned that IoT can be very helpful in agricultural activities. An IoT-based system was developed to receive and store information such as humidity, temperature, etc. in the cultivated area. The system consists of master and leaf nodes. The division into master and slave modules reduces the need for subscriber identity module (SIM) cards, energy consumption and cost. The nodes can cover a large area. All nodes consume ultra-low power and have long battery life. While the master node transmits the data to the cloud via GPRS connection, the slave nodes send the received measurements (temperature, humidity, etc.) to the master node. Texas CC1310 microcontroller, which has low power consumption in agricultural field and radio-frequency on it, was used and information such as temperature, humidity, etc. was obtained from the soil by universal asynchronous receiver transmitter (UART) communication from this controller and MSP430 microcontroller. Since the GSM/GPRS cellular network consumes more power than other subsystems, it can be supported by solar panels or large batteries.

Kwarteng et al., (2021) conducted a study to investigate power distribution transformer anomalies, system control, and health status. Transformers have a high demand for reliable power generation. Arduino was used as the processor to process and transmit the detected data; an ACS712 hall effect current sensor for current measurement; an ultrasonic sensor for oil level;

the DHT22 module for temperature and humidity measurement; a LM2596 buck converter for 5V and 9V voltage; a GSM module and Nodemcu (Esp32) were used to transfer the data to the internet. The Blinky application was used to visualize the data on the internet. In case of exceeding the threshold, a notification was sent by SMS.

Barot et al., (2020) conducted a study to investigate the IoT based air quality monitoring is targeted. The proposed system has also been tested indoors and outdoors at different quality service levels. The research work addresses communication topology design, accuracy, sensing throughput, and power consumption against QoS levels. Esp32, temperature, humidity, and air quality sensors have been used. The security of the data is emphasized by the protocol used.

Deny et al., (2021) created an IoT-based energy meter, which aims to monitor energy consumption through a GSM module. The data on the energy consumed can be sent to the cell phone in the form of SMS, and at the same time, the energy can be turned on and off. Energy consumption can be monitored remotely. Past use of energy consumption is recorded. They aimed to pay the bill through a cashless transfer. They used only the SMS feature of the GSM/GPRS module. They stated that they aim to transfer data to the internet via GPRS for future studies.

Kadechkar et al., (2020) emphasize that, with today's increasing dependence on electricity, the reliability of electricity transmission systems is of critical importance. To achieve this goal, the power grid needs to be continuously monitored, and the use of IoT solutions and low-cost sensors facilitates this monitoring. He introduces a new family of power connectors called "SmartConnector". These connectors include a microcontroller-powered thermal energy harvesting system and electronic sensors to monitor different parameters of the connector in real time. The measurements are sent to a gateway via Bluetooth 5 wireless communication, and this data is transferred to a database server for further analysis. The paper presents experiments evaluating the measurement accuracy of the SmartConnector against wired systems and testing the performance of the thermal energy harvesting system. It also provides a protection solution for electronic devices operating in high-voltage environments. This solution can be considered an important step towards increasing the reliability of power grids.

Campo et al., (2018) describe an open IoT solution for measuring energy consumption in factories, detailing the energy consumption of each industrial machine and the energy consumed for a specific industrial process. The solution is based on BatNET wireless communication technology and has been installed in 16 factories worldwide that produce steel and aluminum parts for the automotive industry. BatNET is based on open and standardized protocols and was originally designed for building and smart city applications. To meet the demands of the industrial environment, some adaptations in hardware, software, and firmware are required. This paper presents the implementation challenges faced in deploying this industrial IoT system in real-world environments. Special attention is given to the constraints motivated by factory requirements and embedded processing and data storage capabilities, as well as their solutions and end results.

Stolojescu-Crisan et al., (2021) propose a system that interconnects sensors, actuators, and other data sources for which various home automations are intended. This system is called qToggle and works using the power of a flexible and powerful application programming interface (API), which forms the basis of a simple and common communication scheme. The use of Google Assistant is supported by using the Raspberry Pi or Esp8266 instead of Arduino. Home automation is emphasized. qToggle is intended to be a complete smart home prototype with many functionalities such as automation, control, monitoring, and security. It is mentioned that an alert system can be added in future studies.

Asadullah and Ullah, (2017) present a low-cost and user-friendly remote controllable home automation system. This system is developed using components such as an Arduino board, a Bluetooth module, a smartphone, an ultrasonic sensor, and a humidity sensor. The proposed system includes a smartphone application that allows controlling a total of 18 devices, including home appliances and sensors, through Bluetooth technology. This represents a general purpose home automation system at a time when most existing home automation systems are designed for specialized purposes. However, the proposed system is only capable of controlling devices within a short distance.

Bhatt and Patoliya, (2016) using a wifi module, aimed to communicate many devices in the home with the modem and at the same time control them over the internet. The Rasberry Pi was used as a router. The visualization was done using an Android application called HubDroid.

Heterogeneous sensor and actuator nodes based on wireless network technologies are distributed in the home environment. These nodes communicate with a middleware that works as a home automation server and acts as an agent to facilitate the message queuing telemetry transport (MQTT) connection protocol. The middleware facilitates the control of wireless nodes in the local network and remote network. The proposed system is designed to be low cost and scalable to control various devices.

Che Soh et al., (2019) presents a home energy consumption monitoring system and provides alerts through Ubidots Cloud Services. In this research work, an IoT framework is implemented to monitor home energy consumption data and transmit this data to the IoT Cloud. The system uses a current sensor module (ACS712) connected with an Intel Edison microcontroller to measure power consumption in the home. Power consumption data is collected and stored in Ubidots IoT Cloud Services. Consumers can monitor their home energy consumption on a daily basis and thus control their daily energy use. The data collection experiment is carried out to see the amount of energy consumption in the home. Power consumption data is also displayed on the display before and after the load is connected to the system. The IoT dashboard allows the user to look at the daily power consumption using the IoT dashboard. IoT Cloud performs some event analysis on the collected power consumption data and sends alerts, recommendations to the homeowner via electronic mail and telegram, thus informing their customers about exceeding the limits of electrical energy usage and thus helping to reduce energy overuse.

Jose et al., (2020) propose a smart distribution board that replaces traditional distribution boards with solid-state circuit breakers for faster switching. This smart distribution board is integrated with IoT-based real-time monitoring and aims to improve energy management by informing the user about the current, power, and energy status of each sub-circuit. The distribution board also provides home automation by providing a user-friendly experience. In this literature study, a hall effect based current transformer, voltage sensor, Arduino, Esp8266, Amazon Alexa, and Blinky (Android mobile visualization and alert application) were used. The solid-state relay was controlled by Amazon Alexa. A warning system was created with Blinky. System control with Blinky was faster than with Amazon Alexa. Network security is important. The performance of the solid state circuit breaker was evaluated with MATLAB Simulink software.

Mallahi et al., (2022) propose a distribution board with a solar tracking system that can self-clean, display power consumption, and also cut the circuit in case of overcurrent. Arduino, dust sensor, servo motors, current-voltage sensor, relays, display, and Bluetooth were used. All input data is collected and transmitted via Bluetooth using a working prototype created on the Arduino board. The mobile application is also used to allow individual users to monitor their system in real time, instantly view their energy usage by alerting users about electrical system problems, and provide other important alerts and warnings.

Ramdham, (2022) aimed to examine the health status of the transformer. A nodemcu was used to transfer the data to the internet environment, a gas sensor for gas leaks that may occur in the transformer, a buzzer for warning, and a Thingspeak application to view the data on the internet. In the event of exceeding the threshold value, information can be sent via SMS. This low-cost system can be installed to provide remote monitoring of the transformer and can not only determine its health status but also help in predicting its lifetime.

Jaya Deepthi et al., (2019) used Arduino, GSM module, buzzer, and current transformer. If the difference between current transformers is above a certain threshold level, the buzzer gives an audible warning, and the transformer information is sent to the relevant people via SMS. In real-time applications, it is a very difficult process to realize and solve the theft. It is mentioned that theft is prevented in this study.

Ostertagová, (2012) focuses on the use of a polynomial regression model for analyzing curvilinear relationships between variables. The author employed the least squares method to estimate the model parameters and assessed the model's suitability. To evaluate the accuracy of the regression model, commonly used indicators were utilized. The analyses were conducted using the MATLAB computer program. Multiple regression analysis, which involves creating mathematical models with multiple variables, was also discussed. Polynomial regression models, which consist of consecutive power terms, were identified as a special case of multiple linear regression. The ordinary least squares analysis was highlighted as a widely used method for fitting regression models. However, it relies on certain assumptions, such as minimizing the sum of squared errors and assuming that these errors are uncorrelated random variables with zero mean and constant variance. The text concludes by mentioning the availability of various statistical tests to assess the validity of these assumptions.

Probst et al., (2019) conducted a literature review and benchmarking study for the random forest (RF) algorithm, examining the hyperparameters that affect the performance of the algorithm. The study investigates the effects of hyperparameters such as the number of trees, mtry, sample size, and node size on performance, finding mtry to be particularly influential. Additionally, different splitting rules and tuning methods' effects on performance are compared. The results suggest that tuning random forests can improve performance, although the effect is less pronounced compared to other machine learning algorithms like support vector machines. A suggested method is to use Sequential Model-Based Optimization (SMBO) to simultaneously tune parameters such as mtry, sample size, and node size. This study demonstrates that using the R package tuneRanger can lead to improved performance, but if computational speed is crucial, the tuneRF function of the randomForest package is recommended.

Hossain Lipu et al., (2023) investigated the use of RFR algorithm optimized with differential search algorithm (DSA) to develop an improved state of charge (SOC) prediction model. The proposed algorithm improves the SOC prediction accuracy by automatically determining the appropriate value for the tree and leaves of RFR. Experiments show that high SOC accuracy is achieved in different lithium-ion batteries such as lithium nickel cobalt aluminum oxide (LiNCA) and lithium nickel manganese cobalt oxide (LiNMC). The proposed algorithm has also shown robust performance under varying temperature conditions. It also achieved higher accuracy, compatibility, and robustness than other popular algorithms such as extreme learning machine (ELM), support vector machine (SVM), and long short-term memory (LSTM). Future work could include further validation of SOC prediction accuracy under the aging cycle test.

Wu et al., (2009) developed a new model, Hybrid Genetic Algorithm Support Vector Regression (HGA-SVR), which aims to optimize the kernel function type and kernel parameter values for SVR. Experimental results show that the proposed HGA-SVR model outperforms previous models and can be successfully applied to complex prediction problems. In environments such as the european network of institutes of technology (EUNITE) network competition, which is one of the test platforms, it is observed that it provides a significant increase in accuracy compared to other electric charge prediction models. In particular, the new HGA-SVR model is able to achieve the lowest error values in electrical load forecasting and successfully determine the optimal type and all optimal values of the SVR parameters.

Oliveira et al., (2010) stated that project managers in the software industry often rely on their previous experience to estimate the number of man-hours required for each software project. They emphasized that the accuracy of these predictions is important for the efficient use of human resources. They also noted that machine learning techniques such as radial basis function (RBF) neural networks, multilayer perceptron (MLP) neural networks, SVR, bagging estimators and regression-based trees have recently been used to estimate software development effort. They stated that these studies have shown that the accuracy of predictions depends on the parameter values of these methods and that the choice of input features can also affect the prediction accuracy. In this context, they stated that the aim of the study is to propose the use of a GA to improve the accuracy of software effort predictions and that simulations show that the proposed method improves the performance of machine learning methods and provides a significant reduction in the selection of input features. As a result, they stated that optimizing the selection of input features and the parameters of machine learning methods improves the accuracy of the software development effort and helps to understand the importance of each input feature by reducing model complexity. Therefore, they suggested that some input parameters can be neglected without loss of accuracy in predictions.

Üstün et al., (2005) stated that a method such as SVR has become popular as an alternative to the traditionally used partial least squares (PLS) regression technique for quantitative analysis of near infrared spectroscopic data. They stated that SVR is preferred due to its high overall performance and its ability to model nonlinear relationships. However, they stated that one factor limiting the use of SVR is the parameters that the user needs to define. Therefore, they emphasized the need to develop an automated optimization approach to determine the optimal SVR parameter settings. In this paper, we present an SVR parameter optimization approach based on genetic algorithms and simplex optimization and conclude that SVR is less sensitive to spectral noise and more robust than PLS. The GA/simplex optimization procedure is shown to be a fast and accurate alternative. Furthermore, the lack of physicochemical interpretation and some advantages of SVR over PLS are discussed.

Khorsheed and Beyca, (2020) investigate the integration of ML and decision-making techniques for early detection of failures in bearings, which are widely used in machines operating at high loads and rotational speeds, and timely delivery of accurate maintenance interventions. The accuracy of the Predictive Maintenance (PdM) module is tested on real industrial datasets and an efficient PdM methodology using different ML algorithms is proposed. The results show that the tested ML algorithms are successful and effective. Furthermore, a framework for assessing failure probabilities and determining appropriate maintenance interventions using utility theory is presented. As a case study, the model is applied to the forward thrust pumping stations of the Waste Water Treatment Company (STC), one of the largest sewage stations in Qatar.

Li et al., (2021) proposed a hydraulic pump pressure signal prediction method as a solution to the failure prediction difficulties and poor real-time condition monitoring performance of aviation hydraulic pumps. This method involves analyzing the chaotic properties of the system with chaos theory and makes the time series pressure signal predictable. The data is processed by phase space reconstruction (PSR) and the pressure signal is predicted using SVR model. The PSR-GA-SVR model has higher prediction accuracy than the other models, reducing the minimum mean square error (MSE) up to 73.40% and the mean absolute error (MAE) up to 90.41%. They also found that the confidence level of PSR-GA-SVR is high and the coefficient of determination is greater than 0.98.

Cerrada et al., (2016) investigated new methods to improve the reliability, efficiency and accuracy of gear fault detection. The focus of the research is on the identification of specific condition parameters extracted from vibration signals and their use for a multi-class fault diagnosis system in a spur gear train. For this purpose, a diagnostic system was built using GAs and a random forest-based classifier. The results show that high classification accuracy is achieved along with an effective reduction of the condition parameters.

Mujawar et al., (2024) aimed to investigate the use of IoT-enabled smart irrigation systems in rice farming. For this purpose, sensors were used to collect environmental data such as temperature, water level, humidity in real time. Machine learning models such as artificial neural networks (ANN), support vector machines (SVM), decision trees (DT) and RFs were used to process the collected data. The experimental results showed the effectiveness of the system, with ANN showing the highest accuracy with 95.6%, followed closely by SVM with 93.2%, DT with 88.7% and RF with 86.5%. These performance indicators highlight the robustness and accuracy of the model for predicting environmental conditions for irrigation.

The effectiveness of each model is examined in more detail through confusion matrices. The effective fusion of IoT and machine learning reduces the possibility of resource misuse by allowing for a proactive response to evolving field conditions and fine-tuning the water supply. These results highlight the potential of technology-driven solutions to boost precision agriculture and the importance of sustainable practices that optimize yields while conserving resources.

Yousuf et al., (2024) developed and implemented a unified condition monitoring and fault detection system for AC induction motors. This system aims to improve the operational reliability of industrial motors by combining sensors, GSM communication and a cloud-based IoT platform. The system includes strategically placed sensors to collect critical data such as temperature, vibration, current, voltage and speed. These sensors provide data to an Arduino-based controller that takes care of data collection and processing. An alarm system and GSM communication are also included in the system, providing timely alerts for abnormal situations. In addition, users can remotely access engine health data and real-time status. Historical data is analyzed through the cloud-based Blynk-IoT platform. The system also utilizes relay modules to provide motor control and protection. The proposed system is tested using Proteus and Arduino and is able to effectively detect abnormal motor behavior. In this way, it is possible to prevent disasters by enabling predictive maintenance. The results show that the system can detect abnormalities in important parameters such as vibration, temperature, speed, three-phase currents and voltages with 99% accuracy.

This chapter of the thesis presents a comprehensive review of the literature on regression methods, genetic algorithms and IoT systems in electricity generation, transmission and distribution. According to the literature review, IoT systems have been involved in electricity generation, transmission and distribution. However, the use of IoT systems in power distribution boards has been found to be widely used to control the performance of the boards. Various data acquisition servers have been used to obtain the performance of the systems. IoT systems have even been used to improve agricultural productivity. The number of studies on predictive maintenance of distribution panels in power generation, transmission and distribution with IoT systems has increased. In this study, it is aimed to develop an application that can predict the maintenance date of the system, warn the user in case of failure and at the same time provide easy use.

Many different sensors, microcontrollers, communication modules, and servers where data is stored are used in the integration of IoT systems. In a comprehensive review of studies conducted in recent years, sensors, microcontrollers, and user interfaces used in IoT systems are presented in Table 2.1.

Table 2.1. Some examples of microcontrollers, sensors, and user interfaces used in IoT systems.

References	User Interfaces	Microcontrollers	Sensors
Peter et al., 2020	Microsoft Azure	NodeMCU	Fault sensors such as voltage, current, temperature, vibration, or light sensors
Nath et al., 2018	LabVIEW GUI	Arduino	Gsm/Gprs module and Zegbee Modem
Onibonoje et al., 2019	Application on Web	Arduino	Gsm/Gprs module, temperature, current and voltage module
Asadullah and Ullah, 2017	Application on Smart Phone	Arduino	Ultrasonic and soil moisture sensor
Shaban et al., 2022	Thingspeak IoT Platform	NodeMCU	Wifi, current and voltage module
Jain et al., 2012	Sms	Undefined	Gsm/Gprs module, current and voltage module, AMR sensor
Papakitsos et al., 2019	Cloud(Undefined)	Texas CC1310 ARM, Msp430	Gsm/Gprs module, Rf, temperature and humidity module,
Deepthi et al., 2019	Sms	Arduino	Gsm/Gprs module, buzzer, current module
Ramdham et al., 2022	Thingspeak IoT Platform	NodeMCU	Gsm/Gprs module, buzzer, current and voltage module, smoke sensor
Mallahi et al., 2022	Application on Smart Phone	Arduino	Gsm/Gprs module, buzzer, current and voltage module, dust sensor, servo motors, relays, display

3. MATERIALS AND METHODS

Most of the innovations are in the application part, built in the Python language. The embedded software will transfer data from the irrigation pump to the Thingspeak platform.

3.1. Embedded Part

Embedded systems have become an integral part of many aspects of our lives, from cell phones and home appliances to medical devices and automobiles. They are specialized hardware and software designed to perform specific tasks efficiently and accurately. These systems offer many benefits, such as improved functionality, security, and energy efficiency. They enable the operation of devices, the automation of processes, and real-time monitoring. Embedded systems are used in industries to enable automation, reduce errors, and optimize production processes. They also play a critical role in areas such as energy management, environmental protection, and smart cities. As technology advances, embedded systems are becoming more sophisticated, compact, and versatile, paving the way for a smarter and more connected world.

Overall, embedded systems have become an integral part of modern life, enabling convenience, efficiency, and sustainability. They continue to evolve and offer new opportunities across industries and applications.
Figure 3.1. shows the embedded system created to transfer the data from the irrigation pump to the cloud layer.

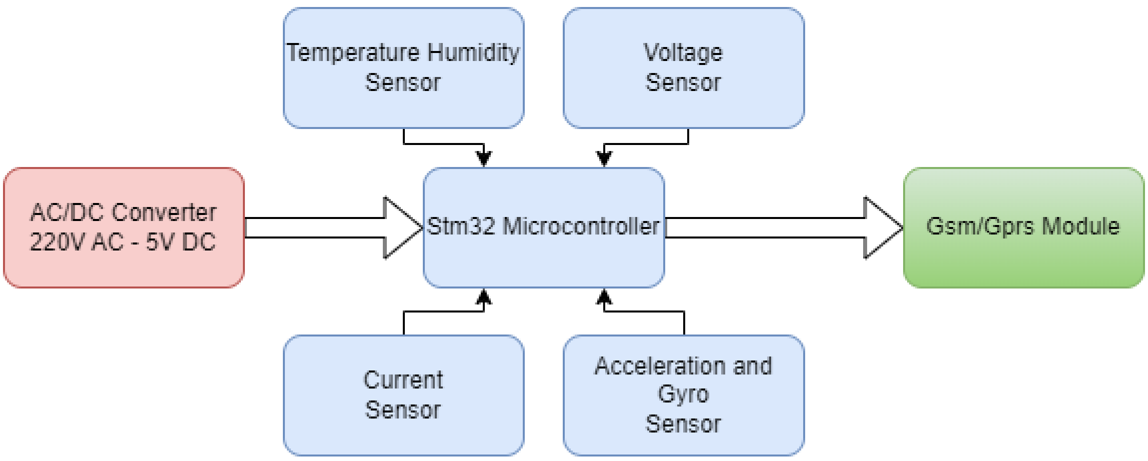

Figure 3.1. Block diagram of the embedded system

3.1.1. Microcontroller

STM32F103 is a popular 32-bit ARM Cortex-M3 microcontroller family developed by STMicroelectronics. This microcontroller comes with a high performance processor, a large memory capacity, and various integrated peripherals. The STM32F103 family is widely used in various application areas such as industrial control, automation, medical devices, and consumer electronics due to its advantages such as low power consumption, high-speed data processing capabilities, and wide connectivity options. There are many input and output pins on the microcontroller. Figure 3.2 shows the pinout diagram of the microcontroller.

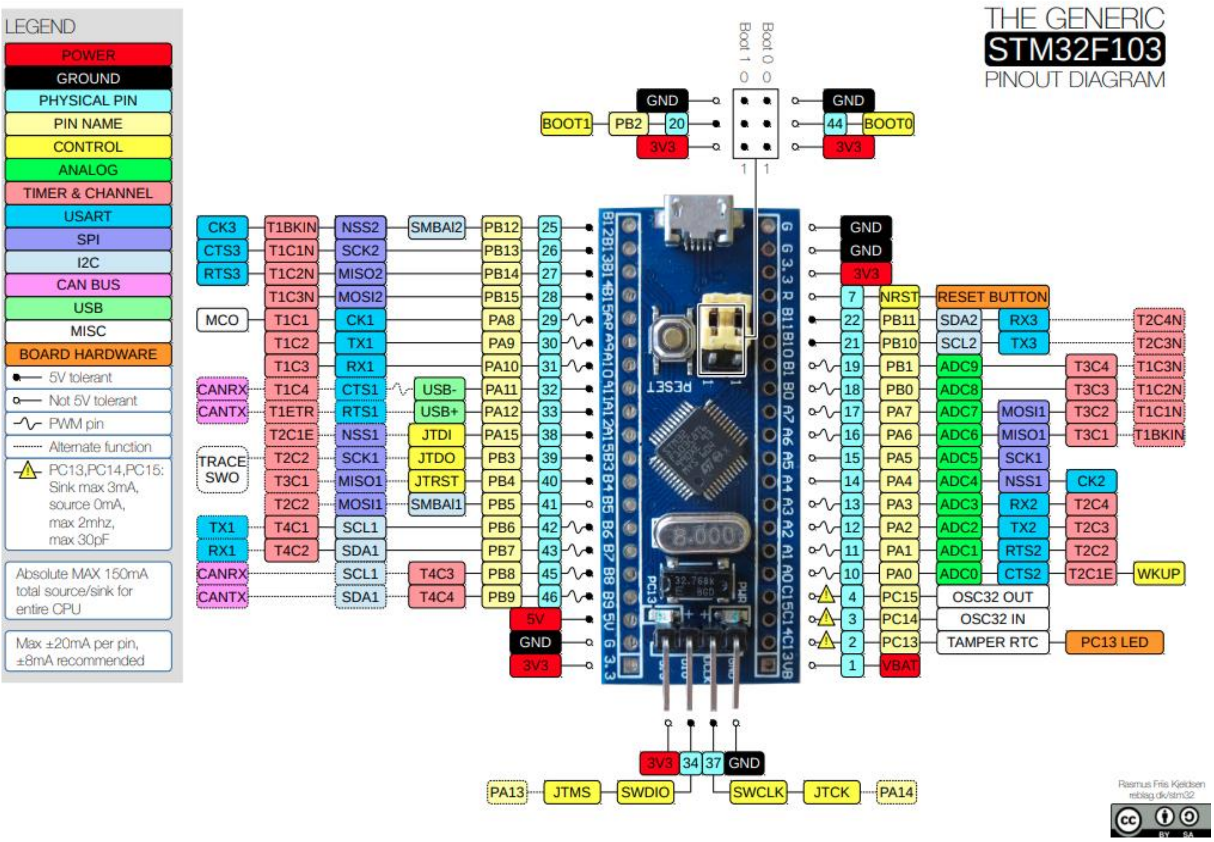

Figure 3.2. Microcontroller pinout diagram(Kjeldsen, 2020)

STM32F103 microcontrollers operate up to 72 MHz and provide high performance data processing thanks to their internal memory. Flash memory capacity typically ranges from 32KB to 128KB, while RAM memory capacity can range from 20KB to 64KB. In addition, the STM32F103 has a rich set of peripherals, such as various communication interfaces, GPIOs, ADCs, DACs, PWMs, and multiple timers. ("STM32F103 - Arm Cortex-M3 Microcontrollers (MCU)", n.d.) These features enable the microcontroller to be used in a wide range of applications and offer flexibility and performance to designers.

Our modules work with 5V. The microcontroller works at 3.3V. We connected our modules to pins with a 5V tolerance. Sometimes we used a resistor divider for our digital outputs. By taking the algorithmic average of the data we receive from the modules, we try to minimize the error.

3.1.2. Temperature humidity sensor

The DHT22 is a sensor used to make precise, accurate temperature and humidity measurements. As a digital sensor, the DHT22 performs accurate measurements and can transmit the results digitally. This sensor is widely used in the form of a sensor module containing a DHT22 chip. The working principle of DHT22 is that it uses a special semiconductor sensor for temperature and humidity measurement.("DHT22 Humidity and Temperature Sensor," 2024) The sensor detects the temperature and humidity levels in the environment and processes the data electronically. The DHT22 sensor can measure temperature and humidity at different scales depending on the user's needs. Common applications of the DHT22 include weather stations, air conditioning systems, greenhouses, automatic irrigation systems, and industrial control systems. This sensor is usually connected to microcontrollers or microprocessors to process and analyze the data. The DHT22 is a sensor that stands out for its accuracy, precision, and ease of use. It provides a reliable solution for many applications where temperature and humidity measurements are important. There are also many types of temperature and humidity sensors on the market. Since the sensitivity level is better than many temperature and humidity sensors, we decided that it makes sense to use this one. The sensor used is shown in Figure 3.3.

Figure 3.3. Temperature humidity sensor("DHT22 Humidity and Temperature Sensor," 2024)

The communication structure of the DHT22 sensor is based on the serial communication protocol. The sensor uses timing signals to send data to the microcontroller. The communication signal is shown in Figure 3.4. Here is an example that explains the communication structure of the DHT22 sensor step-by-step:

Initialization signal: The microcontroller sends an initialization signal to the sensor to send a data request. The initialization signal must be at least 1 millisecond low level (LOW) and then at least 250 microseconds high level (HIGH).

Sensor Response: The sensor detects the initialization signal and responds to the microcontroller. The sensor sends a low signal for a minimum of 80 microseconds, followed by a high signal for a minimum of 80 microseconds.

Data Transmission: The sensor transmits 40 bits of data in sequence. Each bit is represented by a signal that is low-level for at least 50 microseconds and then high-level for 26-28 microseconds. The data includes temperature and humidity readings.

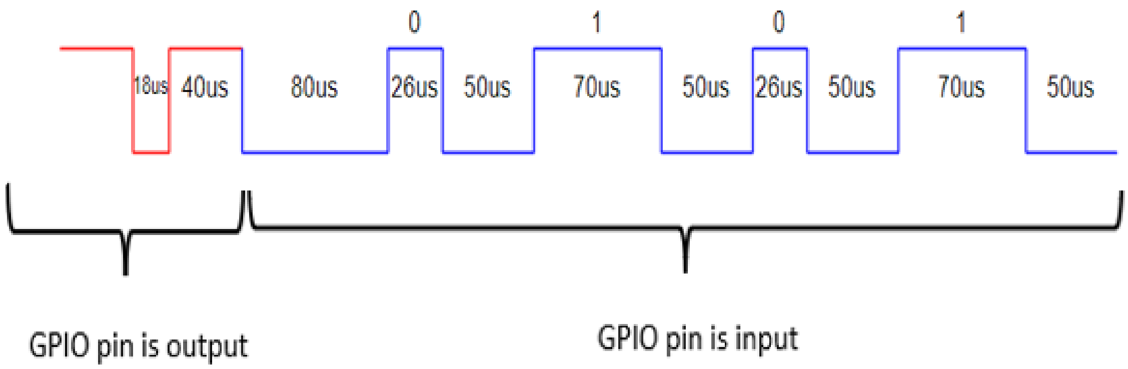

Figure 3.4. The duration of the high voltage defines if the bit value is one or zero.

So first the GPIO pin acts as an output to start the process, then it acts as an input and reads the data stream. At the end, the voltage returns to a high value, which is the role of the pull-up resistor attached to the data pin.

CRC check: After sending data, the sensor sends a CRC value to verify the accuracy of the data. The CRC is used to verify the integrity of the data.

The microcontroller follows these steps while receiving data from the DHT22 sensor and performing the necessary calculations to obtain the temperature and humidity values.

The DHT22 is a three-pin sensor device labeled VCC, DATA, and GND, which are to be connected directly to the microcontroller. It takes a source voltage between 3-5V DC for operations and gives out a digital signal. This digital signal is sent out to the microcontroller from the DATA pin. If the microcontroller is connected to GSM, it will send the data it receives to the Thingspeak platform.

3.1.3. Current transformer module

The ZMCT103C is a current transformer sensor for direct measurement of AC current. (“ZMCT103C,” 2021) This sensor features a high-precision semiconductor sensor and can be integrated into microcontrollers or other electronic circuits, converting AC current signals into low-level DC signals. The ZMCT103C provides an isolated measurement for current measurement and is used in many fields, such as power distribution systems, energy monitoring systems, and industrial control applications. Its high accuracy and reliability characteristics make the ZMCT103C the sensor of choice for AC current measurements. The consumed current is converted into a form that the microcontroller can understand through the sensor shown in Figure 3.5.

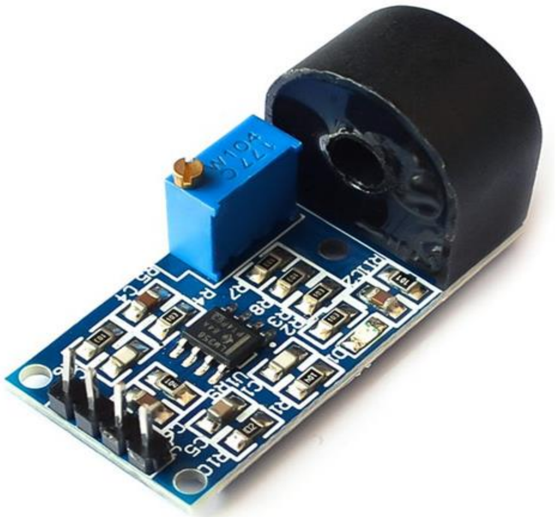

Figure 3.5. Current transformer module(“ZMCT103C,” 2021)

The communication structure of the ZMCT103C sensor includes the following steps:

Connection: You must make the proper connections to the ZMCT103C sensor. Typically, the primary winding of the sensor is connected in series with a section of the circuit through which the current to be measured flows. The secondary winding is connected to the GND and analog output pins of the sensor.

Analog Output: The ZMCT103C outputs the amplitude variations of the measured current as an analog voltage signal. The analog output pin of the sensor represents this voltage signal.

Measurement: To read the analog voltage output from the sensor, you need to use an ADC. The ADC on the microcontroller converts the analog output of the sensor to a digital value that can be processed by the microcontroller.

Conversion and Calculation: The ADC converts the analog output of the sensor to a digital value by sampling. The converted digital value can be used by the microcontroller to obtain the current value. This conversion process may vary depending on the microcontroller and ADC used.

3.1.4. Voltage transformer module

The ZMPT101B is a sensor for direct measurement of AC voltage. This sensor has a high-precision semiconductor sensor to provide an isolated measurement. The ZMPT101B converts AC voltage signals into low-level DC signals and can be integrated into microcontrollers or other electronic circuits.(Hareendran, 2022) Thanks to these features, it is used in many fields, such as power distribution systems, industrial control applications, and energy metering systems. High accuracy and fast response time make the ZMPT101B a reliable choice for AC voltage measurements. The system voltage is converted into a form that the microcontroller can understand through the sensor shown in Figure 3.6. The read voltage is connected to the analog input of the microcontroller through the transformer, which converts it to the appropriate voltage. Figure 3.7 shows the voltage entering and leaving the module.

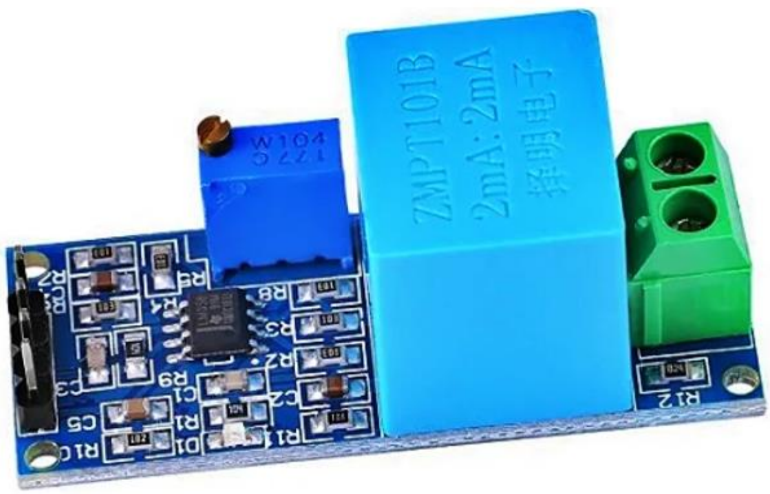

Figure 3.6. Voltage transformer module(Hareendran, 2022)

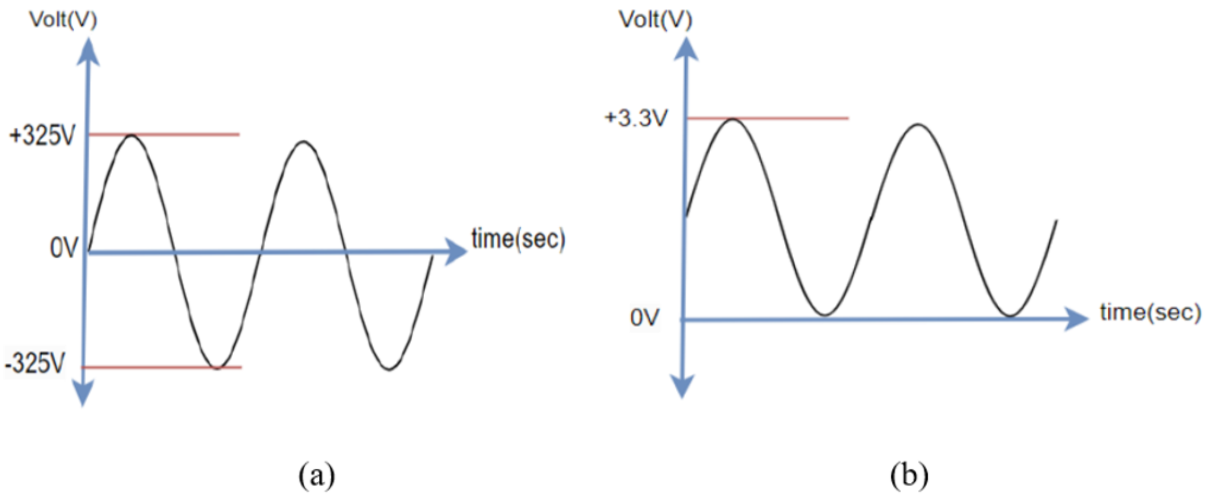

Figure 3.7. a) The input voltage from the AC mains(230V) b) The output voltage from the voltage transfor module

Similar to the ZMCT103C sensor, the read value is interpreted by the microcontroller. The RMS value is obtained by filtering the read values. Calibration is needed due to the analog circuit structure.

3.1.5. Acceleration and gyro sensor module

The MPU6050 sensor is an Inertial Measurement Unit (IMU) sensor known for its ability to measure vibration and motion.("MPU6050 Accelerometer and Gyroscope Module," 2021) In AC motors, the use of MPU6050 is an effective method to measure motor vibrations and analyze this information to gain valuable insights into the motor's condition.

Vibrations in motors often indicate early signs of issues such as wear, imbalance, mechanical problems, or connection faults. MPU6050 can measure the vibration spectrum of the motor, allowing the detection of these problems. Vibrations at specific frequencies, in particular, can point to potential failure points. This information can be used to plan regular maintenance activities and extend the motor's lifespan. The vibration data provided by MPU6050 is processed by a microcontroller or another control device. The analysis results can be used to determine which components of the motor are experiencing issues. For example, problems such as bearing wear, shaft imbalance, or gear issues can be identified through vibration analysis. This system can also contribute to preventing unforeseen failures. When the vibration characteristics of the motor are regularly monitored, potential issues can be identified in advance, and preventive maintenance can be applied to prevent costly failures. In conclusion, when combined with vibration analysis in AC motors, the MPU6050 sensor contributes to the effective management of motor maintenance and enhances system reliability. The sensor communicates with the microcontroller via the I^{2}C line. The sensor pinouts are shown in Figure 3.8.

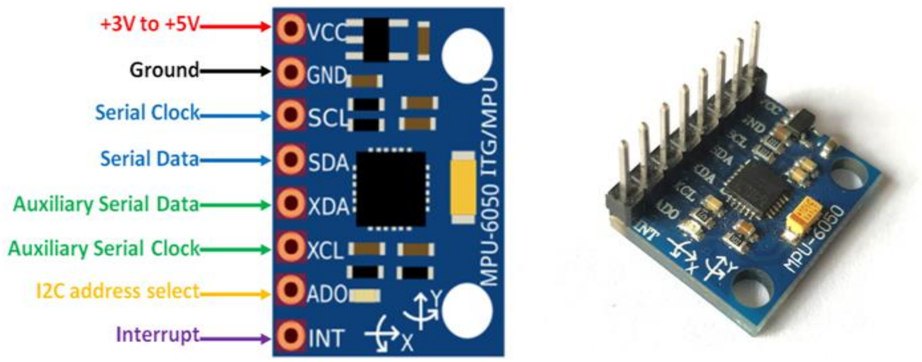

Figure 3.8. Acceleration and gyro sensor module(“MPU6050,” 2021)

3.1.6. Sim800L Gsm/Gprs module

The SIM800L GSM/GPRS module is a communication module known for its low power consumption and compact size. The module provides voice, data, and SMS communication by connecting to GSM networks.(Hareendran, 2017) The module has a SIM card slot, can be controlled using the AT command set, and can communicate with a microcontroller or other devices via serial communication. Figure 3.9 shows the pinout diagram of the GSM module.

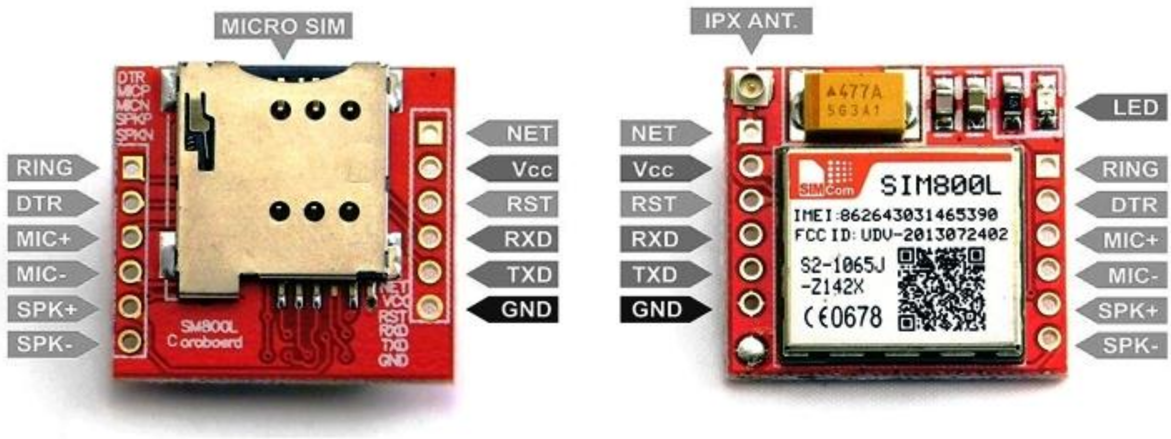

Figure 3.9. SIM800L Gsm/Gprs module(Hareendran, 2017)

SIM800L is a module that can operate in a quad-band frequency band. This allows it to operate in the 850/900/1800/1900 MHz frequency range, enabling it to be used in GSM networks worldwide. The module transmits data via a GPRS connection and can connect to the Internet using the transmission control protocol (TCP) / internet protocol (IP). These features allow it to be used in various projects, such as data-driven applications and remote control systems. The SIM800L has a built-in TCP/IP stack, so it can download web pages, send e-mails, and transmit data packets. The module supports communication protocols such as hypertext transfer protocol (HTTP), file transfer protocol (FTP), and MQTT, making it suitable for various IoT applications. In addition, the SIM800L is equipped with a voice processor that supports the functions of making and receiving voice calls. By attaching a microphone and speaker, conversational communication can also be provided. The SIM800L module allows users to easily insert and remove SIM cards thanks to the built-in SIM card slot. It also has a subMiniature version A (SMA) antenna connector so that signal reception and transmission can be improved by connecting an external antenna. The module can be controlled using AT commands over the serial communication line. A serial connection to a microcontroller or other devices can be established, and the functions of the SIM800L can be managed by commands. The SIM800L module has a wide operating voltage range (3.4V – 4.4V). It offers low power consumption thanks to its built-in power management circuit. It can also enter sleep mode to save power. It is generally known that the SIM800L module can draw an average current of approximately 2A when sending data via GPRS. However, these values may vary depending on the operating conditions of the module, communication speed, and network conditions. The instantaneous current consumption of the SIM800L module may vary depending on the data packet size, the connection speed, and the intensity of the communication. The instantaneous current consumption usually increases during data transmission and then returns to lower levels. When attempting to send data, the SIM800L would reset due to insufficient power. Fixed shutdown problem when sending SIM800L messages with a parallel 16V 470μF capacitor on the power line.

3.1.7. AC/DC converters

AC/DC converters are electronic devices that convert alternating current (AC) power to direct current (DC) power. An AC/DC converter typically operates as a switching power supply, converting the AC input voltage to a constant DC output voltage with low ripple. The example considered in this text is an AC/DC converter, shown in Figure 3.10, which converts an input voltage of 220V AC to an output voltage of 5V DC and an output voltage of 3.3V DC.

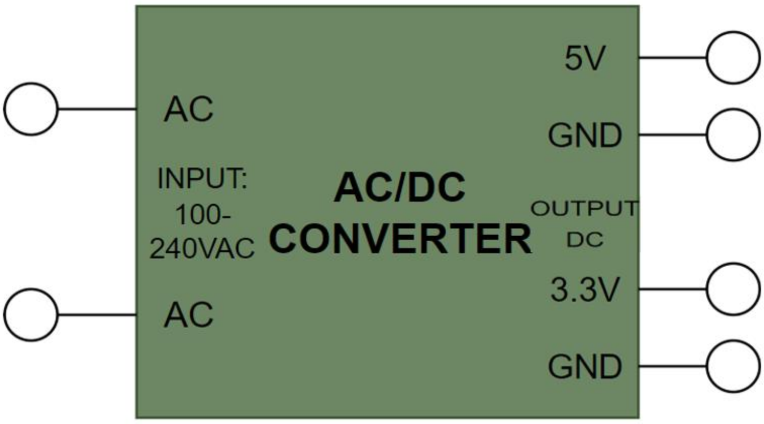

Figure 3.10. AC/DC converter

This type of converter works through a process that usually involves a series of steps. In the first step, the AC input voltage is converted to DC by a rectifier circuit. Next, a switching circuit converts the DC current into a high-frequency ripple current. This ripple is then converted to a low-voltage DC output by a transformer and rectifier circuit. In the final step, the DC output is smoothed and filtered to the desired level. Since the embedded system equipment operates on both 5V and 3.3V, a power converter is needed.

3.1.8. The design and manufacturing of prototype system

The aforementioned sensors and microcontroller were brought together as shown in Figure 3.11 and made ready for data acquisition. An OLED screen was used to facilitate the debugging process. Since it was a plug-and-play product, it was used only during debug time. The temperature sensor were placed inside the vibration sensor. Motor temperature data was obtained this way.

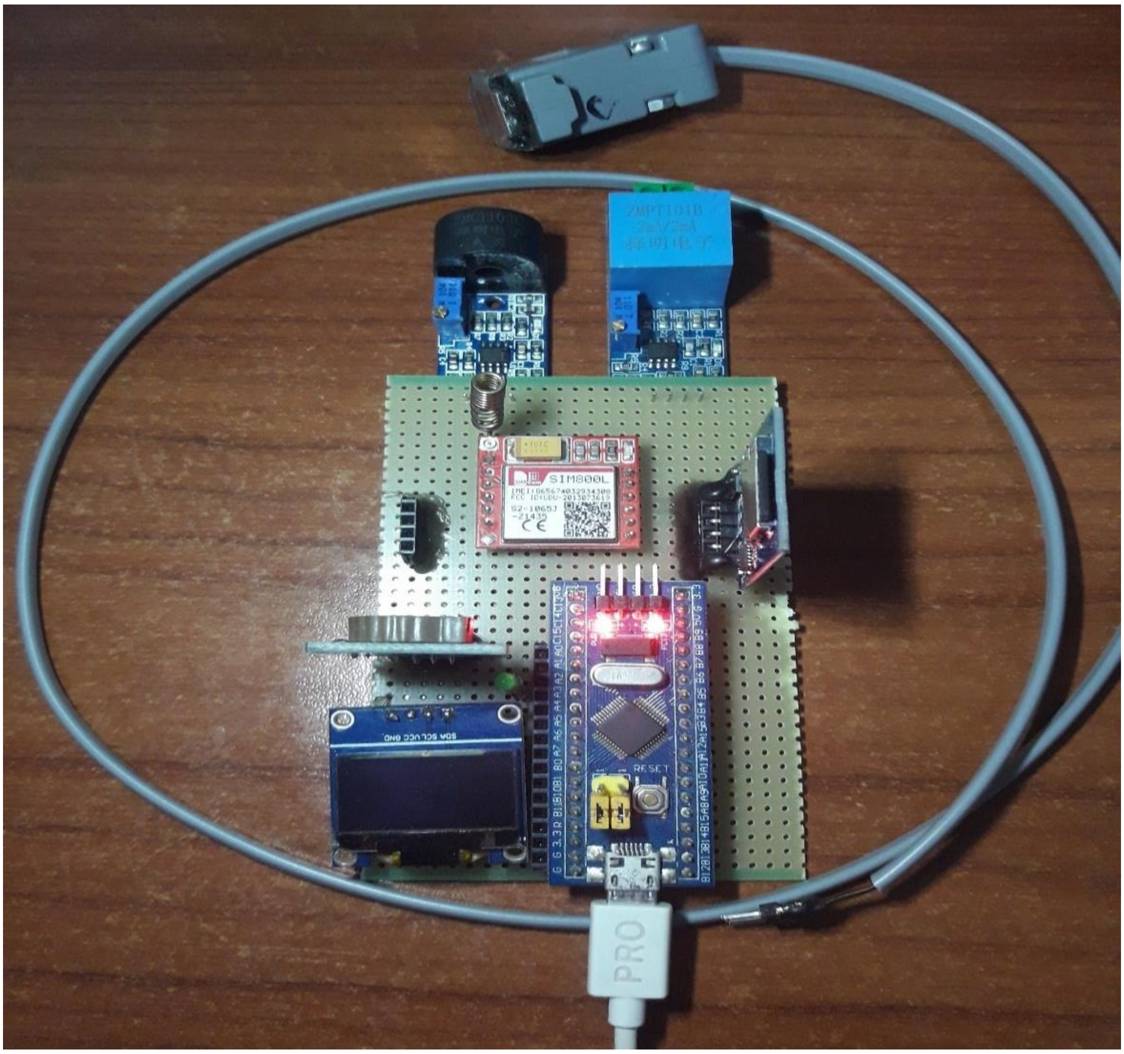

Figure 3.11. Experimental setup created to collect data

First, the modules were progressed step by step, as shown in Figure 3.12. Since the GSM module has not been added yet, vibration values are printed on the debug screen and saved to the memory card with time information. A nut was added to the fan to create vibration, and the fan was destabilized to simulate a faulty condition. A blinking green led was added to recognize that the system was recording to the memory card.

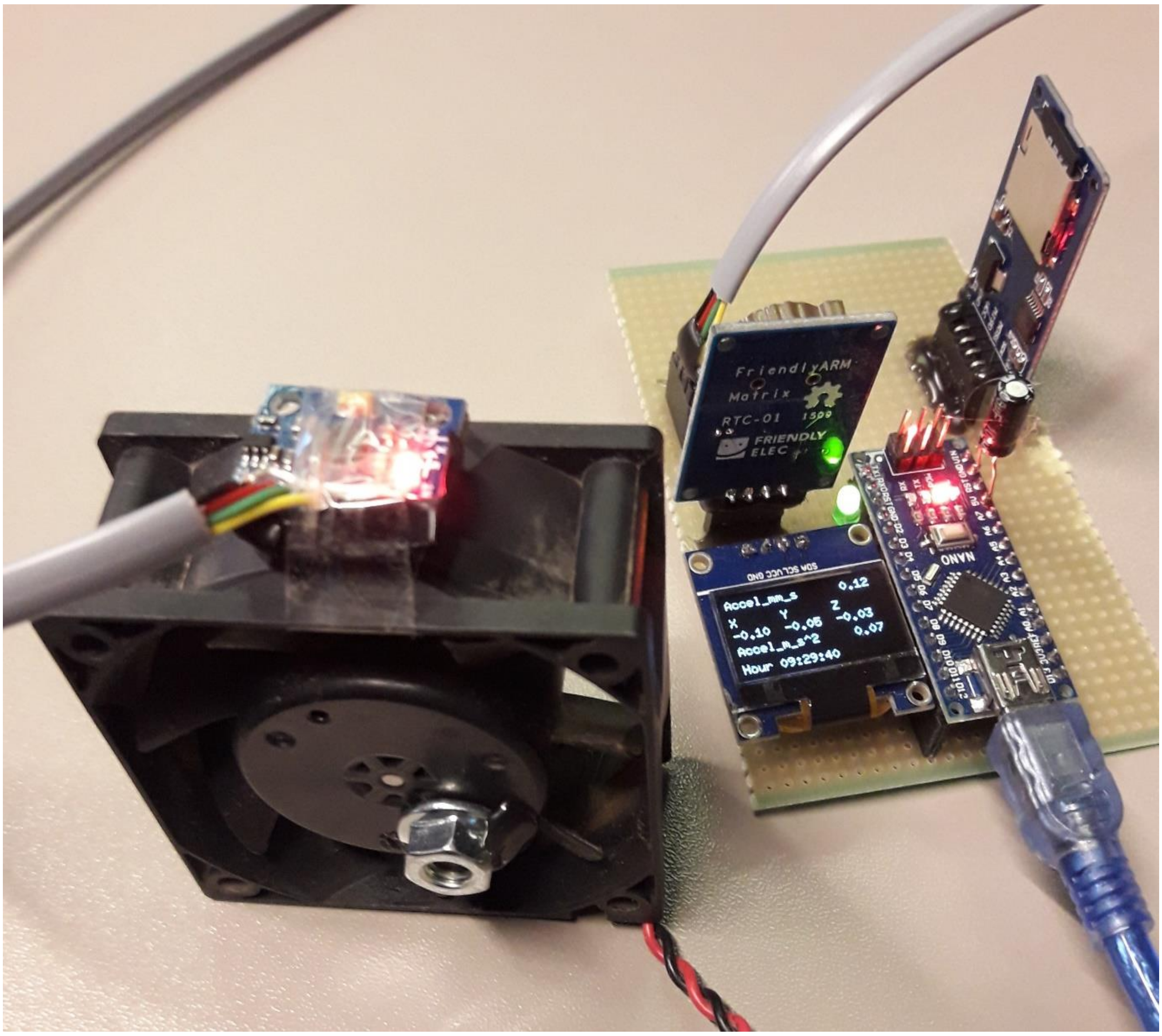

Figure 3.12. Images from testing the vibration module

The test rig was set up to obtain initial vibration and temperature data, as shown in Figure 3.13. For the accuracy of the data and reference adjustments, the vibration values of the 1.5kW motor during stop, start and loose connection were recorded on a memory card. Healthy motor and faulty motor values were recorded in the laboratory environment. A magnetic structure was added for the vibration sensor to hold the motor securely and the vibration sensor was placed inside the 3D print.

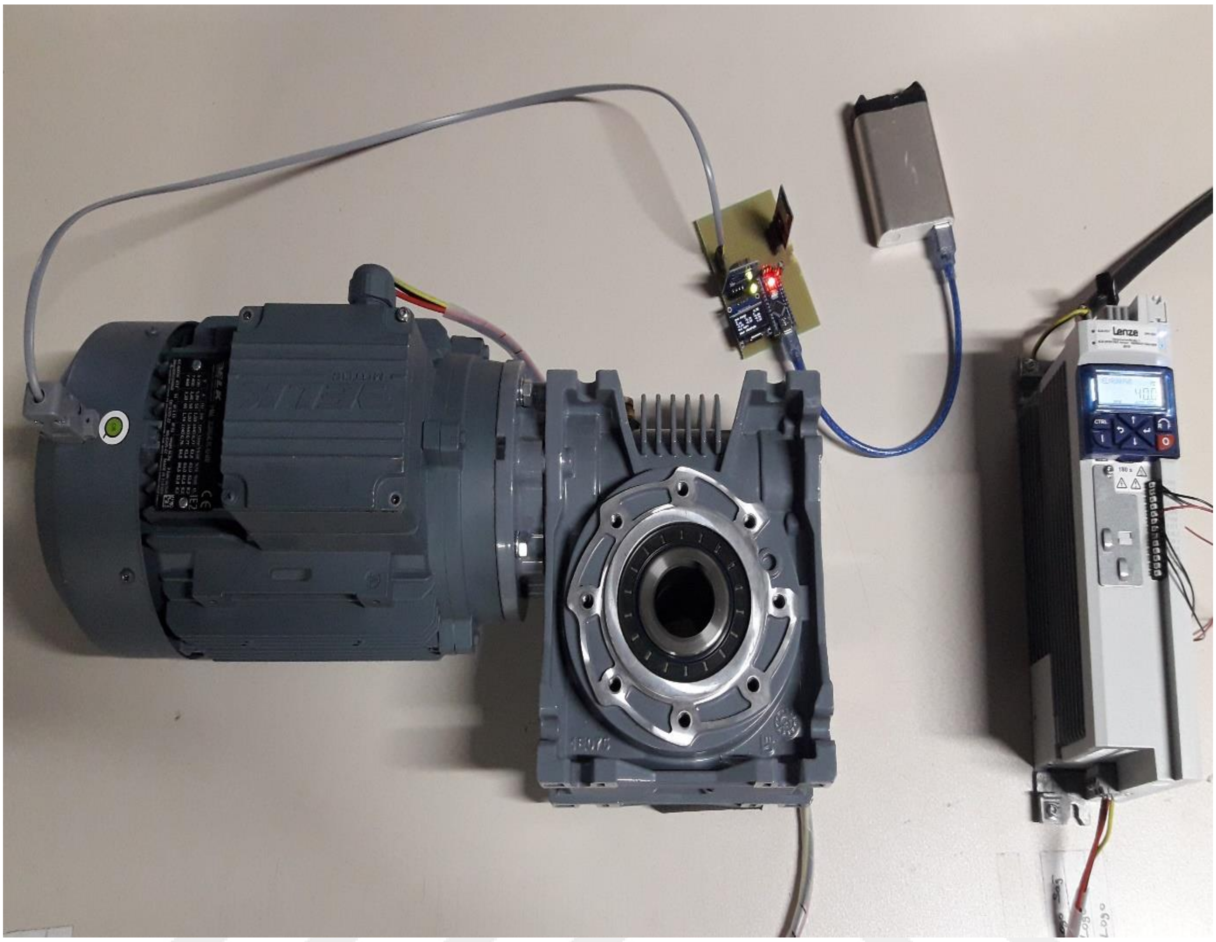

Figure 3.13. Obtaining vibration data on a 1.5kW motor

3.2. Iot Platform

ThingSpeak is a cloud-based IoT platform. It provides capabilities to collect, store, analyze, and visualize data from Internet-connected devices. (”IoT Analytics - ThingSpeak Internet of Things”, n.d.) ThingSpeak enables users to develop IoT projects in various application areas and streamlines data collection and management processes. When the application is logged in, you are greeted with this screen, as shown in Figure 3.14. From this image, it can be seen that the channel was created 1 year ago and the last data was received 3 months ago. Also, the recorded data can be downloaded as an excel file.

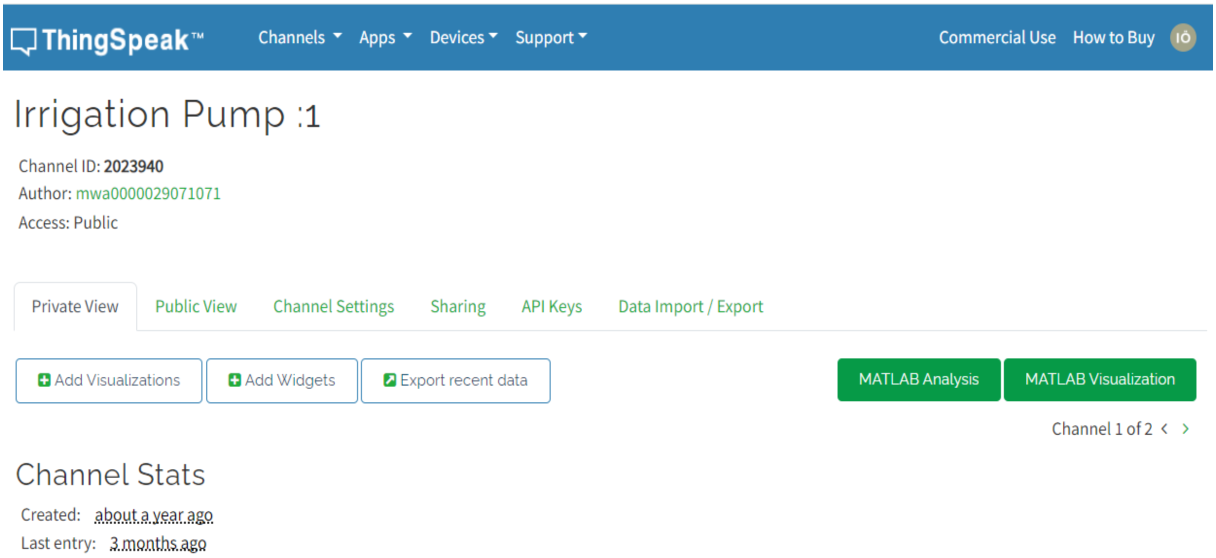

Figure 3.14. The Thingspeak user interface allows you to review incoming data.("ThingSpeak," n.d.)

ThingSpeak receives data from IoT devices using the MQTT and HTTP protocols. These protocols allow devices to send data securely. ThingSpeak can also receive data in JSON format, providing access to easy-to-use and easy-to-read data structures. The platform provides users with customizable channels. Channels are used to group measurements. Each channel gives users the flexibility to set data points (sampling times, data type, labels, etc.) and sharing options. Users can also share channels with other users to collaborate or integrate data. Figure 3.15 shows the vibration variation graph of the irrigation pump over time on the Thingspeak platform.

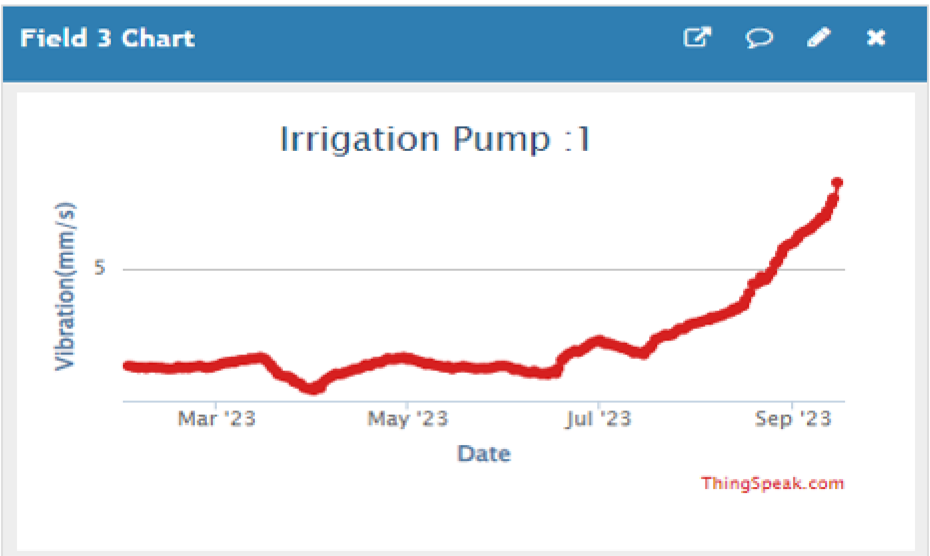

Figure 3.15. The vibration information of the irrigation pump can be displayed graphically.("ThingSpeak," n.d.)

ThingSpeak provides users with an easy-to-use interface for tracking and analyzing data using tools such as graphs, tables, and customizable visualizations. This interface provides users with the information they need to understand data, identify trends, and respond to events. Users can set thresholds and receive alerts to quickly identify critical situations. ThingSpeak provides powerful processing and analysis capabilities with MATLAB integration. MATLAB allows users to process data with more complex algorithms and analysis. This integration allows users to perform advanced processing such as scalable data analysis and machine learning.
In this study, only the Thingspeak platform is used to store data. Although Thingspeak has the ability to visualize the data, the developed Qt application is used to calculate the predictive maintenance date using machine learning algorithms.

3.3. Software Language

Python is a general-purpose programming language, and its use in embedded systems has increased in recent years. Embedded software usually refers to software that runs on resource-constrained devices such as microcontrollers or microprocessors. (“Welcome to Python.org”, n.d.) Python has become the language of choice for embedded systems because of its simple and readable syntax, extensive library support, and ease of use.
The benefits of using Python in embedded systems include rapid prototyping, code readability, and accelerated development processes. Python has a rich library ecosystem that can be used especially for data analysis, sensor data processing, and IoT projects. In addition, the Python language can be integrated with low-level languages such as C or C++, allowing for greater control and optimization of performance-critical parts. However, using Python in embedded systems can present some challenges. In particular, on resource-constrained devices, Python's memory consumption and high-level execution environment can cause performance issues. Therefore, for applications that require critical timing or resource-intensive systems, lower-level languages may be preferable. However, the use of Python in embedded systems has become a common choice due to its advantages, such as rapid prototyping, operating system independence, and data analysis. Python is also easy to use and open source, allowing developers and data scientists to quickly prototype machine learning projects and improve their applications. Furthermore, Python's vast library ecosystem supports many different machine learning tasks, such as data analysis, big data processing, deep learning, and natural language processing, enabling a variety of application areas.

Qt is a popular software development framework used to create user interface applications on a variety of platforms. (“Qt | Tools for Each Stage of Software Development Lifecycle”, n.d.) Qt is a library written in C++, but can also be integrated with high-level programming languages such as Python. Python, when used with Qt, provides a powerful combination for developing applications with user-friendly and visually appealing interfaces. Toolkits such as PyQt or PySide can be used to integrate Python with Qt. These tools add the flexibility of the Python language to the rich features and GUI components of Qt. Qt provides various components such as widgets, graphics, database operations, and network programming. Using these components, Python programmers can create fast and interactive desktop applications, games, data visualizations, and more.

The combination of Qt and Python accelerates the software development process and facilitates the creation of user-friendly interfaces. Qt's portability and platform independence make it possible to create applications that run on Windows, MacOS, Linux, and even embedded systems using the same code base. This makes Qt and Python a preferred choice and widely used in both hobbyist projects and commercial applications.

3.4. Machine Learning

3.4.1. Linear Regression

Linear regression is a statistical method used to analyze the relationship between two or more variables and use that relationship to predict the value of a dependent variable (usually a numerical value). Basically, it attempts to predict the value of the dependent variable using the combined effect of the independent variables.(Kumari & Yadav, 2018)

There are two basic types of linear regression: univariate (simple linear regression) and multivariate (multiple linear regression). Here are the basic formulas for both:

Univariate linear regression is a type of regression where there is only one independent variable. In this case, the regression equation is as follows:

$$Y = a + bX \tag{3.1}$$

- Y is the dependent variable.
- X is the independent variable.

- a represents the intercept and is the point where the regression line crosses the Y-axis.
- b represents the slope and shows the effect of a one unit change of variable X on variable Y.

This equation represents a line that attempts to best represent the distribution in the data set.

Multivariate linear regression is a type of regression with more than one independent variable. In this case, the regression equation is as follows:

$$Y = a + b_1X_1 + b_2X_2 + \cdots + b_nX_n + e \tag{3.2}$$

- Y is the dependent variable.
- X_1, X_2, ..., X_n are the independent variables.
- a represents the intercept and is the point of intersection between the regression plane and the Y-axis.
- b_1, b_2, ..., b_n represent the coefficients of each independent variable. These coefficients represent the effect of each independent variable on Y.
- e represents the error vector. Represents the differences between observed and predicted values, indicating how well the model does not fit the actual data.

Multivariate linear regression is used to assess the effect of multiple independent variables simultaneously and allows for more complex data analysis.

While correlation shows the strength of a linear relationship between a pair of variables, regression presents the relationship as an equation.(Kumari & Yadav, 2018)

3.4.2. Polynomial Regression

Polynomial regression is a regression method used to express the relationship between the dependent variable and independent variables in a non-linear way, in contrast to linear regression. This method allows for modeling the relationship between data using more complex curves or polynomials.

The basic formula for polynomial regression is as follows:

$$Y = a + b_1X + b_2X^2 + b_3X^3 + \cdots + b_nX^n + e \tag{3.3}$$

- Y is the dependent variable.
- X is the independent variable.
- a represents the intercept and is the point where the regression curve intersects the Y-axis.
- b_1, b_2, ..., bn are the coefficients of the polynomial terms. Each coefficient represents the effect of the X term of the corresponding degree.
- n represents the degree of the polynomial. The degree determines the number of terms used.
- e represents the error vector. Represents the differences between observed and predicted values, indicating how well the model does not fit the actual data.

Polynomial regression is used to obtain curves that better fit the actual distribution of the data.(Ostertagová, 2012) For example, in cases where linear regression does not fit the data, polynomial regression can give better results. However, it is important to choose the degree of the polynomial correctly. If the degree is chosen too high, the model may suffer from overfitting. Polynomial regression is usually implemented using data analysis software or mathematical computing tools. Steps such as selecting the degree and validating the model are important because incorrect degree selection or model overfitting can affect the reliability of predictions.

The fit models are shown in Figure 3.16. When calculating regression, keeping the polynomial degree too high leads to overfitting, while keeping the polynomial degree too low leads to underfitting. Finding the optimum value reduces the error rate considerably.

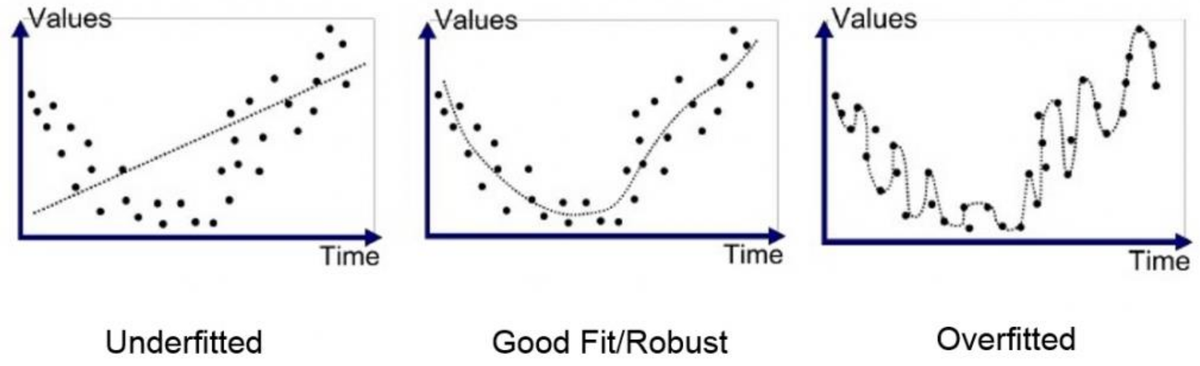

Figure 3.16. Example of evaluating the performance of a regression.(Raghav, 2022)

The concepts of underfitting, overfitting, and good fit are used to evaluate the performance of a regression or classification model.

- **Underfitting:** The model does not fit the data appropriately. In this case, the model fails to capture the complexity of the data set, and predictions are often too simple or result in low accuracy.
- **Overfitting:** The model becomes too sensitive to noise or random variations in the data set. In this case, the model overfits the data but may make poor predictions on new data.
- **Good Fit:** The model represents the data set appropriately. The model captures the key relationships in the data set and can make reasonable predictions on new data.

3.4.3. Support Vector Regression

Support Vector Regression (SVR) is a method used especially in machine learning and regression analysis. Basically, it is an adaptation of the classification algorithm called Support Vector Machine (SVM) to regression problems. The main goal of SVR is to create a hyperplane for a regression problem. Instead of separating data points into two different classes, this hyperplane creates a "corridor" between the majority of data points. This corridor is used to predict the target variable. Basically, SVR ensures that the data points stay within this corridor and at the same time tries to minimize the regression error.(Quan et al., 2022)

Because of their excellent classification accuracy and the availability of optimal algorithms even for non-linear problems, SVM is a popular choice for large sample size classification and regression. In this regard, SVM has garnered a lot of interest in a number of studies recently, particularly in the area of machine condition monitoring and diagnostics.(Khorsheed and Beyca, 2020)

The working principle of SVR is based on the concept of support vectors. Support vectors are data points that are critical in determining the hyperplane. These support vectors are the points that are closest to the hyperplane and are used to determine the hyperplane. The optimization goal of SVR is to minimize both the regression error (the difference between the predicted value and the true value) and the distance between the hyperplane and the data points. Therefore, the use of SVR as a regression model is particularly effective in the presence of outliers, such as outliers.

SVR has several parameters, the most important of which are the following:

- **C:** C is a parameter that controls the regression error. The larger the value of C, the smaller the regression error, but it can lead to overfitting of the model.
- **ε (Epsilon):** Epsilon sets a margin in the determination of the hyperplane. This margin controls the distance between the hyperplane and the data points.
- **Kernel:** The kernel function handles nonlinear relationships by transforming data points into high-dimensional spaces. The choice of kernel function significantly affects the performance of the model.
- **Gamma (γ):** A parameter specific to kernel functions, particularly important for non-linear kernels like the RBF kernel. Gamma defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. It plays a crucial role in determining the complexity of the decision boundary.
- **σ (Sigma):** The RBF kernel's width parameter in the SVR. Every training sample is taken as the center of the RBF kernel, and its distance from this center is used to determine its weight. This distance is computed using a measure called sigma. The domain is wider and the model fits are smoother when the sigma is smaller. The larger the sigma, the narrower the sphere of influence and the more sharply the model fits.

In summary, SVR is a powerful machine learning algorithm for solving regression problems. It is based on the concepts of support vectors and hyperplanes and attempts to minimize the regression error while controlling the distance between data points.

3.4.4. Random Forest Regression

Random Forest Regression (RFR) is a machine learning technique and is used to predict data sets consisting of many features. RFR can be thought of as a collection of decision trees. Each tree is trained with a random subset of the data set, and then the predictions are averaged together (for regression) or a class label is chosen (for classification).(Yuchi et al., 2019)
There is no interaction between the individual trees in a random forest. It appears that a random forest is an estimation approach that combines the results of several decision trees to obtain the optimal result.(Mercadier & Lardy, 2019)

Some important points about Random Forest Regression:

- **Decision Trees:** RFR involves a set of decision trees. Decision trees make predictions by partitioning the data based on an ordered set of tests of features. Each tree is trained on different subsets.
- **Prediction Method:** In the case of regression, each tree makes a prediction of one point of the sample. Then, the predictions of all trees are averaged to obtain the final prediction.
- **Parameters:** The main parameters of RFR are the number of trees, the depth of the trees and the size of the subset of features. These parameters affect the performance and generalization ability of the model.
- **Overfitting Reduction:** RFR generally reduces the risk of overfitting. This happens because each tree is only trained on a random subset and the trees are independent of each other.
- **Unbalanced Data:** Copes well with unbalanced datasets. Because each tree is trained on a random subset of samples, it has the opportunity to accurately model minority classes.
- **Randomness:** The concept of "randomness" refers to the fact that each tree randomly selects the features used in training. This ensures that each tree is different from each other and helps make the overall model more diverse and robust.

Random Forest Regression has a wide range of applications and is often used in industrial forecasting, financial analysis, medical diagnosis and other fields. This method is popular due to its strong predictive capabilities and having a robust model.

3.4.5. Genetic Algorithm

Genetic algorithms are founded on the survival principle of the fittest person in the population, which aims to preserve genetic information from generation to generation. Their main idea is to imitate events in natural systems that are necessary for evolution.(Patil et al., 2012)
Genetic algorithm is a heuristic algorithm inspired by natural selection and genetics that provides an optimization and search technique by pitting possible solutions against each other. Genetic algorithm is a search technique for adaptive global optimization that emulates the biological principle of evolution. When compared to conventional optimization algorithms, genetic algorithm offers the advantages of great global search ability, high efficiency, and quick

search speed without requiring the use of certain mathematical equations or derivative expressions.(Xue et al., 2020)

Figure 3.17 shows the genetic algorithm flow diagram.

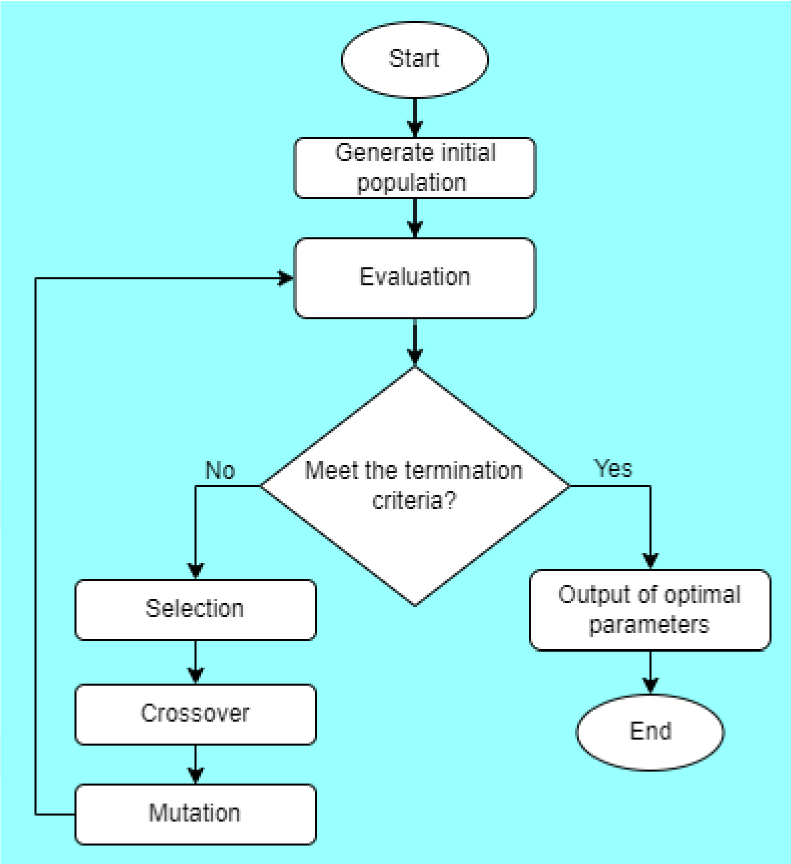

Figure 3.17. Genetic algorithm flow diagram.

The basic idea of genetic algorithm is to mimic biological processes such as natural selection and genetic crossover in a computer environment. This algorithm creates and evolves a population to optimize a problem or find the best solution. This evolution is iterative, and the population is improved by subjecting it to genetic operators such as natural selection, crossover, and mutation.

The working principle of genetic algorithms includes the following steps:

- **Creating the Initial Population:** In the first step, an initial population of random solutions suitable for the problem domain is created.
- **Calculation of Fitness Values:** A fitness function is used to measure how well each solution matches the problem. This function evaluates the performance of each solution and provides a fitness value.

- **Selection:** Individuals are then selected from the population based on their fitness values. This selection is based on the principles of natural selection to select more suitable solutions.
- **Crossover:** The selected individuals are brought together to create new individuals using the crossover operator. Crossover produces new solutions by combining the genetic material of the parents.
- **Mutation:** The newly created individuals are randomly modified using the mutation operator. This increases diversity and provides a wider search space.
- **Repopulation Reconstruction:** The old population is combined with the newly created individuals and sorted according to their fitness values.
- **Stop Condition Check:** The steps of calculating fitness values for repopulation are repeated until a certain stopping condition is met. This stopping condition can be based on criteria such as a certain number of iterations or a certain fitness level.

Genetic algorithms are used to solve a wide variety of problems in search fields and are particularly effective for problems with multiple variables, complex, nonlinear, or intricate structure. Advantages of this algorithm include the ability to compute in parallel, efficient results in large search spaces, and the ability to converge to the global optimum. However, there are also challenges such as tuning the parameters depending on the problem domain and evaluating the quality of the solutions.

3.4.6. Hybrid Methods

Hybrid methods that combine genetic algorithms (GA), polynomial regression, support vector regression (SVR), and random forest regression (RFR) can be an effective approach to solving regression problems.

Genetic algorithms are an optimization technique used in evolutionary computing that mimics the principles of biological evolution, such as natural selection and genetic crossover. Polynomial regression performs regression analysis by approximating the relationship between dependent and independent variables with polynomials. SVR is a derivative of the SVM used to solve regression problems and ensures that data points remain within a "corridor". RFR is an adaptation of the random forest algorithm to regression problems and is a tree-based ensemble method. To improve the prediction accuracy of the SVR algorithm, the parameters (C, σ) are modified by adding the GA, which is then used in conjunction with SVR to achieve multi-step prediction.(Xue et al., 2020)

Using the genetic algorithm to optimize parameters C and gamma can not only reduce model training time, but also eliminate overfitting and underfitting difficulties throughout the training process, resulting in improved prediction accuracy.(Li et al., 2021)

In order to guarantee the discovery of the global optimal values for every parameter, the population size and the number of generations were raised.(Wu et al., 2009)

Figure 3.18 shows the structure of hybrid methods.

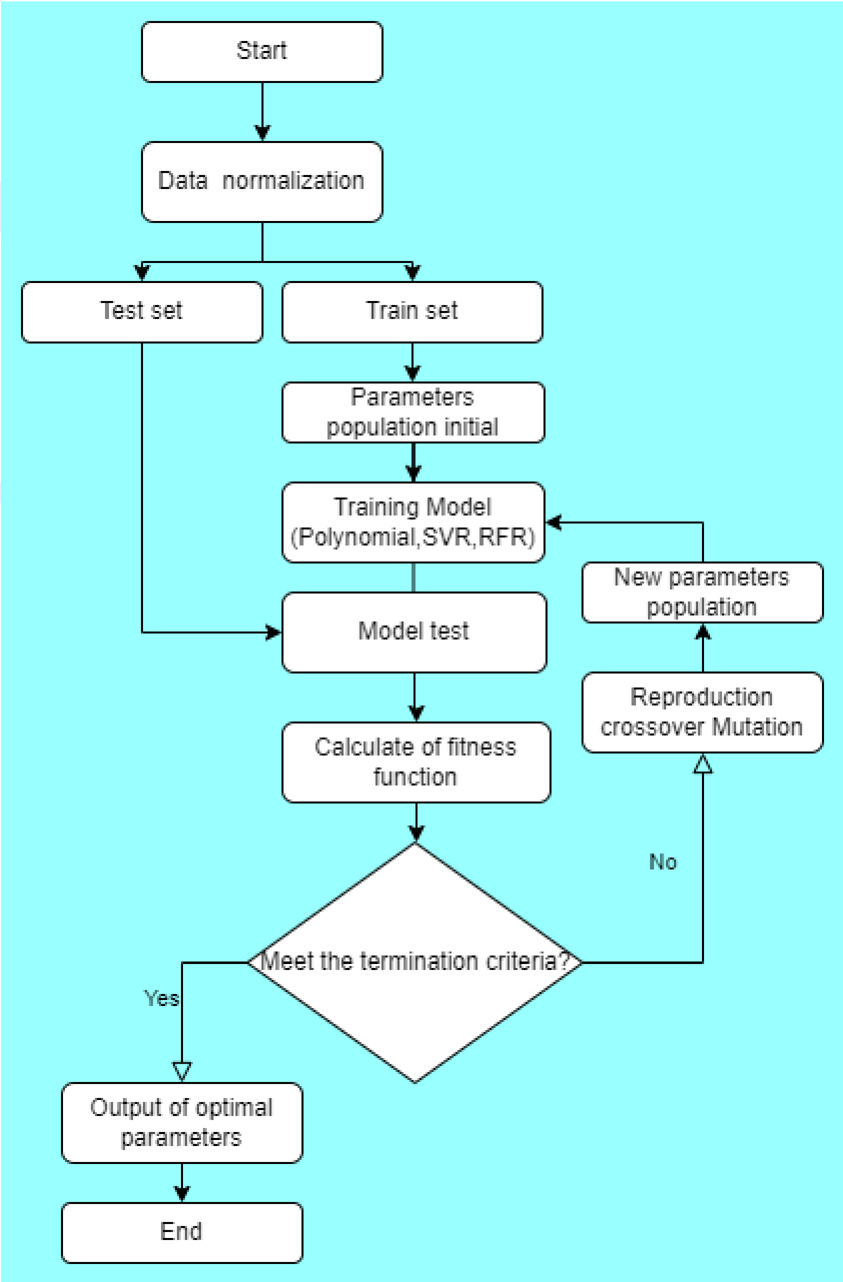

Figure 3.18. Structure of hybrid methods.

Combining these methods can provide a powerful approach to solving regression problems by exploiting the advantages of each method. For example, genetic algorithms are known for their ability to navigate multiple solution spaces and converge to the global optimum, while polynomial regression offers flexibility and simplicity. SVR is robust to outliers and performs well on low-dimensional datasets, while RFR is effective when working with high-dimensional datasets and multiple features. As hybrid methods, these methods can be combined, often compensating for the weaknesses of one method with the strengths of the other, and producing better results over a larger solution space. For example, genetic algorithms can optimize the coefficients of polynomial regression and can be used to find the optimal hyperplane of SVR. RFR, on the other hand, can be used in combination with genetic algorithms to increase the diversity among trees and provide a better generalization capability.

The RF-based fault classifier uses the features that were chosen as input attributes after the feature selection problem is solved using GA. The key contribution is to have an effective feature selection mechanism that can result in a greater diagnostic performance. Feature selection is an optimization problem that tries to minimize a particular classification performance parameter.(Cerrada et al., 2016)

Such hybrid methods are common in the literature and can be found in different variations according to their application areas. These methods are usually developed as hybrid approaches that are adapted and optimized for a specific problem or data set.

3.5. Instant Monitoring of Irrigation Pump Status

The machine learning algorithm and the user interface were written in Python. Information from the embedded system (temperature, humidity, vibration, current, voltage, etc.) was stored on the Thingspeak server. The fact that Thingspeak has many advantages, such as ease of use, free and paid options, integration capabilities, security, and real-time monitoring, played an active role in our server selection.

As shown in Figure 3.19, the data coming from the embedded software side is transferred to the Thingspeak server. The developed application then retrieves this data from the Thingspeak servers.

Figure 3.19. Block diagram showing the transfer of data from sensors to the application.

The data received from the Thingspeak platform with the Python software is transferred to the interface screen and the machine learning algorithm. The microcontroller circuit installed in the irrigation pump in the countryside sent the data it received from the sensors to the cloud, called Thingspeak. Using the Python programming language and the PyQt5 GUI (Graphical User Interface) framework, the goal is to predict the maintenance time of the irrigation pump by receiving data from the IoT data platform called Thingspeak and visualizing these data in Qt. Popup windows for alerts were created, and the Matplotlib library was used to visualize the data and perform operations on the data. The first step is to create threads in Python. This allows multiple tasks to be performed simultaneously. Data retrieval is done in Python using the Thingspeak API. At this stage, HTTP requests are sent to retrieve data from Thingspeak using libraries such as requests or urllib. The data received is returned in JSON format. This data is processed and made available for visualization on Qt.

PyQt5 is a popular framework for developing GUI (Graphical User Interface) applications in the Python language. (“PyQt - Python Wiki”, n.d.) Using PyQt5 classes such as QMainWindow or QDialog, a suitable window is designed to display the data. Pop-up windows are created to alert the user in some undesirable situations, such as when the temperature of the irrigation pump exceeds the threshold value. Matplotlib is a widely used visualization library in the Python language. In this step, the data acquired using Matplotlib is visualized with graphs. A Matplotlib plot area is created in the PyQt5 window, and the data is plotted in this plot area. Receiving input values for warning popups from the user. The data received continuously from Thingspeak using the threading method is stored in the queue structure. At the same time, the visualization and updating of the data received by the threading method is performed. When the device is turned on for the first time, since it has no data information, it warns that the maintenance time is estimated. Here the QTimer structure is used. In this way, the Qt window and Matplotlib graphs were kept up-to-date, and the data was monitored immediately. The block diagram of the application is shown in Figure 3.20.

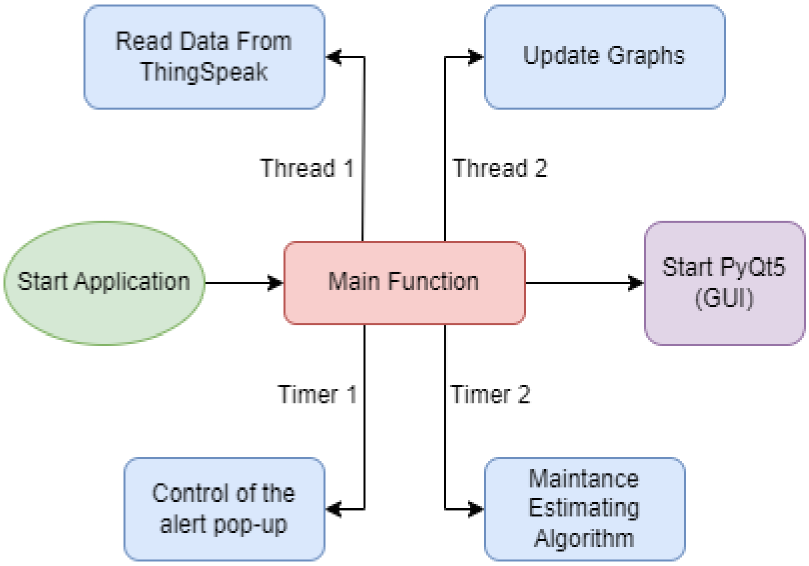

Figure 3.20. Block diagram of the developed software

The application developed in Python language was converted into exe format with Auto py to exe application and made suitable for use on computers with windows operating system. When the application is first opened, the user is presented with an interface, as shown in Figure 3.21. Graphs are created, and data from the server is expected. According to the incoming data, the intervals in the graphs are automatically adjusted in an optimal way. However, the user can examine the data in more detail by using the graph tools on the top left. These graphing tools are provided by the Matplot library.

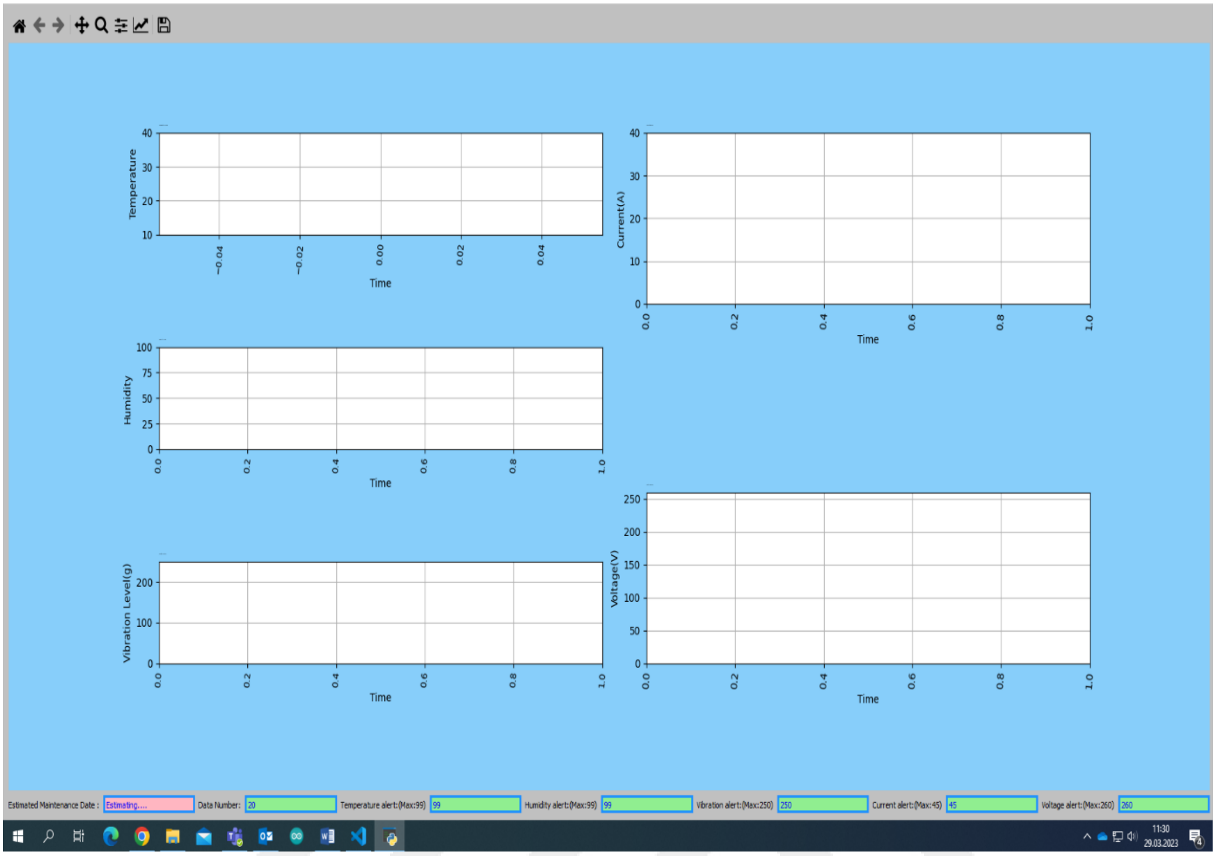

Figure 3.21. The splash screen of the developed application

In the bottom left corner of the application, "Estimating…." appears in the "Estimated Maintenance Date" field when the application is first opened, when data has not yet arrived from the server, and until data arrives. The Python thread module controls the server in the background. It prepares the incoming data and passes it to the machine learning algorithm.

As shown in Figure 3.22 the planned maintenance date was calculated as a result of incoming data from the application. It takes place in about 2 seconds after the application is opened. Incoming data is analyzed by directing it to a thread structure. In this structure, the number of data and whether any of the data is above the warning values are checked. Then the values are directed to the thread with machine learning, and the maintenance date of the irrigation pump is estimated.

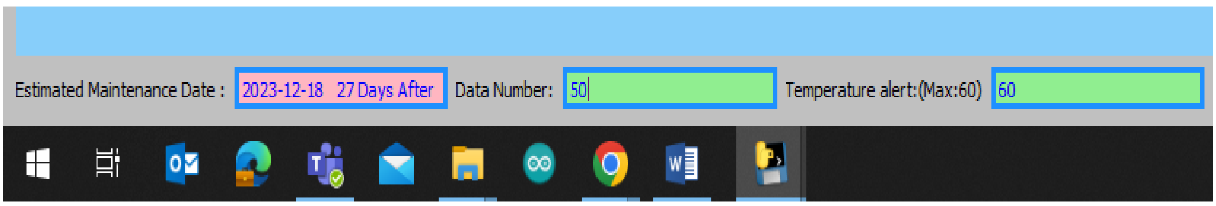

Figure 3.22. Estimated maintenance date generated by the application

4. RESULTS AND DISCUSSION

Irrigation pumps in rural areas need periodic maintenance. Maintenance time varies considerably according to the irrigation pump feature. With the developed application, maintenance costs were reduced by performing periodic maintenance. By detecting abnormal situations, the performance of the irrigation pump in rural areas has increased. You can observe that the Thingspeak platform is weak on data visualization. There are not enough toolboxes when we want to analyze the data in detail. With the Thingspeak platform, the maintenance date of the irrigation pump system cannot be predicted. So an application was developed using Python. Machine learning was added to the application.

4.1. Testing the developed application

For this thesis, a very comprehensive application was created, specifically using Python and the Qt user interface creation toolkit. The application was tested with data from the embedded system. The data that the embedded system sends to the Thingspeak platform is collected by the application in an array on the Internet. This array is then partitioned, and the data is classified. A problem with the irrigation pump in a rural area reached the user at the computer. The user sees a warning screen like Figure 4.1. If the limit values are exceeded, a warning pop-up appears on the GUI screen.

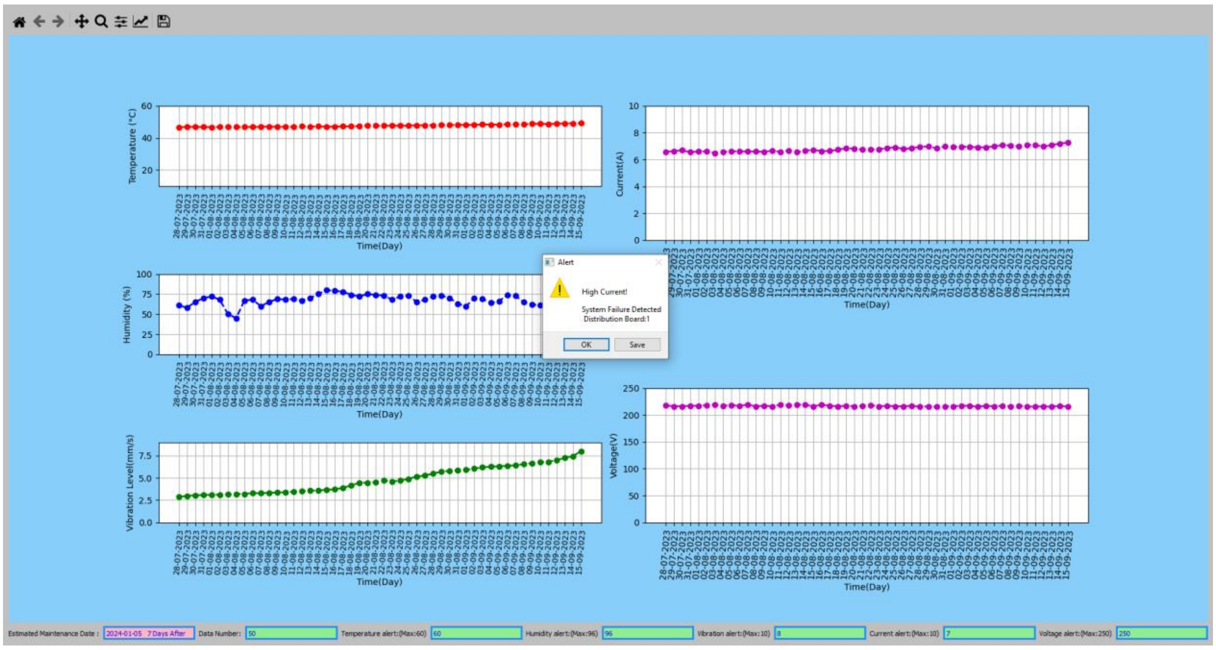

Figure 4.1. A pop-up screen alerting the user

In this pop-up screen, which feature has exceeded the limit value is indicated. Thus, the user is warned against an unpredictable error. It is not removed from the screen without user approval. Figure 4.2 shows the pop-up window in more detail.

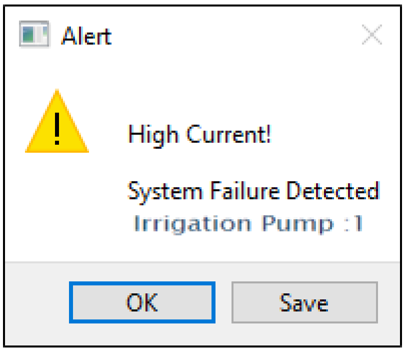

Figure 4.2. The detected error type is indicated to the user.

As can be seen in Figure 4.3, the granularity of the application is better than the Thingspeak platform. At the same time, if you want to show values in more detail, you can zoom in using the Matplotlib toolkit. In addition, you can personalize the graphic. When the graph is carefully examined, it is observed that the current value of the irrigation pump increases day by day.

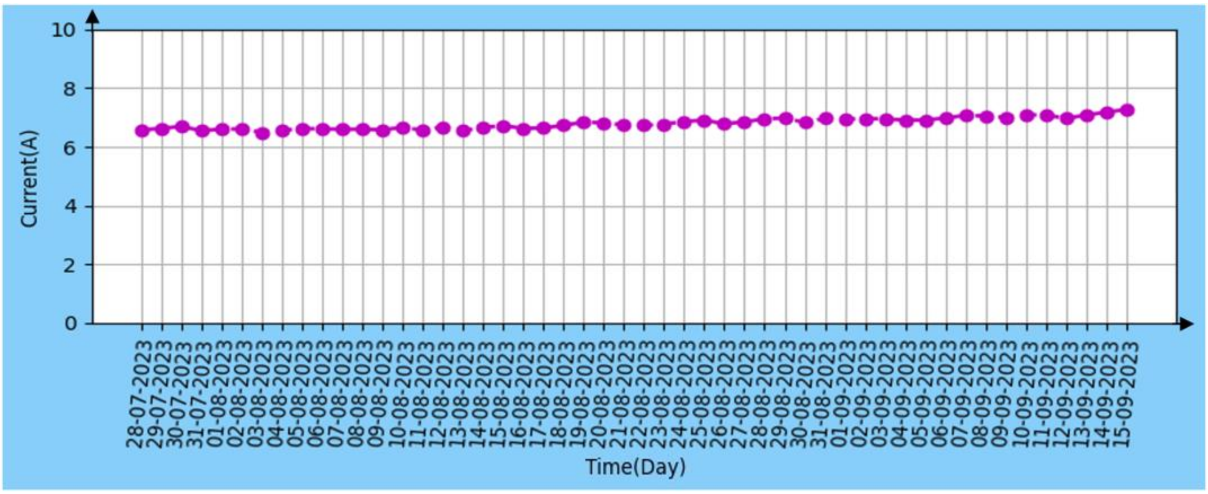

Figure 4.3. Current-time graph of the irrigation pump

For the functionality and safety of the irrigation pump, it is important to monitor and control the temperature and humidity levels. This ensures long life of the irrigation pump and safe distribution of electrical energy. Figure 4.4 shows the time dependent variation of the irrigation pump temperature. As the vibration value increases, the temperature of the irrigation pump increases.

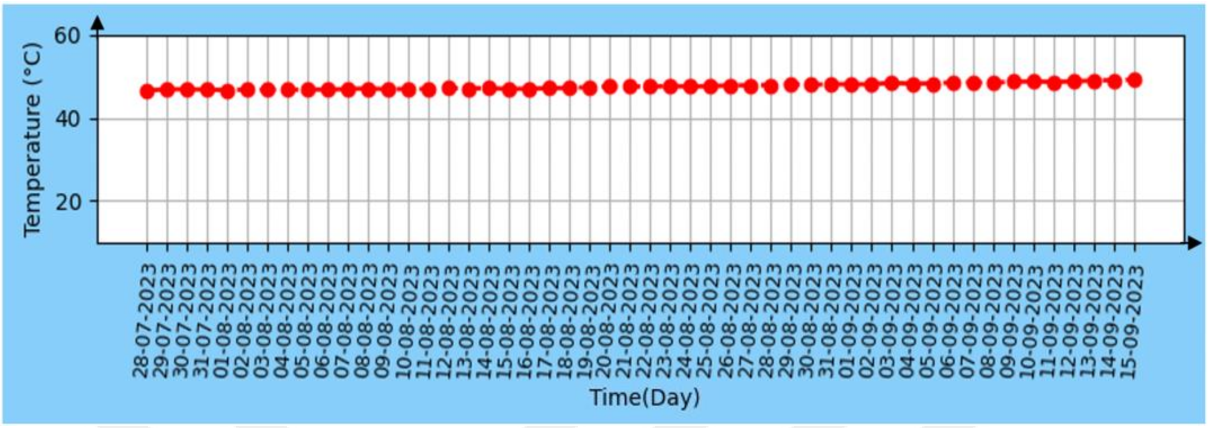

Figure 4.4. Temperature-time graph of the irrigation pump

As shown in Figure 4.5, this function is used to calculate after how many days the received data reaches the threshold value. The calculated value is then added to the current date to estimate the maintenance date of the irrigation pump. If the threshold value to be calculated has been exceeded before, it is also possible to find out how many days ago this threshold value was exceeded. Figure 4.6 shows this feature.

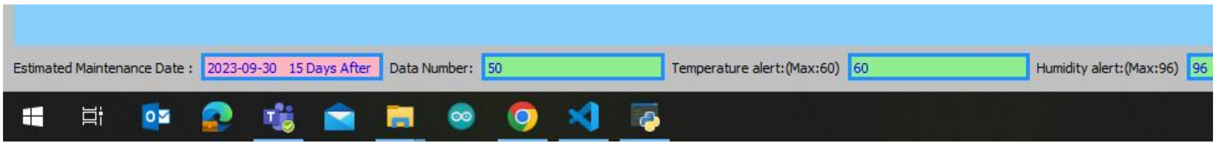

Figure 4.5. The vibration was set to 10 mm/s, and the estimated maintenance time is 15 days later.

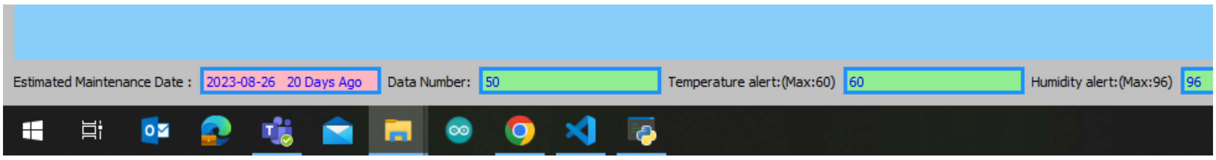

Figure 4.6. The vibration was set to 5 mm/s, and the estimated maintenance time was due 20 days ago.

4.2. Correlation of the relationship between the data obtained from the irrigation pump

Some of the correlation methods are as follows: Spearman Correlation Coefficient, Kendall Tau Correlation Coefficient, Pearson Correlation Coefficient.

The Pearson correlation coefficient was chosen because it determines whether the relationship between variables is linear or not, and the strength and direction of the relationship. The correlation coefficient takes a value between -1 and 1. A positive correlation approaching 1 indicates a strong positive relationship between two attributes. That is, when the value of one attribute increases, the other also tends to increase. A negative correlation approaching -1 indicates a strong negative relationship between the two attributes. This means that when the value of one attribute increases, the other tends to decrease. A correlation close to 0 indicates that there is no relationship or a very weak relationship between the two attributes. In Figure 4.7 the correlation of the relationship between the data is colored.

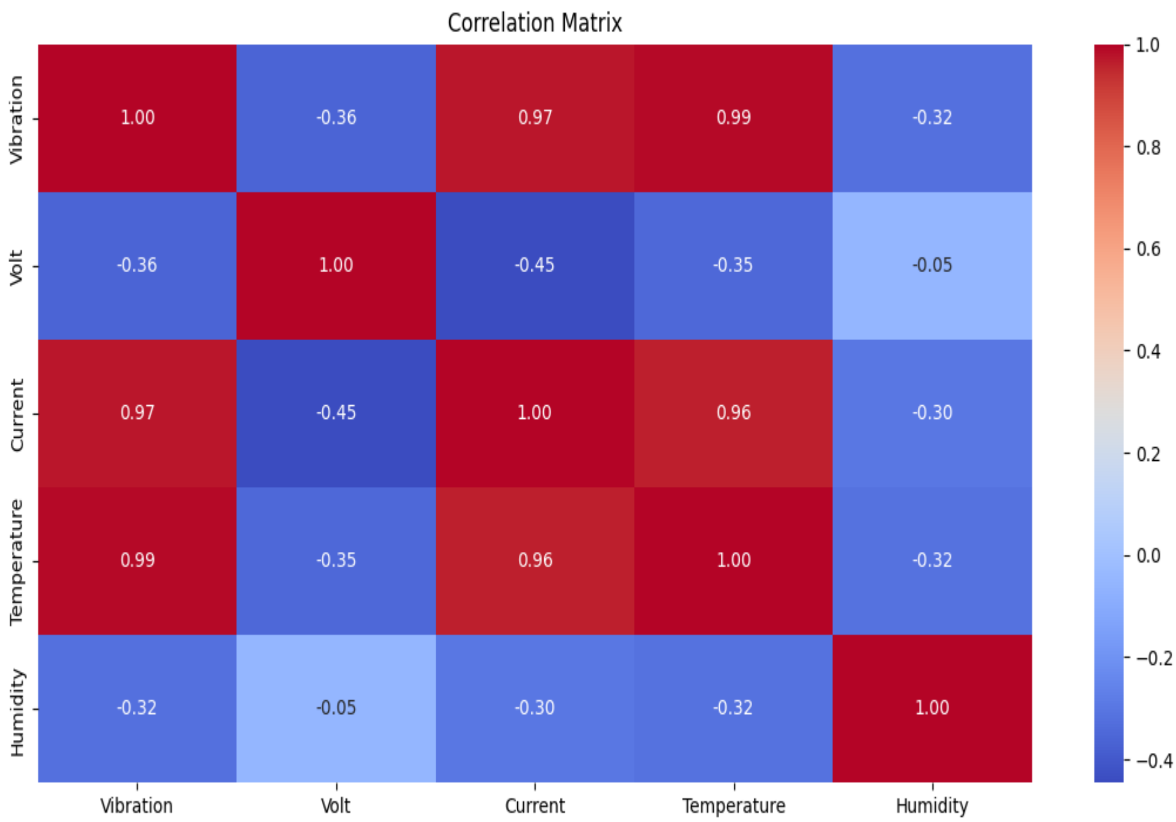

Figure 4.7. Correlation matrix between the variables.

There is a high positive correlation between "current" and "vibration" with a value of 0.97. This positive correlation indicates a strong linear relationship between current and vibration. This high correlation indicates that current levels tend to increase with increasing vibration. That is, generally speaking, as vibration increases, current levels also increase.

There is a correlation of -0.36 between "volt" and "vibration". There is a weak inverse relationship between volt and vibration. This negative correlation usually indicates that voltage levels tend to decrease with increasing vibration. However, since the correlation coefficient is around -0.3, this relationship can be considered very weak. Therefore, it is difficult to say that there is a definite causal relationship between "volts" and "vibration". However, this weak relationship could mean that under certain conditions voltage could influence vibration levels and vice versa. Therefore, it is important to conduct further analysis and consider other variables.

There is a high positive correlation of 0.99 between "temperature" and "vibration". This positive correlation indicates a strong linear relationship between temperature and vibration. This high correlation indicates that temperature levels tend to increase with increasing vibration. So, generally, as vibration increases, temperature levels also increase. However, since the correlation coefficient is around 0.99, this relationship is quite strong and it is safe to say that there is a linear relationship between temperature and vibration.

There is a correlation of -0.32 between "humidity" and "vibration". There is a weak inverse relationship between humidity and vibration. That is, it can be seen that humidity tends to decrease as vibration increases. This indicates that when vibration increases, generally the humidity level tends to decrease. However, since the correlation coefficient is around -0.3, this relationship can be considered very weak. Therefore, it is difficult to say that there is a definite causal relationship between humidity and vibration.

4.3. Application of the trained models to the test dataset and error values.

We divided the dataset into training dataset and test dataset. We trained linear regression, polynomial regression, random forest regression and support vector regression models with the training dataset. The red color indicates our test data. Purple color represents linear regression model, orange color represents polynomial regression model, blue color represents support vector regression model and green color represents random forest regression model. For our models, inputs are current, voltage, humidity, temperature and volts, output is vibration.

Figure 4.8 vibration-test dataset graph is given. In the graph, we see that our red color test data falls under the green random forest model.

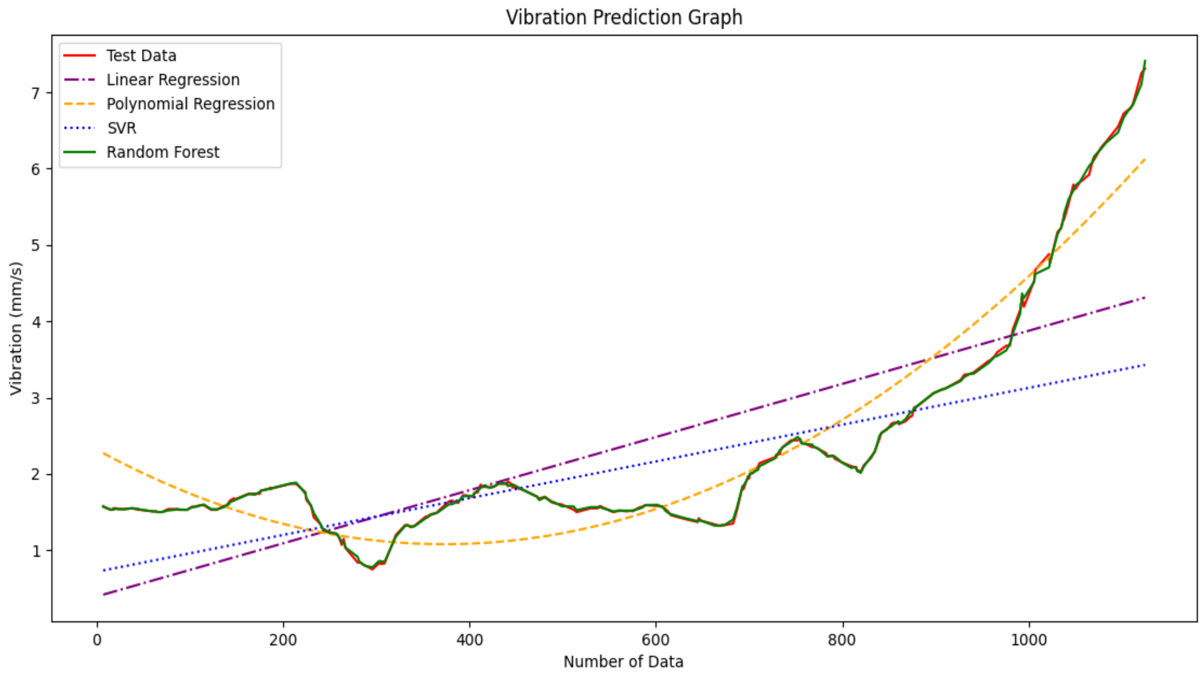

Figure 4.8. Application of models trained with training data on vibration test data.

Figure 4.9 shows the current and test data set graph. It can be seen that the linear regression and SVR exhibit similar behavior. They are not as close to the test data as the other models. The random forest model has a significantly better agreement with the test data.

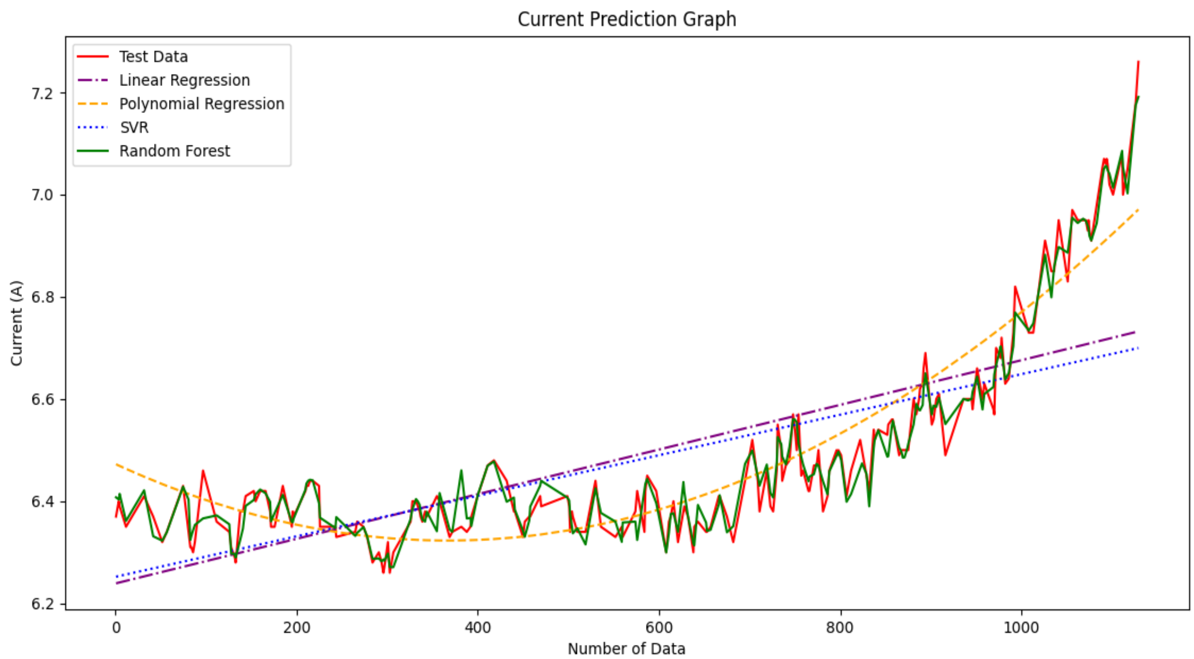

Figure 4.9. Application of models trained with training data on current test data.

Figure 4.10 shows the plot of the humidity and test data set. It can be seen that linear regression, polynomial regression and SVR exhibit similar behavior. The random forest model has a significantly better fit with the test data.

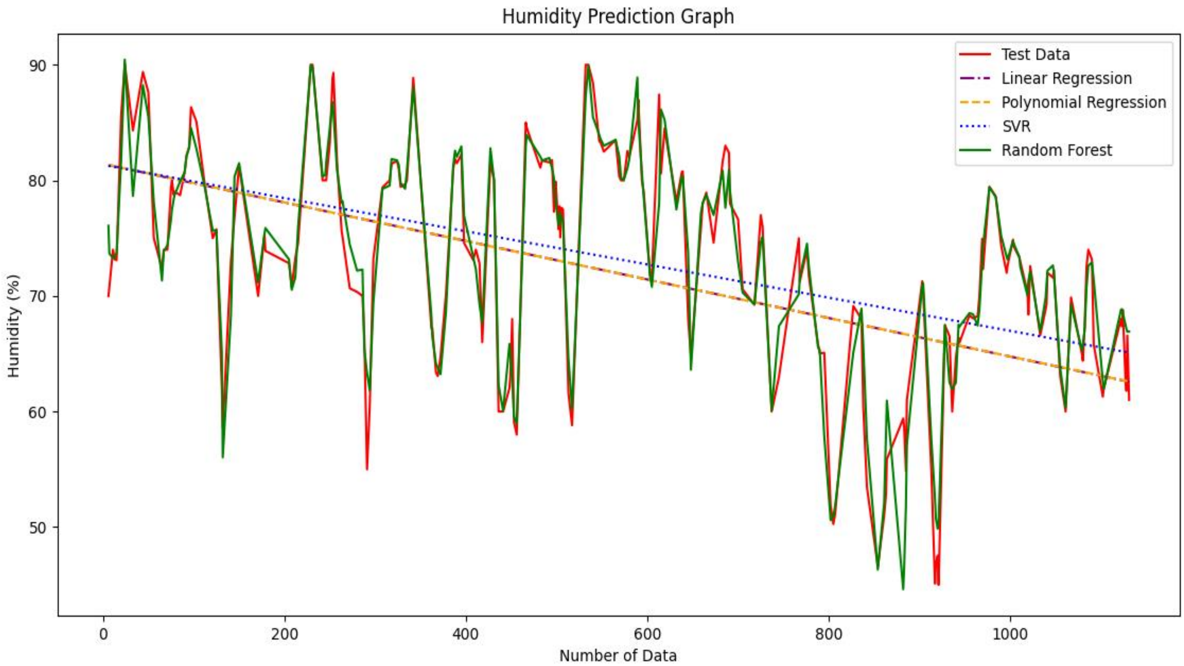

Figure 4.10. Application of models trained with training data on humidity test data.

Figure 4.11 shows the plot of temperature and test data set. It can be seen that the linear regression and SVR exhibit similar behavior. The polynomial regression model has a better fit than the linear regression and SVR model. The random forest model has a significantly better fit with the test data.

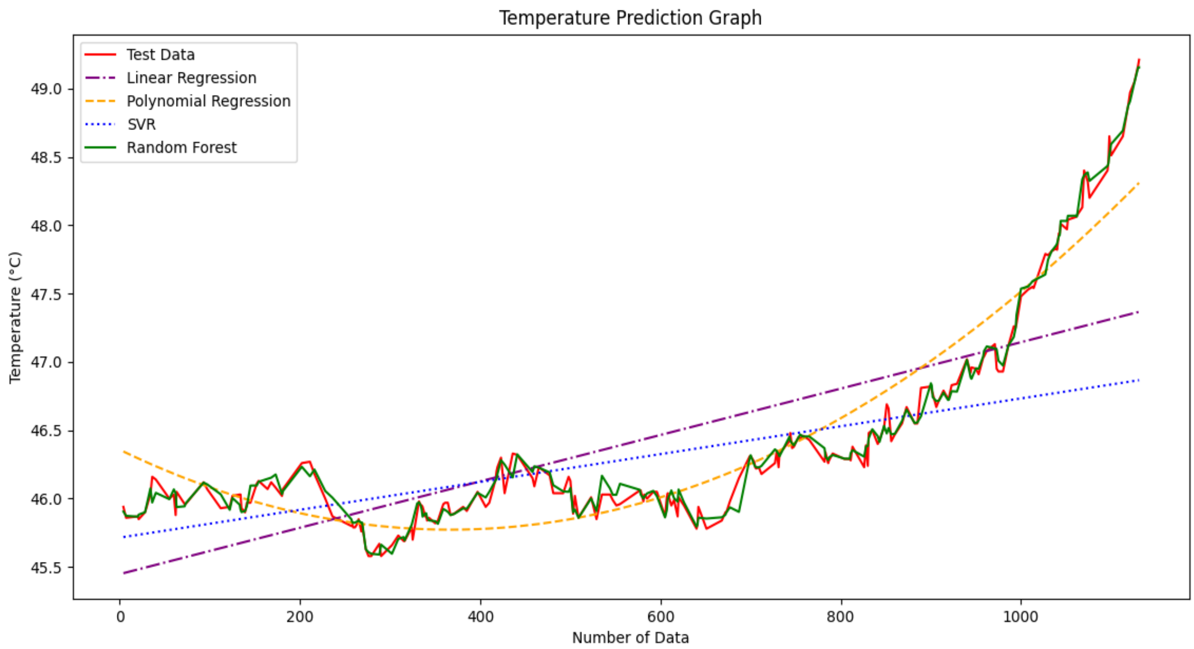

Figure 4.11. Application of models trained with training data on temperature test data.

Figure 4.12 shows the plot of volt and test data set. The models seem to have a general difficulty. It can be seen that the linear regression and SVR exhibit similar behavior and are worse with the test data than the other regression models. Polynomial regression model has a better fit than linear regression and SVR model. The best result was obtained with the random forest model. From Figure 4.8 to Figure 4.12 demonstrate how well the models fit the test data.

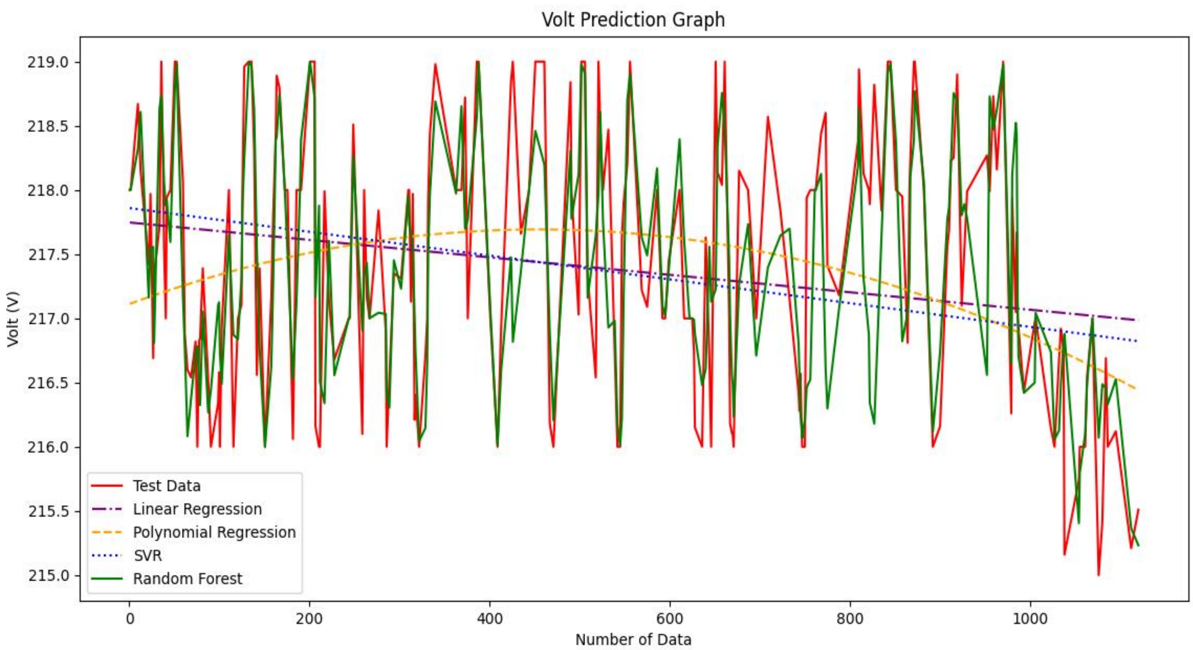

Figure 4.12. Application of models trained with training data on volt test data.

Table 4.1 is a table of results showing the error values of the models. If R-square is 1, a perfect linear relationship exists between x and y. That is, 100% of the variation in y is explained by the variation in x.(Kumari & Yadav, 2018) It is seen that as the R-square value approaches 1, the prediction probability becomes stronger. The model with the closest R-squared value to 1 in the whole table belongs to the random forest regression model. The model with the smallest MSE value in the whole table belongs to the random forest regression model. When these graphs are analyzed, it is observed that the random forest regression model fits the test data better.

Table 4.1. The names and error values of the models used for each feature.

Model	MAE	MSE	MAPE	RMSE	R-square
Current					
Linear Regression	0.105	0.018	1.600	0.136	0.571
Polynomial (degree=2)	0.061	0.006	0.927	0.075	0.870
Random Forest Regression	**0.022**	**0.001**	**0.331**	**0.029**	**0.981**
Support Vector Regression	0.101	0.019	1.531	0.138	0.557
Humidity					
Linear Regression	7.269	76.860	10.558	8.767	0.182
Polynomial (degree=2)	7.269	76.871	10.558	8.768	0.181
Random Forest Regression	**1.709**	**7.204**	**2.538**	**2.684**	**0.923**
Support Vector Regression	6.972	75.089	10.317	8.665	0.200
Temperature					
Linear Regression	0.395	0.253	0.843	0.503	0.592
Polynomial (degree=2)	0.203	0.067	0.433	0.259	0.892
Random Forest Regression	**0.045**	**0.004**	**0.097**	**0.065**	**0.993**
Support Vector Regression	0.376	0.344	0.795	0.587	0.446
Vibration					
Linear Regression	0.752	0.861	36.727	0.928	0.569
Polynomial (degree=2)	0.392	0.219	19.475	0.468	0.890
Random Forest Regression	**0.020**	**0.001**	**0.937**	**0.031**	**0.999**
Support Vector Regression	0.658	1.010	28.018	1.005	0.495
Volt					
Linear Regression	0.870	1.017	0.400	1.008	0.014
Polynomial (degree=2)	0.852	0.975	0.392	0.988	0.054
Random Forest Regression	**0.466**	**0.425**	**0.215**	**0.652**	**0.587**
Support Vector Regression	0.873	1.038	0.401	1.019	-0.007

All features and trained regression models and their relationships with the test samples are visualized in Figure 4.13 to 4.18.

Figure 4.13 shows the actual values of the features in the test dataset and the trained regressions. The more data sets on the black diagonal in the graphs, the better the performance of the trained model.

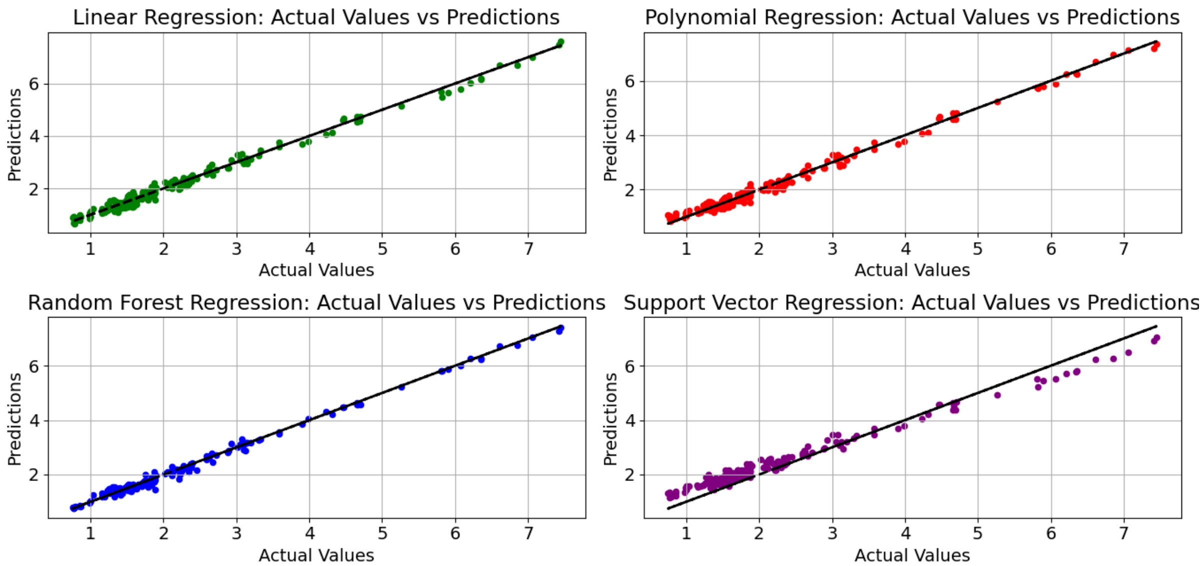

Figure 4.13. Actual values-predictions graph of the models.

A comparison was made between the actual values of the features in the test dataset and the trained linear regression model and the results were analyzed. Figure 4.14 shows that the linear regression model is a better method than the SVR model.

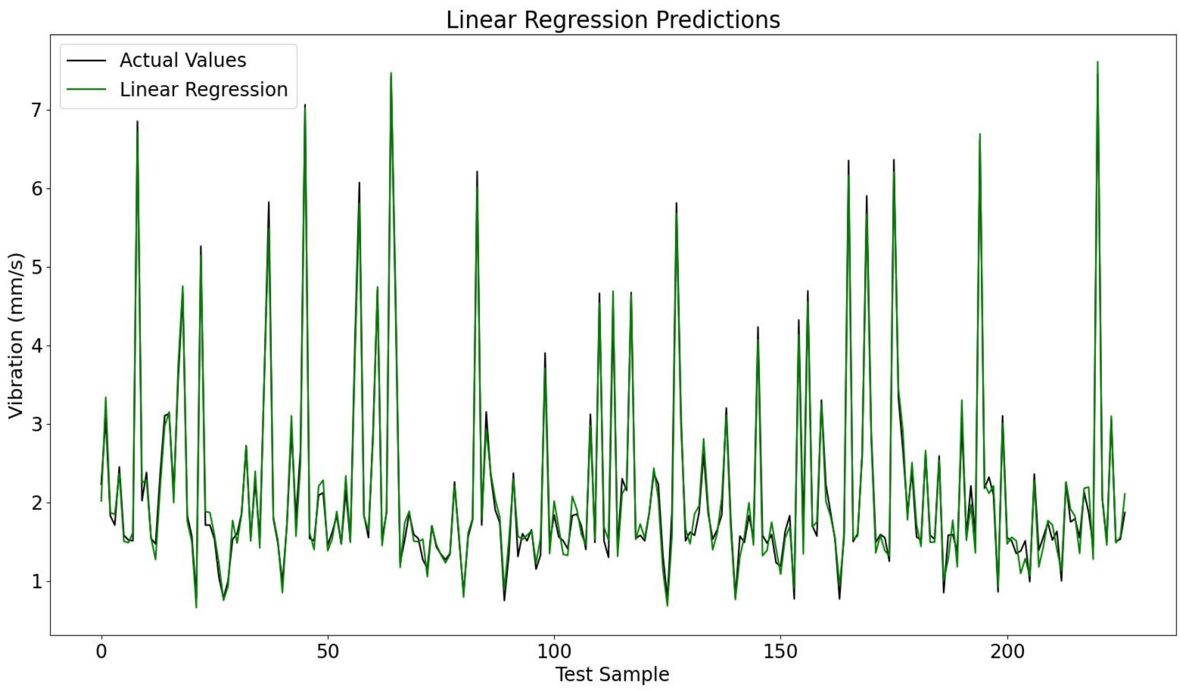

Figure 4.14. Actual values of the test samples with all features and the trained linear regression model.

A comparison was made between the actual values of the features in the test dataset and the trained polynomial regression model and the results were analyzed. Figure 4.15 shows that polynomial regression is worse than the RFR model and better than the other models.

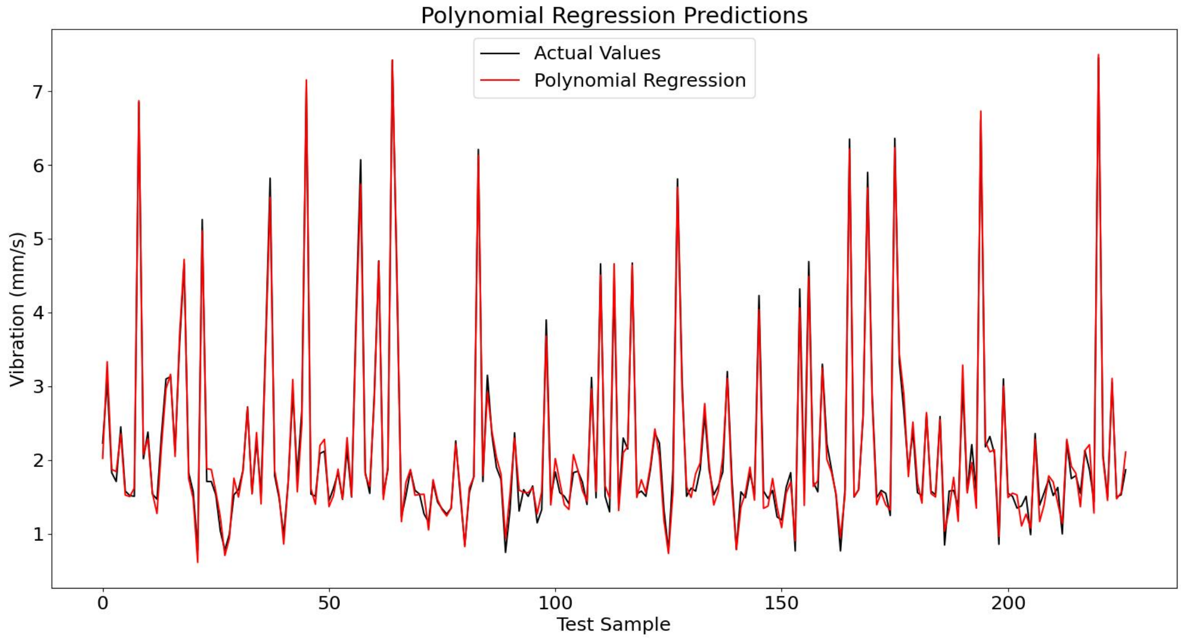

Figure 4.15. Actual values of the test samples with all features and the trained polynomial regression model.

A comparison was made between the actual values of the features in the test dataset and the trained RFR model and the results were analyzed. Figure 4.16 is the visual that helps to assess how well the model fits the real data.

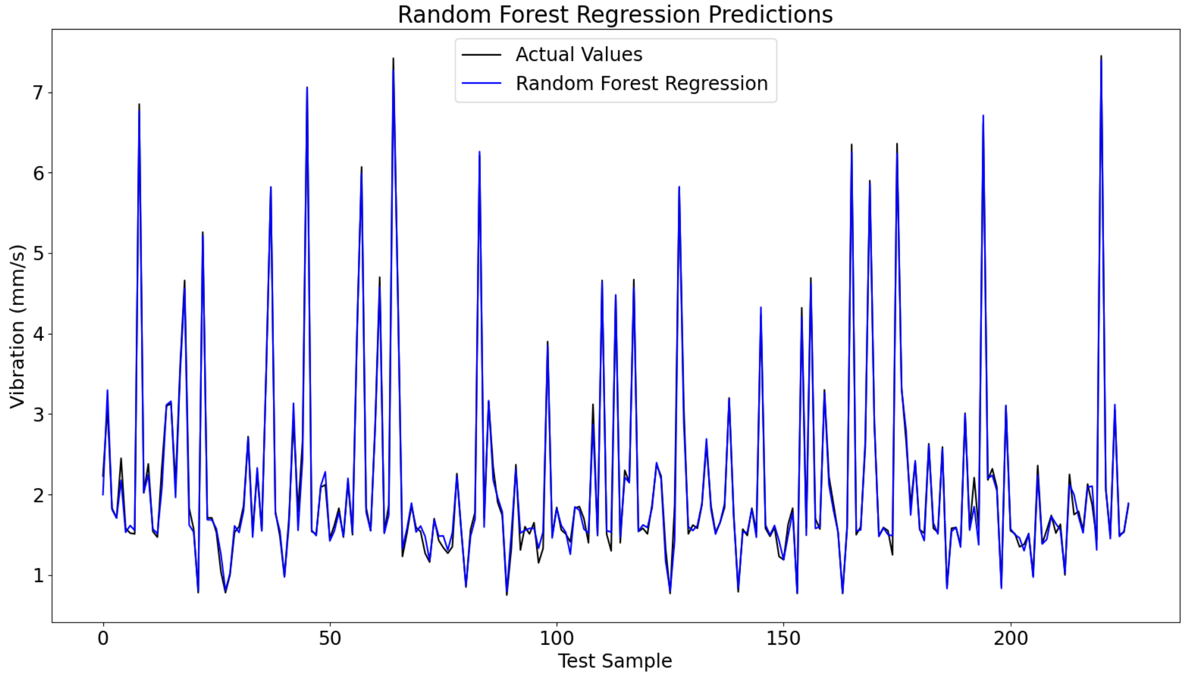

Figure 4.16. Actual values of the test samples with all features and the trained random forest regression model.

A comparison was made between the actual values of the features in the test dataset and the trained SVR model and the results were analyzed. Figure 4.17 shows that the SVR model is not a good method compared to other models.

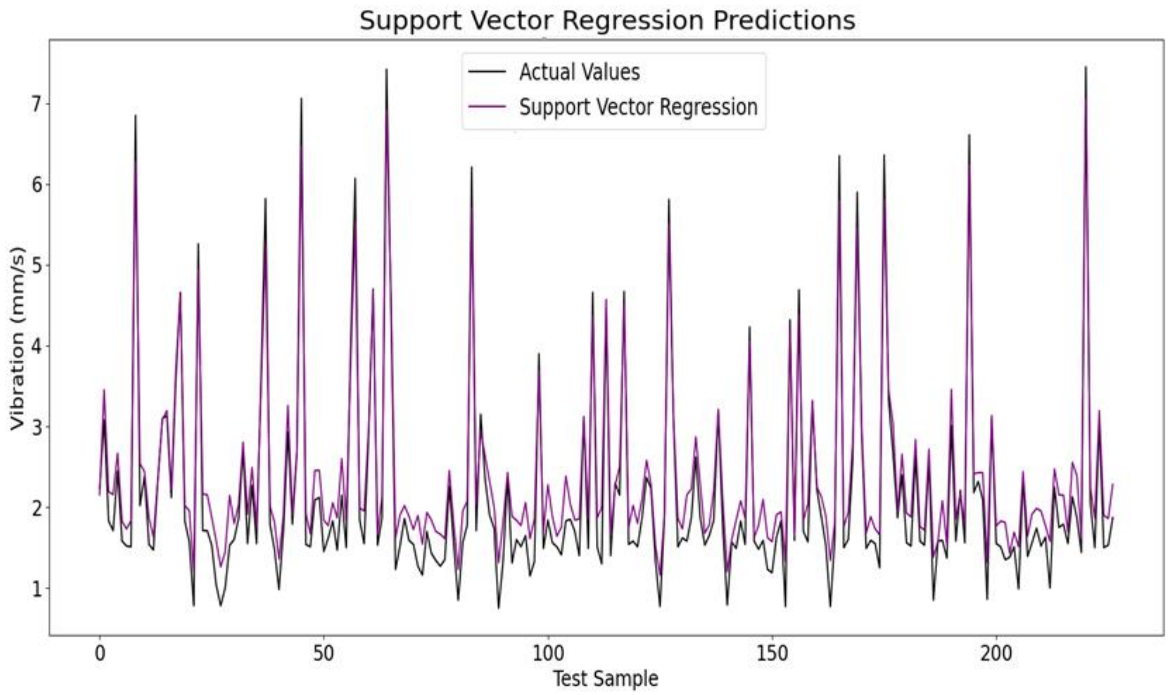

Figure 4.17. Actual values of the test samples with all features and the trained support vector regression model.

Table 4.2. shows the regression models and their associated error values. For the model to be suitable for the test data, the MSE value should be close to 0. The model with the best MSE value is the random forest model.

Table 4.2. The names and error values of the models used for all features.

Model	MAE	MSE	RMSE	R-square
Linear Regression	0.0161	0.00037	0.0193	0.9886
Polynomial Regression (degree=2)	0.0154	0.00035	0.0186	0.9894
Random Forest Regression	**0.0098**	**0.00019**	**0.0139**	**0.9941**
Support Vector Regression	0.0397	0.0020	0.0452	0.9376

4.4. Optimizing the parameters of regression models with genetic algorithm

By searching for combinations of the parameters of the models, the genetic algorithm can find the optimal parameter values to maximize the performance of the model. In this way, the prediction performance of the model can be improved and a better fit to the data can be achieved.(Wu et al., 2009)

To optimize the polynomial regression model, the optimal degree value is found with the genetic algorithm. The higher the degree of the polynomial, the greater the flexibility of the model. Higher degree polynomials can provide a closer fit to the data, but can lead to overfitting. This means that the model overfits the training data and cannot generalize. Therefore, it is important to find the optimal degree in polynomial regression.

Genetic algorithm can be used to find the optimal polynomial degree of the polynomial regression model. In this process, different polynomial degrees are tested and the performance of the model is evaluated using an objective function. The polynomial degree that provides the best performance is considered the optimal degree.

Figure 4.18 shows the iteration and objective function improvement of the genetic algorithm when searching for the optimal degree for the polynomial model.

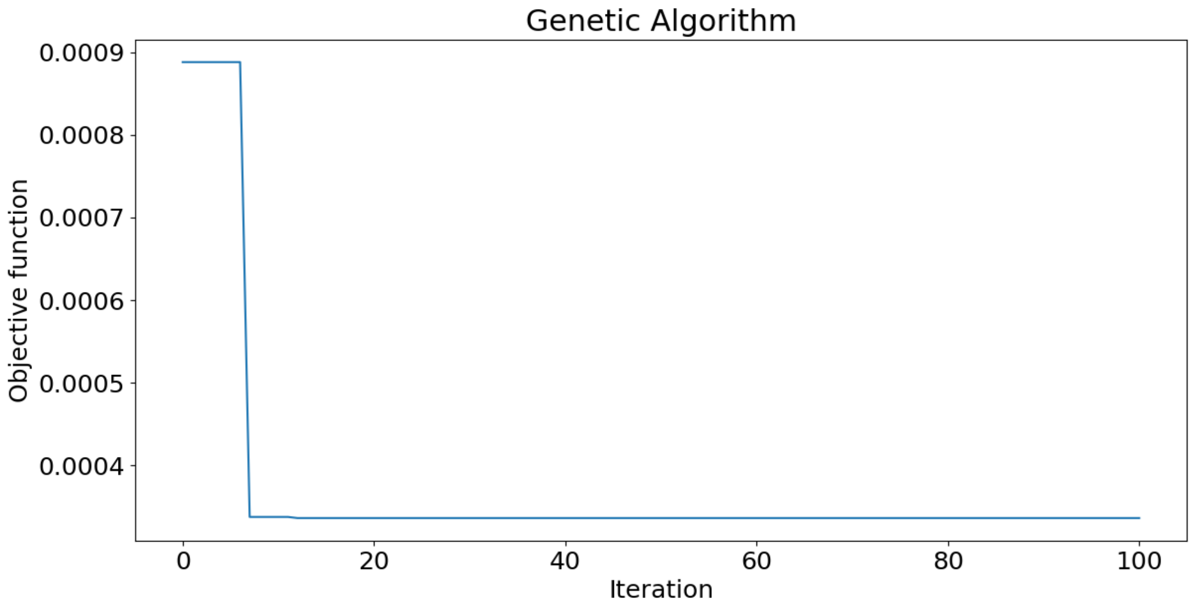

Figure 4.18. Finding the optimal degree values for polynomial regression model by applying genetic algorithm.

Figure 4.19 shows the application of the new model created with the help of the optimum degree parameter found by the genetic algorithm for polynomial regression to the test data.

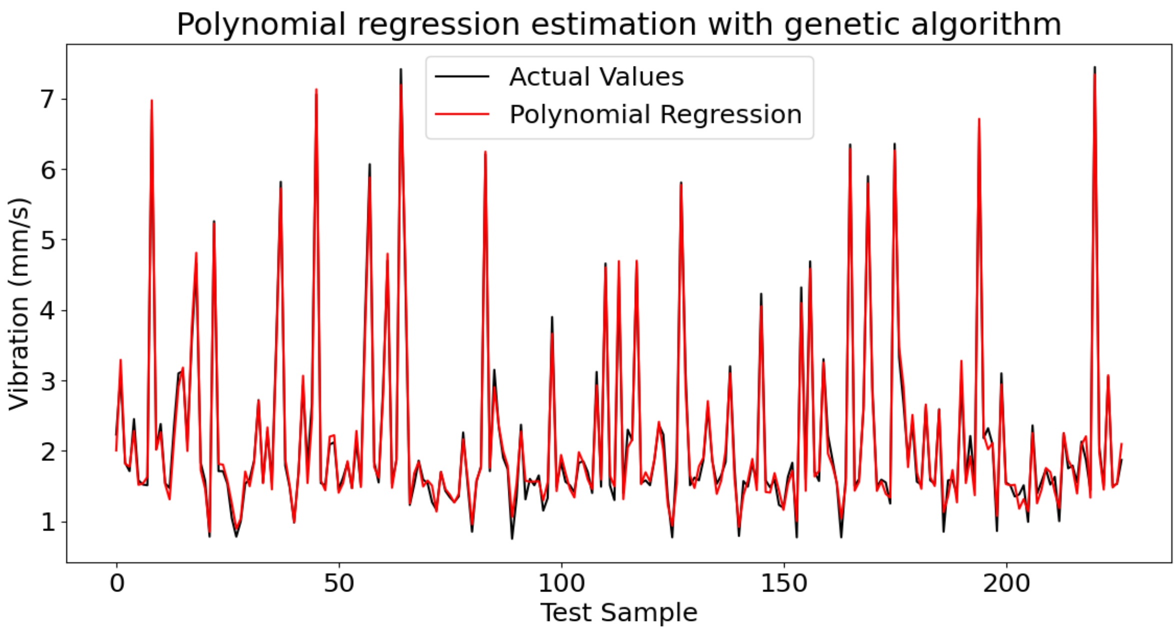

Figure 4.19. Polynomial regression estimate after parameters improved by GA

As shown in Table 4.3, the optimized polynomial degree of the model with the genetic algorithm is 3 and the MAE value=0.0149, MSE value=0.00034, RMSE value=0.0183 and R-square value=0.9897. The genetic algorithm is an effective method to ensure that the model best fits the data by searching for the polynomial degree and maximizing the performance of the model. In this way, the accuracy of the polynomial regression model can be improved and more reliable predictions can be obtained.

Table 4.3. Optimal polynomial degree and error values found for the polynomial regression model by applying the genetic algorithm

Model	MAE	MSE	RMSE	R-square
Polynomial Regression (Optimized Polynomial Degree=3)	0.0149	0.00034	0.0183	0.9897

Figure 4.20 shows the iteration and objective function evolution of the genetic algorithm while searching for the optimal value of C and gamma for the SVR model.

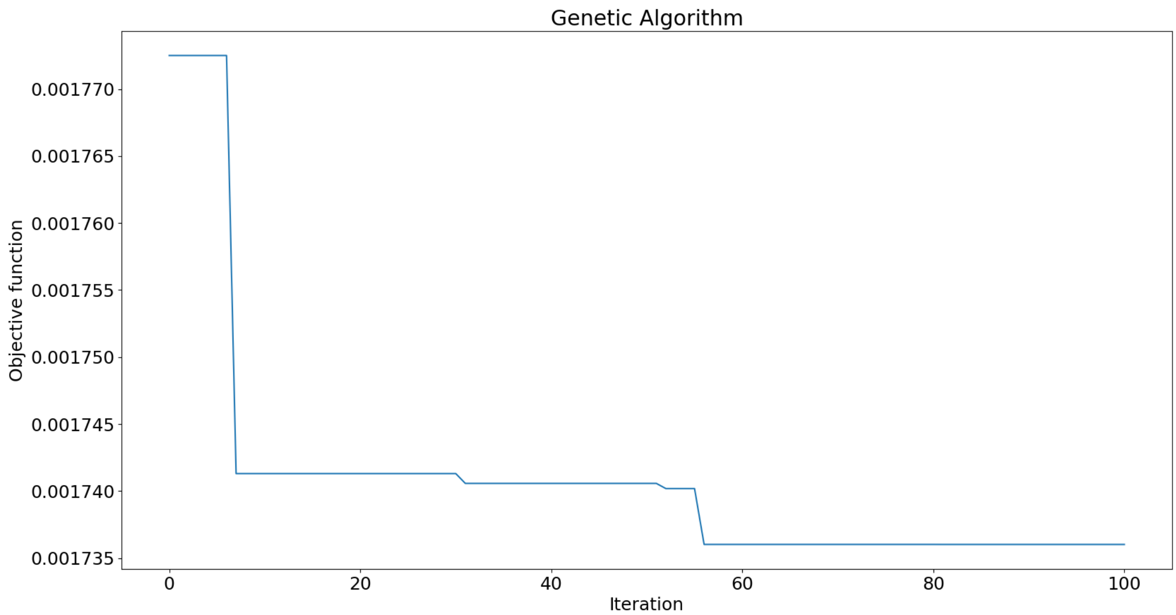

Figure 4.20. Finding the optimal C and Gamma values for SVR model by applying genetic algorithm.

Figure 4.21 shows the application of the new model created with the help of the optimum C and gamma parameters found by the genetic algorithm for SVR to the test data.

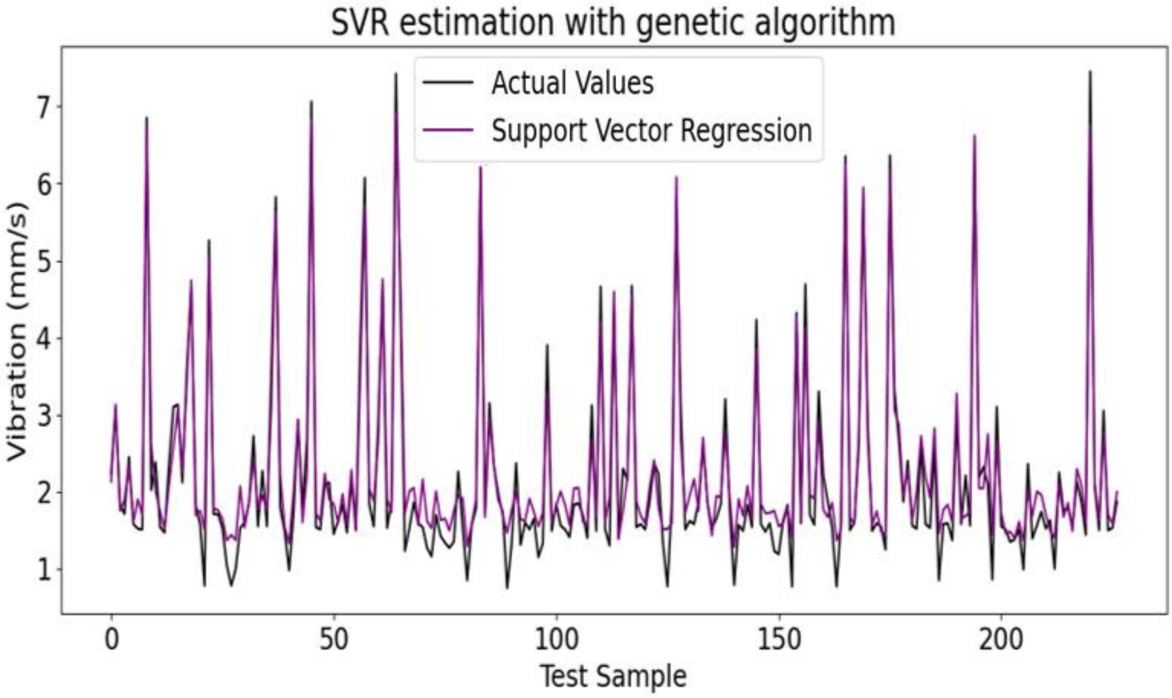

Figure 4.21. SVR estimate after parameters improved by GA

As shown in Table 4.4, the optimized parameters of the model with the genetic algorithm are C=60.75 and gamma=1.504 and the new values after the optimized parameters are MAE =0.0336, MSE =0.00174, RMSE =0.0417 and R-square =0.9471.

Table 4.4. Optimal C, Gamma values and error values found for the SVR model by applying the genetic algorithm

Model	MAE	MSE	RMSE	R-square
SVR (Optimized C=60.75, Optimized Gamma=1.504)	0.0336	0.00174	0.0417	0.9471

The accuracy of the SVR model is super-sensitive to the parameters C and gamma.(Quan et al., 2022) To optimize the support vector regression model, the optimal C and gamma values are found with the genetic algorithm. The C value controls how much the regression is regularized. Higher C values make the model fit the training data more tightly, but can lead to overfitting. The gamma value determines the width of the RBF kernel. Smaller gamma values result in a wider RBF kernel, while larger gamma values result in a narrower RBF kernel.

Figure 4.22 shows the iteration and objective function evolution of the genetic algorithm while searching for the optimum value of n estimators, max depth, min samples split parameters for the RFR model.

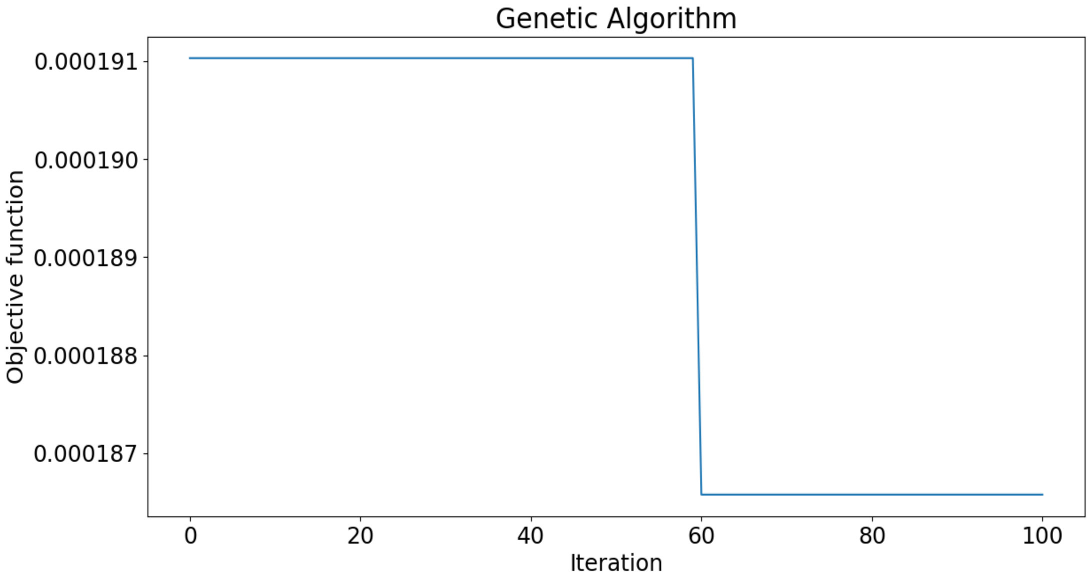

Figure 4.22. Finding the optimal n estimators, max depth and min samples split values for RFR model by applying genetic algorithm.

Figure 4.23 shows the application of the new model created with the help of the optimum n estimators, max depth and min samples split parameters found by the genetic algorithm for RFR to the test data.

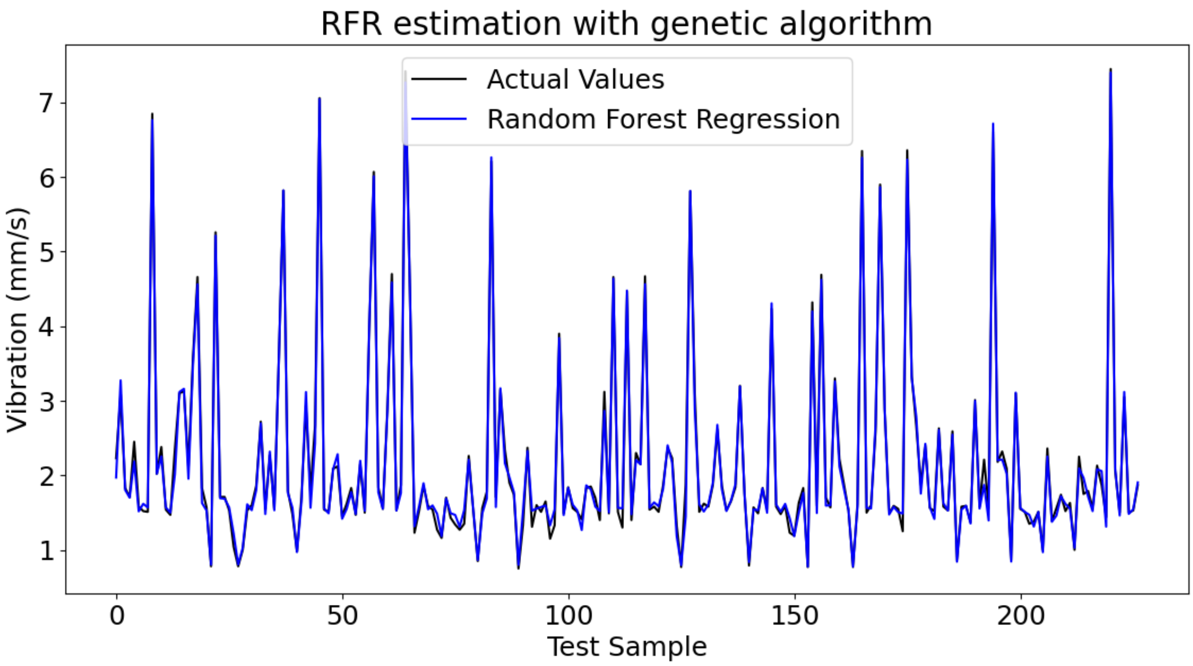

Figure 4.23. RFR estimate after parameters improved by GA

To optimize the random forest model, the genetic algorithm is used to find the optimal n estimators, max depth and min sample split values. Each of these parameters should be within a certain range, as excessively large or excessively small values can negatively affect the performance of the model.

The n estimators determines the number of trees in a random forest model. More trees can provide more diversity and a stronger model. However, using too many trees can increase the computation time of the model and lead to overfitting. This is because too many trees can create a model that over-fits the training data and does not generalize.

The max depth determines the maximum depth of each tree. The deeper the tree, the greater the model's ability to learn more complex relationships. However, trees that are too deep can lead to overfitting. This is because trees can fully memorize the training data and cannot generalize well with new data. Therefore, controlling the maximum depth is important to increase the model's ability to generalize.

The min samples split determines the minimum number of samples required before splitting an internal node. This parameter can increase the model's resistance to overfitting. This is because a larger value of min samples split can create less branching and a more general model. However, choosing a value that is too large can reduce the fit of the model to the training data and negatively affect its ability to generalize.

As shown in Table 4.5, the parameters of the model optimized by the genetic algorithm are n estimators value=194, max depth value=56, min samples split values=2 and the new error values after optimized parameters are : MAE =0.0097, MSE =0.00018, RMSE =0.0137 and R-square =0.9942.

Table 4.5. Optimal n estimators, max depth, min samples split values and error values found for RFR model by applying genetic algorithm

Model	MAE	MSE	RMSE	R-square
Random Forest (Optimized n estimators=194, Optimized max depth=56, Optimized min samples split=2)	0.0097	0.00018	0.0137	0.9942

Figure 4.24 shows the GA-improved results of all models. When all models are analyzed, it is observed that the SVR model has lower precision. The RFR model shows us the best results among these three models. This is also confirmed by the error values.

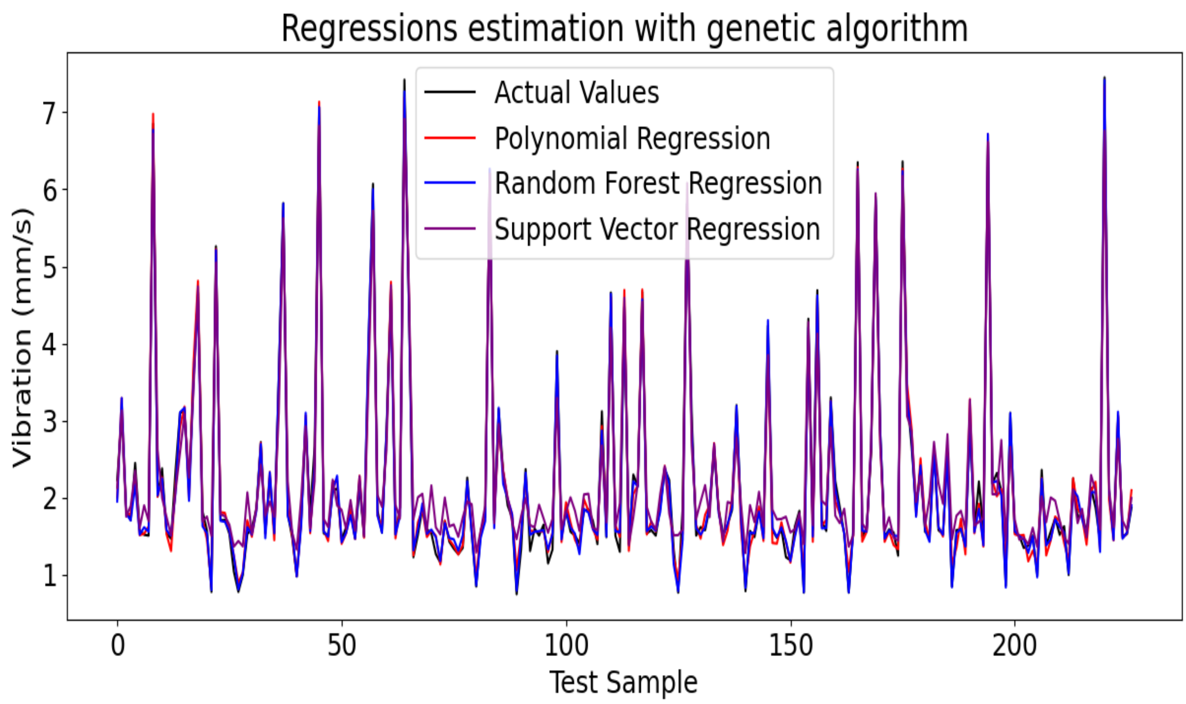

Figure 4.24. All models regression estimates after parameters improved by GA

The error values of all the models before applying the genetic algorithm and after applying the genetic algorithm are shown in Table 4.6.

Table 4.6. Error values of the models before and after the application of the genetic algorithm.

Model	MAE	MAE (GA)	MSE	MSE (GA)	RMSE	RMSE (GA)	R-square	R-square (GA)
Linear Regression	0.0161	-	0.00037	-	0.0193	-	0.9886	-
Polynomial Regression	0.0154	0.0149	0.00035	0.00034	0.0186	0.0183	0.9894	0.9897
Random Forest Regression	**0.0098**	**0.0097**	**0.00019**	**0.00018**	**0.0139**	**0.0137**	**0.9941**	**0.9942**
Support Vector Regression	0.0397	0.0336	0.0020	0.00174	0.0452	0.0417	0.9376	0.9471

After normalization of the data obtained from the irrigation pump, current, temperature, humidity, voltage and vibration data were randomly divided into 80% training and 20% test data sets. Predictive maintenance of the system consists of a three-stage process in the field of machine learning. In the first stage, the correlation matrix was created. There is a positive correlation between vibration, current and temperature, and a negative and weak correlation with voltage. There is a negligible correlation with humidity, but we included all parameters in the models. In the second stage, the linear regression, polynomial regression, SVR and RFR models were evaluated based on MSE value. In this evaluation, MSE values of 0.00037 for linear regression, 0.00035 for polynomial regression, 0.00019 for RFR and 0.0020 for SVR were obtained and the best results were observed in the RFR model. In the third stage, genetic algorithm was used to determine the optimum parameters of polynomial regression, SVR and RFR models and MSE values of 0.00034 for polynomial regression, 0.00018 for RFR and 0.00174 for SVR were found. The best performance was observed in the RFR model. RFR has the lowest MSE error value and also the largest R-squared value. SVR has the highest MSE error value. As can be seen from the graphs, it does not fit the test data as well as other models. The models after the genetic algorithm show improved results.

5. CONCLUSIONS

With the development of IoT systems and the subsequent incorporation of artificial intelligence, solutions have been provided to problems in many areas. The use of IoT systems and artificial intelligence plays an important role in many problems encountered in daily life, as understood by academic studies. It is still in its infancy in rural irrigation pump. In the coming years, companies that manufacture and maintain irrigation pump will be able to control the maintenance and health status of their irrigation pump from a single point. In this paper, we aimed to create an application with a simple user interface that predicts the health status of irrigation pumps in rural areas with an IoT system and maintenance date using artificial intelligence, and provides the necessary data to develop this prototype into a product. The application created for this thesis is unique in that it includes the Python programming language, the Qt user interface creation tool, and an artificial intelligence algorithm. The thesis consists of a two-stage process in the field of machine learning. In the first stage, four different models were evaluated. In the second stage, the optimal parameters of polynomial regression, SVR and RFR models were determined using genetic algorithm. This study has reached the following results:

- From the correlation matrix, there is a positive correlation between current, temperature and vibration, while there is a weak negative correlation between voltage and vibration. No relationship was found between humidity and vibration.

- Analyzing all the graphs so far, it can be seen that as the motor vibration increases, the motor temperature also increases. Depending on the vibration, the current value increased and the voltage decreased accordingly. The correlation matrix supports this situation.

- Genetic algorithm was used to determine the optimum parameters of polynomial regression, SVR and RFR models and MSE values of 0.00034 for polynomial regression, 0.00174 for SVR and 0.00018 for RFR were found.

- Table 4.6 shows that the random forest regression has the lowest MSE error value. Among these models, random forest regression was found to be the best among the models, and the optimal parameter values were found using the genetic algorithm to further improve the model.

- Four models were applied to the data set: linear regression, polynomial regression, support vector regression, and random forest regression. The genetic algorithm found the optimal polynomial degree to be 3. For the SVR model, the genetic algorithm found the optimal C=60.75 and the optimal Gamma=1.504. The genetic algorithm found optimal n estimators=194, optimal max depth=56, optimal min samples split=2 for the RFR model.

The developed application stands out because the graphics are more detailed, can alert the user, and include artificial intelligence algorithms compared to the other academic studies. In this thesis, an innovative application and embedded system structure has been created for companies engaged in irrigation pump production and maintenance. It is hoped that the application will become an automation product in the future.

6. RECOMMENDATIONS

The system created here has pioneered how the maintenance, health and performance of a rural irrigation pump can be achieved. The created system can be integrated into a company that manufactures and maintains irrigation pumps by renting a server. Thus, a good innovation gain can be achieved for the company. It plays an important role in reducing the company's maintenance costs and attracting more customers. The workload is reduced by providing the possibility to control the irrigation pump in the rural area from a single control point. The system is planned to be supported with more advanced machine learning methods in future studies. The system created can be a pioneer for companies with innovation goals. By using the vibration feature of the created system, maintenance of rails in rail transportation, maintenance of blades in wind turbines and highway maintenance after installation on vehicles can be predicted.

REFERENCES

Amruthnath, N., & Gupta, T. (2018). A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance. *2018 5th International Conference on Industrial Engineering and Applications, ICIEA 2018*, 355–361. https://doi.org/10.1109/IEA.2018.8387124

Asadullah, M., & Ullah, K. (2017). Smart home automation system using Bluetooth technology. *ICIEECT 2017 - International Conference on Innovations in Electrical Engineering and Computational Technologies 2017, Proceedings*. https://doi.org/10.1109/ICIEECT.2017.7916544

Barot, V., Kapadia, V., & Pandya, S. (2020). QoS enabled IoT based low cost air quality monitoring system with power consumption optimization. *Cybernetics and Information Technologies*, *20*(2). https://doi.org/10.2478/cait-2020-0021

Bhatt, A., & Patoliya, J. (2016). Cost effective digitization of home appliances for home automation with low-power WiFi devices. *Proceeding of IEEE - 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics, IEEE - AEEICB 2016*. https://doi.org/10.1109/AEEICB.2016.7538368

Campo, G. Del, Calatrava, S., Canada, G., Olloqui, J., Martinez, R., & Santamaria, A. (2018). IoT Solution for energy optimization in industry 4.0: Issues of a real-life implementation. *2018 Global Internet of Things Summit, GIoTS 2018*. https://doi.org/10.1109/GIOTS.2018.8534537

Cerrada, M., Zurita, G., Cabrera, D., Sánchez, R. V., Artés, M., & Li, C. (2016). Fault diagnosis in spur gears based on genetic algorithm and random forest. *Mechanical Systems and Signal Processing*, *70–71*, 87–103. https://doi.org/10.1016/J.YMSSP.2015.08.030

Che Soh, Z. H., Hamzah, I. H., Che Abdullah, S. A., Shafie, M. A., Sulaiman, S. N., & Daud, K. (2019). Energy consumption monitoring and alert system via IoT. *Proceedings - 2019 International Conference on Future Internet of Things and Cloud, FiCloud 2019*. https://doi.org/10.1109/FiCloud.2019.00044

Deny, J., Narasimha, A. B., Vardhan Reddy, R. G., & Sathish, S. (2021). Electricity monitoring and auto bill generation using IOT. *2021 3rd International Conference on Signal Processing and Communication, ICPSC 2021*. https://doi.org/10.1109/ICSPC51351.2021.9451738

DHT22 Humidity and Temperature Sensor. (2024). https://www.circuitgeeks.com/arduino-dht11-and-dht22-sensor-tutorial/

Hareendran, T. K. (2017). *How To Play with SIM800L*. https://www.electroschematics.com/introducingsim800l/

Hareendran, T. K. (2022). *ZMPT101B AC Voltage Sensor Module Primer*. https://www.electroschematics.com/voltage-sensor/

Hossain Lipu, M. S., Hannan, M. A., Hussain, A., Ansari, S., Rahman, S. A., Saad, M. H. M., & Muttaqi, K. M. (2023). Real-Time State of Charge Estimation of Lithium-Ion Batteries Using Optimized Random Forest Regression Algorithm. *IEEE Transactions on Intelligent Vehicles*, *8*(1), 639–648. https://doi.org/10.1109/TIV.2022.3161301

IoT Analytics - ThingSpeak Internet of Things. (n.d.). Retrieved December 7, 2023, from https://thingspeak.com/

Jain, A., Kumar, D., & Kedia, J. (n.d.). *Smart and Intelligent GSM based Automatic Meter Reading S ystem*. www.ijert.org

Jaya Deepthi, B., Ramesh, J., & Chandra Babu Naidu, P. (2019). Detection of Electricity Theft in the Distribution System using Arduino and GSM. *8th International Conference on Computation of Power, Energy, Information and Communication, ICCPEIC 2019*. https://doi.org/10.1109/ICCPEIC45300.2019.9082397

Jose, A., Varghese, A. M., Jenson, F. E., Jacob, P., Anumod, D. M., & Thomas, D. (2020). Solid-state circuit breaker based smart distribution board with IoT integration. *Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020*. https://doi.org/10.1109/ICSSIT48917.2020.9214172

Kadechkar, A., Riba, J. R., Moreno-Eguilaz, M., & Perez, J. (2020). SmartConnector: A Self-Powered IoT Solution to Ease Predictive Maintenance in Substations. *IEEE Sensors Journal*, *20*(19). https://doi.org/10.1109/JSEN.2020.2998157

Khorsheed, R. M., & Beyca, O. F. (2020). An integrated machine learning: Utility theory framework for real-time predictive maintenance in pumping systems. *Https://Doi.Org/10.1177/0954405420970517*, *235*(5), 887–901. https://doi.org/10.1177/0954405420970517

Kjeldsen, R. F. (2020). *Stm32f103 pinout diagram*. https://tr.m.wikipedia.org/wiki/Dosya:Stm32f103_pinout_diagram.png

Kumari, K., & Yadav, S. (2018). Linear regression analysis study. *Journal of the Practice of Cardiovascular Sciences*, *4*(1), 33. https://doi.org/10.4103/JPCS.JPCS_8_18

Kwarteng, A., Okrah, S. K., Asante, B., Amanor, P., Aquesi, P., & Baidoo, A. (2021). Design and Construction of an IoT Based Distribution Transformer Condition Design and Construction of an IoT Based Distribution Transformer Condition Monitoring System. *International Journal of Engineering and Science*, *10*(June).

Li, Y., Wang, Z., Li, Z., & Jiang, Z. (2021). Pressure Signal Prediction of Aviation Hydraulic Pumps Based on Phase Space Reconstruction and Support Vector Machine. *IEEE Access*, *9*, 2966–2974. https://doi.org/10.1109/ACCESS.2020.3047988

Liu, J., Zhao, Z., Ji, J., & Hu, M. (2020). Research and application of wireless sensor network technology in power transmission and distribution system. *Intelligent and Converged Networks*, *1*(2). https://doi.org/10.23919/icn.2020.0016

Mallahi, F. A., Mohamed, M., & Shaker, Y. O. (2022). Integration of Solar Energy Supply on Smart Distribution Board Based on IoT System. *Designs*, *6*(6). https://doi.org/10.3390/designs6060118

Mercadier, M., & Lardy, J. P. (2019). Credit spread approximation and improvement using random forest regression. *European Journal of Operational Research*, *277*(1), 351–365. https://doi.org/10.1016/J.EJOR.2019.02.005

Motlagh, N. H., Mohammadrezaei, M., Hunt, J., & Zakeri, B. (2020). Internet of things (IoT) and the energy sector. In *Energies* (Vol. 13, Issue 2). https://doi.org/10.3390/en13020494

MPU6050 Accelerometer and Gyroscope Module. (2021). https://components101.com/sensors/mpu6050-module

Mujawar, R. Y., Lakshminarayanan, R., P., J. A., P., S. P., Patnaik, S., & Dhayalini, K. (2024). IoT-Enabled Intelligent Irrigation System with Machine Learning-Based Monitoring, for Effective Rice Cultivation. *International Journal of Intelligent Systems and Applications in Engineering*, *12*(11s), 557–565. https://www.ijisae.org/index.php/IJISAE/article/view/4476

Nath, R., & Thakur, R. (2018). Zigbee and gsm based fault detection system for low tension pillar. *Advances in Intelligent Systems and Computing*, *624*. https://doi.org/10.1007/978-981-10-5903-2_78

Oliveira, A. L. I., Braga, P. L., Lima, R. M. F., & Cornélio, M. L. (2010). GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation. *Information and Software Technology*, *52*(11), 1155–1166. https://doi.org/10.1016/J.INFSOF.2010.05.009

Onibonoje, M. O., Nwulu, N. I., & Bokoro, P. N. (2019). An Internet-of-Things Design Approach to Real-Time Monitoring and Protection of a Residential Power System. *Proceedings of 2019 the 7th International Conference on Smart Energy Grid Engineering, SEGE 2019*. https://doi.org/10.1109/SEGE.2019.8859879

Ostertagová, E. (2012). Modelling using polynomial regression. *Procedia Engineering*, *48*, 500–506. https://doi.org/10.1016/j.proeng.2012.09.545

Papakitsos, E., Rigakis, I., Chrysocheris, E., Potamitis, I., & Papakitsos, E. C. (2019). *A Low Power Internet of Things (IoT) Communication System for Agricultural Applications Applications of Systemic Methodology via OMAS-III View project Cretan Protolinear Script View project A Low Power Internet of Things (IoT) Communication System for Agricultural Applications*. https://doi.org/10.31058/j.as.2019.33001

Patil, S. G., Mandal, S., & Hegde, A. V. (2012). Genetic algorithm based support vector machine regression in predicting wave transmission of horizontally interlaced multi-layer moored floating pipe breakwater. *Advances in Engineering Software*, *45*(1), 203–212. https://doi.org/10.1016/J.ADVENGSOFT.2011.09.026

Peter, G., Stonier, A. A., Gupta, P., Gavilanes, D., Vergara, M. M., & Lung sin, J. (2022). Smart Fault Monitoring and Normalizing of a Power Distribution System Using IoT. *Energies*, *15*(21). https://doi.org/10.3390/en15218206

Probst, P., Wright, M. N., & Boulesteix, A. L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *9*(3), e1301. https://doi.org/10.1002/WIDM.1301

PyQt - Python Wiki. (n.d.). Retrieved December 7, 2023, from https://wiki.python.org/moin/PyQt

Qt | Tools for Each Stage of Software Development Lifecycle. (n.d.). Retrieved December 7, 2023, from https://www.qt.io/

Quan, Q., Hao, Z., Xifeng, H., & Jingchun, L. (2022). Research on water temperature prediction based on improved support vector regression. *Neural Computing and Applications*, *34*(11), 8501–8510. https://doi.org/10.1007/S00521-020-04836-4/METRICS

Raghav. (2022). *How to avoid Under-fitting and Over-fitting while training a Deep Neural Network?* https://www.datasciencewithraghav.com/2022/10/28/how-to-avoid-under-fitting-and-over-fitting-while-training-a-neural-network/

Ramdham, M. (2022). IoT Based Distribution Transformer Health Monitoring System. *International Journal for Research in Applied Science and Engineering Technology*, *10*(6). https://doi.org/10.22214/ijraset.2022.44754

Shaban, M., & Alsharekh, M. F. (2022). Design of a Smart Distribution Panelboard Using IoT Connectivity and Machine Learning Techniques. *Energies*, *15*(10). https://doi.org/10.3390/en15103658

Shahinzadeh, H., Moradi, J., Gharehpetian, G. B., Nafisi, H., & Abedi, M. (2019). IoT Architecture for smart grids. *International Conference on Protection and Automation of Power System, IPAPS 2019*. https://doi.org/10.1109/IPAPS.2019.8641944

STM32F103 - Arm Cortex-M3 Microcontrollers (MCU) 72 MHz - STMicroelectronics. (n.d.). Retrieved January 18, 2024, from https://www.st.com/en/microcontrollers-microprocessors/stm32f103.html

Stolojescu-Crisan, C., Crisan, C., & Butunoi, B. P. (2021). An iot-based smart home automation system. *Sensors*, *21*(11). https://doi.org/10.3390/s21113784

Üstün, B., Melssen, W. J., Oudenhuijzen, M., & Buydens, L. M. C. (2005). Determination of optimal support vector regression parameters by genetic algorithms and simplex optimization. *Analytica Chimica Acta*, *544*(1–2), 292–305. https://doi.org/10.1016/J.ACA.2004.12.024

Voronov, S., Krysander, M., & Frisk, E. (2020). Predictive Maintenance of Lead-Acid Batteries with Sparse Vehicle Operational Data. *International Journal of Prognostics and Health Management*, *11*(1), 1–17. https://doi.org/10.36001/IJPHM.2020.V11I1.2608

Welcome to Python.org. (n.d.). Retrieved December 7, 2023, from https://www.python.org/

Wu, C. H., Tzeng, G. H., & Lin, R. H. (2009). A Novel hybrid genetic algorithm for kernel function and parameter optimization in support vector regression. *Expert Systems with Applications*, *36*(3), 4725–4735. https://doi.org/10.1016/J.ESWA.2008.06.046

Xue, Z., Zhang, Y., Cheng, C., & Ma, G. (2020). Remaining useful life prediction of lithium-ion batteries with adaptive unscented kalman filter and optimized support vector regression. *Neurocomputing*, *376*, 95–102. https://doi.org/10.1016/J.NEUCOM.2019.09.074

Yousuf, M., Alsuwian, T., Amin, A. A., Fareed, S., & Hamza, M. (2024). IoT-based health monitoring and fault detection of industrial AC induction motor for efficient predictive maintenance. *Measurement and Control (United Kingdom)*. https://doi.org/10.1177/00202940241231473/ASSET/IMAGES/LARGE/10.1177_00202940241231473-FIG13.JPEG

Yuchi, W., Gombojav, E., Boldbaatar, B., Galsuren, J., Enkhmaa, S., Beejin, B., Naidan, G., Ochir, C., Legtseg, B., Byambaa, T., Barn, P., Henderson, S. B., Janes, C. R., Lanphear, B. P., McCandless, L. C., Takaro, T. K., Venners, S. A., Webster, G. M., & Allen, R. W. (2019). Evaluation of random forest regression and multiple linear regression for predicting indoor fine particulate matter concentrations in a highly polluted city. *Environmental Pollution*, *245*, 746–753. https://doi.org/10.1016/J.ENVPOL.2018.11.034

ZMCT103C - Precision Current Sensor Pinout, Features, Datasheet, Working, Application, Alternative. (n.d.). Retrieved January 18, 2024, from https://components101.com/sensors/zmct103c-precision-current-sensor-pinout-features-datasheet-alternative