Predictive Analytics in Manufacturing

Sowmya Ravichandran

Department of Applied Data Science, San Jose State University

DATA 270: Data Analytics Processes

Dr. Eduardo Chan

December 10, 2021

Abstract

Predictive analytics in manufacturing assists manufacturers in predicting machine failures and anticipating machine maintenance requirements ahead of time to improve the quality of the manufacturing process and to reduce the machine downtime. The water pump failure prevention problem is being addressed by combining IoT sensors to monitor the components, machine learning models to forecast machine failure, and a dashboard to display machine and sensor status. In the past, reactive maintenance was utilized, which involved personally inspecting each component when the water pump failed. The failure of the pump resulted in a significant financial outlay for manual water transportation to that location. The IoT-based monitoring has shown to be efficient and cost effective because of decreasing sensor prices and developments in cloud services. The time series data from sensors is gathered by the IoT gateway, pre-processed using cloud services, and then fed to machine learning models. Four distinct models namely eXtreme Gradient Boosting (XGBoost), Autoregressive Integrated Moving Average (ARIMA), Logistic Regression (LR), and Long Short-Term Memory (LSTM) were implemented, with the XGBoost method surpassing the others in terms of failure prediction with an accuracy of 100%. The ARIMA, LR, and LSTM, on the other hand, exhibited 70%, 85%, and 99% accuracy, respectively. The main benefit of this project is that the expenses of water pump failure were considerably reduced because the problems was discovered, and corrective steps were taken in a timely manner.

Keywords: Water pump failure prediction, IoT sensors, machine learning models, XGBoost, ARIMA, LR, and LSTM

1. Introduction

1.1 Project Background and Execute Summary

Industrial Revolution led to increase in the number of manufacturing companies. It paved way for a transition from goods made manually to machines. As it involves a huge investment in operating the machines, the output produced was very crucial. Due to globalization the industries should raise the quality of the product and on-time delivery to hold a position in the global market. To compete with other manufacturers an effective mechanism should be in place to monitor the functioning of the machines.

Manufacturing industry produces a large amount of data. The rapid development of data analytics made it easy for manufacturers to analyze the data and get useful insights. Analytics is the process of analyzing the data generated to find the underlying relationships and trends (*What is analytics?,* n.d.). Predictive analytics is a branch of data analytics, a data driven approach that helps to predict the outcome based on the historical data, machine learning models and statistical methods (*Predictive analytics: What it is and why it matters,* n.d.). Predictive analytics in the field of manufacturing help the company to identify the failure in advance and to take corrective measures before it paralyzes the production. By using this technique an industry can perform maintenance activity at the right time which would reduce cost, keep the engineers informed about which part of the machinery need to be replaced in advance and to keep a stock of the replacement available on time. Unfortunately, many small and medium scale companies have no such identification mechanism, due to a huge investment in collecting data from various machines and storing in a data warehouse. With the decline in cost of sensor technology and leap in Bigdata, paved way for manufacturers to reap the benefits of this innovation.

The focus of this paper is prediction of possible failure patterns in a water pump machine. The water pump machine often goes down and causes serious problems to the people in an area. This targeted problem is solved by first preprocessing the data collected from the sensors, using preprocessing techniques and applying various machine learning models to make accurate predictions.

IIoT (Industrial Internet of Things) sensors, widely used in manufacturing, supply chain monitor and management, are connected to machines to fetch the real time data more accurately (Posey et al., 2021). It generates massive amount of data, approximately 500 gigabytes per day from a single sensor. Big Data technology to store and process huge volumes of data. Big Data represents the High Volume, Velocity and Variety of data (Gillis, 2021). The data collected is then preprocessed using various preprocessing techniques such as standardization, scaling to different ranges [0,1] and [-1,1], minmax scaler, unit vector scalar/normalizer. The machine learning models are trained using this data. The challenge is to select the best model amongst the many. The data collected for water pump machine is a time series data. Some of the models which are suited for this data are the XGBoost, ARIMA, LR and LSTM. The accuracy of each model is calculated, and the best model is used for prediction.

The contribution of this paper is to solve the water pump machine problem since it is causes serious issues to the people in that area. The possible patterns of failure will be analyzed and then by using the best machine learning models the future failures can be predicted, and corrective measures are taken on time.

1.2 Project Requirements

The data from the water pump machines is collected using IoT sensors. The amount of data produced is very huge and traditional methods cannot be used to analyze this data. Big

data analytics helps to uncover the trends and patterns in a data and machine learning using decision-making algorithm will help to accelerate this process. There are various algorithms which can produces highest prediction accuracies. The efficiency of these algorithms is measured based on the following metrics: Precision, recall, F1-score, confusion matrix and AUC-ROC curve.

To implement predictive analytics on the water pump machine to identify the possible failure patterns, firstly, the data must be defined. The following three types of data need to be collected. The first is the raw data which includes physical and chemical characteristics. The machine data such as vibration, accelerometer, pressure, temperature, motor current, ultrasound. Finally, the product inspection data to validate the dimensional error. The preprocessing of the collected data is important to achieve high prediction accuracy. There are various scaling techniques available, standardization, minmax scaler, unit vector scalar/normalizer. The effectiveness of these techniques is measured by calculating the R^2 score. The scores can be positive or negative, but the best score would be one (Fernando, 2021). This will measure how accurately the model predicts for the future samples.

1.3 Project Deliverables

This paper aims at developing a best model for the proposed problem to predict the water pump failure in advance so that the corrective measures can be taken on time. Also, an alerting system to report pump related anomalies to the field engineers before the system completely paralyses. The sensors installed in the water pump machine must be monitored regularly as the machine leaning model depends on the data that these sensors collect, hence a dashboard is built to monitor the working of the sensors. A mailing system to send the pump

status report in a timely manner to the engineers. Finally, presenting a report detailing the process of solving the water pump machine problem.

1.4 Technology and Solution Survey

Entropy is an acute issue for many manufacturers. Companies have also carried out schedule-based maintenance to conditional based maintenance to find when a machine would fail, but nothing proved beneficial. It caused business slow down and incurred a huge loss. In the past, reactive maintenance was utilized, which involved personally inspecting each component when the water pump failed but again this increased the downtime and degradation to the quality of the product. Predictive analytics in the field of manufacturing help the companies to identify the machine or the system failure in advance. Currently, in many small and medium industries there is no mechanism to predict the machine failures due to huge investment. Data collection and storage was a major concern for these industries.

With the advancement in the cloud technologies and reduction in the cost of sensors many companies have moved forward to adopt the predictive analytics to monitor the machines and improve the productivity (Ennomotive, n.d.). Once the data is collected and stored, data is processed to make insights. The first stage is to pre-process the data so that the machine learning model can interpret it, and then give it to the models so that they may learn from it and forecast future values. Some of the machine learning models for this problem domain are the XGBoost, ARIMA, LR and LSTM.

The XGBoost is a boosting ensemble technique which creates multiple models, learn from the mistake of the previous model and the final model will have a very less error. The XGBoost algorithm is a variation of the gradient boosting technique that considers the expected

value as well as the residue. The learning in the XGBoost algorithm happens by optimizing the loss. It is a very popular algorithm as it aids in improved performance and accuracy (Brownlee, 2021). The pump failures must be investigated as soon as possible. The XGBoost algorithm is known for its execution speed and performance, hence this model was selected for predicting the water pump failures.

The ARIMA is the next model considered for the problem domain. The AR of the model stands for autoregression where the input is taken from the previous observation to predict the next outcome. The I in the model represents the integration, it is calculated by taking the observation from the previous timestamp and subtracting it with the current timestamp. The MA moving average smooths the noisy observations by taking the average of a particular number of observations. The ARIMA model has 3 parameters namely the p, d and q. The p applies to the AR part which holds the number of previous observations given to the model. The d is the order of the integrated part which holds the number of times the difference between previous and current observation are taken. The q is the MA part which holds the average window size. The time series must be a stationary with constant mean and standard deviation. If the time series is not stationary and if it is linear then the I, the integration of the ARIMA model transforms a non-stationary data into a stationary data (Brownlee, 2020). Generally, forecasting is tough but the ARIMA model is a very flexible compared to the other statistical models. The dataset for this problem is a time series dataset, which this model is well suited for.

The LR is one of the simplest binary classification models and used if the two classes are linearly separable. The best fit line is calculated using the linear regression hypothesis, which separates the two classes on either side of the plane. To get the best fit line the summation of all the points along with the distance are calculated and the maximum sum decides the best fit line.

Suppose if there is an outlier then the value of the best fit line will be either greater than one or less than zero. This may greatly affect the model performance and the summation would result in a negative value. To draw the best fit line in the logistic regression model a cost function called sigmoid is used. This function would remove the effect of outliers in the model (*06: Logistic regression,* n.d.). The dataset for water pump problem is linearly separable and the target feature is categorical value that forecasts the likelihood of a given instance belonging to one of several classes. This is one of the simplest algorithms.

The LSTM is the next research model, which was created to overcome the inadequacies of the Recurrent Neural Network (RNN). The RNN model iteratively presents the input and previous learning, to assist the model to predict next outcome. However, if the context of a particular input changes, and the output is dependent on the input provided during the beginning stage, the model must go back in time to get the information, but the model does not remember this knowledge, and the prediction may degrade as a result. The LSTM model was created to address the drawbacks of RNN. The LSTM model has a memory cell for remembering and forgetting the information based on the context of the input. They have the capability of remembering for a long period of time. The RNN has a single neural layer whereas the LSTM has four layers (Sivalingam, 2020). Time series data is collected; hence this model is appropriate as the previous timestamp data is kept in memory for a long time, the model learns from the data in memory to predict the future value.

1.5 Literature Survey of Existing Research

Numerous studies have already been done on this problem. Below is the brief description of few researches done so far in the field of predictive analytics in manufacturing.

Wu et al. (2017) performed a comparative study on different algorithms to predict CNC machine tool wear, which is widely used for drilling, milling, and turning. The failure of the CNC machine is due to overload, corrosion, overheating and friction. The failure of these machine would cost more and lowers the productivity. In this paper, the IoT sensors are installed to monitor the machine parts and send the data to the cloud services, where the data is preprocessed and sent to the machine learning models. The different algorithms in study were the Random Forest (RFs), Artificial Neural Network (ANN), Support Vector Machine (SVM). Among the three, the measure of coefficient of R^2 of RFs was 0.992. Even though, the training time was more for RFs, but it outperformed the other two in terms of squared error and R^2 values predicting with high accuracy.

Han and Chi (2016) focuses on forecasting the wear compensation offset for CNC machine tools. The CNC Die Cutting machine often fails due to many random reasons. Here, the author stresses on focusing more on understanding the collected data. The pre-processing of the data is crucial to attain highest accuracy. Each component of the machine is measured in different units and hence normalization of the data is chief to build a meaningful model. Various scaling techniques such as standardization, minmax scalar, unit vector scalar/normalizer is implemented. The effectiveness of these techniques is measured by calculating the R^2 score. The score can be positive or negative, but the best score would be one. The techniques used here is to scale the data to two different ranges [0,1], [-1,1]. The paper concludes by stating that proper pre-processing of data is very crucial and the method for processing data must be carefully selected. This issue not only applies to CNC but also to any machine in the manufacturing industry.

Kanawaday and Sane (2017) researched on the Industrial Internet of Things (IIoT), that greatly helped the industries to gather a massive amount of data from the machines. The slitting machine has unwinders, rewinders, blades and knives. The machine works by unwinding the roll from the rewinder, straightened, and fed to the machine to cut to desired size. When the machine is turned on, the pressure and tension rise, causing the roll to degrade. When the machine runs, the pressure and tension values are established, but the values may grow, causing serious damage to the material. The time series data is collected, ARIMA model is applied to solve this problem. The paper concludes by stating IoT based machine learning is very efficient and cost effective.

Lechevalier, Narayanan and Rachuri (2014) discussed about a few of machine learning algorithms by stating examples of machineries that uses these algorithms to detect the fault in advance. To detect fault in the Base Transceiver Station (BTS), a Bayesian model was used .

Bayesian network is fed with the reasons of failures from the data which is collected from previous experience. When the system fails, it gets directed to the Bayesian nodes to find the reason for failure and the remedy. This helps the engineers to fix the problem. By using this technique, the time taken to correct a complex fault is drastically decreased. Based on all the above experiments, the author has also built a generic framework for predictive analytics in manufacturing. The main aim is to provide a platform for the manufacturing domain experts to easily build models for predictive analysis. This is a four-step process, first is the meta model repository, the manufacturer must specify the components of the machine. Second is the data collection process that facilities collecting and pre-processing data. In the third stage Diagnostic and Predictive Models are generated based on the

specifications given in the meta model. The last stage is to visualize the results from the model.

While comparing the results of the experiments from all the above papers, it is evident that the dataset considered for the analysis is important, as the machine learning technique is selected based on the type of data. The preprocessing is a crucial step to get results with high accuracy. There are many techniques for scaling the data, the most appropriate technique must be selected based on R^2 value. Similarly, many models have been considered for the study, ANN, ARIMA, SVMs, RFs. Each model has its own pros and cons. Based on the dataset, the accuracy levels must be calculated, and models must be used appropriately.

2. Data and Project Management Plan

2.1 Data Management Plan

The data for the water pump prediction problem is collected by using IoT sensors. The batch pump_sensor_data is collected from Kaggle (*Pump_sensor_data,* 2019). The IoT device connected to the water pump machine will collect two types of data, the status data, and the location data. The status data contains raw information from the sensor which must be processed for analysis. The location data, enables to find the location of the water pump. The data sent by the IoT device can be of any format namely JSON, XML or sometimes text files.

The data is highly unstructured and of high volume. A traditional database however is not very effective to handle such large volumes. This data can be easily stored and accessed in a public cloud infrastructure. Therefore, Amazon S3 bucket is chosen to store the unstructured data. The long-term storage and access of data can be achieved by Amazon S3 Glacier. For the water pump failure problem, few sensors are already installed, and the data is collected from these sensors. Unfortunately, there was no mechanism to store the data and analyze to get useful insights.

Firstly, additional sensors are installed in the water pump machine based on the parameters which are required for analysis. At the time of an event these sensors would transfer the data using IoT gateways to the server in the cloud. The data from the sensors are transferred to IoT gateway using either an ethernet, Wi-Fi or a Bluetooth. The communication between the gateway and the cloud is by using the MQ Telemetry Transport (MQTT) queuing protocol. With the help of Kinesis Data Streams, the data is pushed into the Amazon S3 bucket where a folder is created, inside this folder the data from the machine is stored with the current date as the file name. The data which is captured on a particular day will become a batch data the next day. To

maintain the batch and stream continuity an AWS Lambda function is written to validate and version the batch file inside the same folder. Time series data is collected from the sensors. To check the quality of the data, two major checks must be in place. The timestamp integrity check to ensure if the date-time pair is sequential and the range check to ensures if the values fall in specified upper and lower limits.

The following will be some useful metadata to accompany data collection, where to place the sensors that will aid data collection, internal components of pumps captured by studying the pump design document, heat bearable capacity of the IoT device, data network connections, life span of an IoT sensor, placing the sensors should not short circuit the other electronic equipment's inside the pump.

The access to the data is given only to a group of analysts who will be working on this problem. The collected data is purely about the machine and no personal identification; therefore, it is not required to hide any identity. Data sharing is restricted as is has a complete detailing of the machine and locations of where the pump is installed. The data from the sensors is transmitted through network which causes a high security risk. The following methods can be adopted to secure the data. Changing passwords of the router, using strong passwords, avoid public networks, encrypt the data. Once the data is inside the Amazon S3 bucket the data can be secured by defining IAM roles and giving access only to specific group of people. By this way we can provide secure access to data to all collaborators.

A huge volume of data is expected to be generated from the sensors. To cater to this large volume, we rely on cloud services. The charges for the storage will be based on AWS standards. For long-term storage and backup Amazon S3 Glacier service is used. AWS is responsible for backup and recovery. The data which is already sent to AWS cloud can be easily recovered.

For this problem domain to predict the future failure of the water pump, the historical data is very important. The streaming data is collected from sensors using gateway and sent to the Amazon Kinesis. This data is then stored into Amazon S3 bucket, cleaned using Amazon Glue. Machine learning models are developed using this data to forecast the pump failure in the future. All the data must be securely archived. The accuracy and speed of data generated by each sensor can be analyzed and the sensors can be replaced overtime to produce high data accuracy. Once the machine learning model is implemented and trained by using the historical data, may be a part of the historical data can be deleted.

The complete process is documented which facilities the potential users to find the data. In order gain access the user must be authenticated via the IAM roles and groups. If the project requires additional analysts to work on the problem, then the data must be shared to the person who might involve in the activity. The data sharing is done by providing access through the AWS IAM service.

2.2 Project Development Methodology

The project development methodology used for this problem is the CRoss-Industry Process for Data Mining (CRISP-DM). This method has different phases, in each phase a group of tasks is carried out.

2.2.1 Business Understanding

The first phase is the business understanding, here the business process is studied. In our problem domain, water is delivered to a community of people from lake using pump. Firstly, the location of the pumps is identified, the capacity and power of the pump is studied, electricity consumption is collected. Secondly, the business problem is studied, here the cause of pump failure, increase in water delivery cost and motor maintenance cost are analyzed. The main

business objective is to understand the mechanics of the pump and to install additional sensors wherever required. The end goal is to reduce the pump failure by 80%, thereby reducing the expenses of water pump failure by predicting beforehand and fixing the issue on time.

2.2.2 Data Understanding

The second phase is the data understanding, the data is collected from the IoT device. The collected data can be of any format namely XML, JSON, CSV or sometimes text files. The data is then stored into the Amazon S3 bucket and analyzed using Amazon Athena writing simple SQL queries. Here the independent and dependent features are identified and the important feature that directly affects the functioning of the pump is extracted. The quality and completeness of the data is verified.

2.2.3 Data Preparation

The third phase is the data preparation, here the data is obtained from various sources. The stream data is collected from the IoT sensors and sent to the Amazon Kinesis. By using Amazon Data Stream this data is pushed into the Amazon S3 bucket. The batch data is stored in the Amazon S3 bucket. Now, data validation is carried out for stream and batch data continuity. The data is now transformed using Amazon Glue. The missing data and anomalies are identified, the spark script is updated, ETL job is run.

2.2.4 Modelling

The fourth phase is modelling, the independent and target features are identified, split into training and testing sets. The models are analyzed based on complexity, maintainability, and available resources. The models suitable for time-series data namely XGBoost, ARIMA, LR and LSTM are selected for this project. Building the model by using Scikit-learn and TensorFlow in Amazon SageMaker, training the model by using training data. The best model is selected, then

the models are accessed with various parameters and the changes are observed. The Amazon SageMaker studio is configured to visualize the results of the model.

2.2.5 Evaluation

The fifth phase is the evaluation, the training time is calculated for each model. By using the precision, recall, F1 score, confusion matrix and AUC-ROC curves the performance is measured. The model has been optimized to increase efficiency. The model findings are checked for correctness. Then the results are compared with the requirements. If the results are not satisfactory remodeling is done. Evaluating the visualization tool if it has captured the failure.

2.2.6 Deployment

The last stage is the deployment, here the data pipeline is made ready so that the model is straight forward for the consumers. The model is deployed. A plan for monitoring and maintenance is made. Final report and presentation are made available, and the project is reviewed if all the steps are correctly implemented.

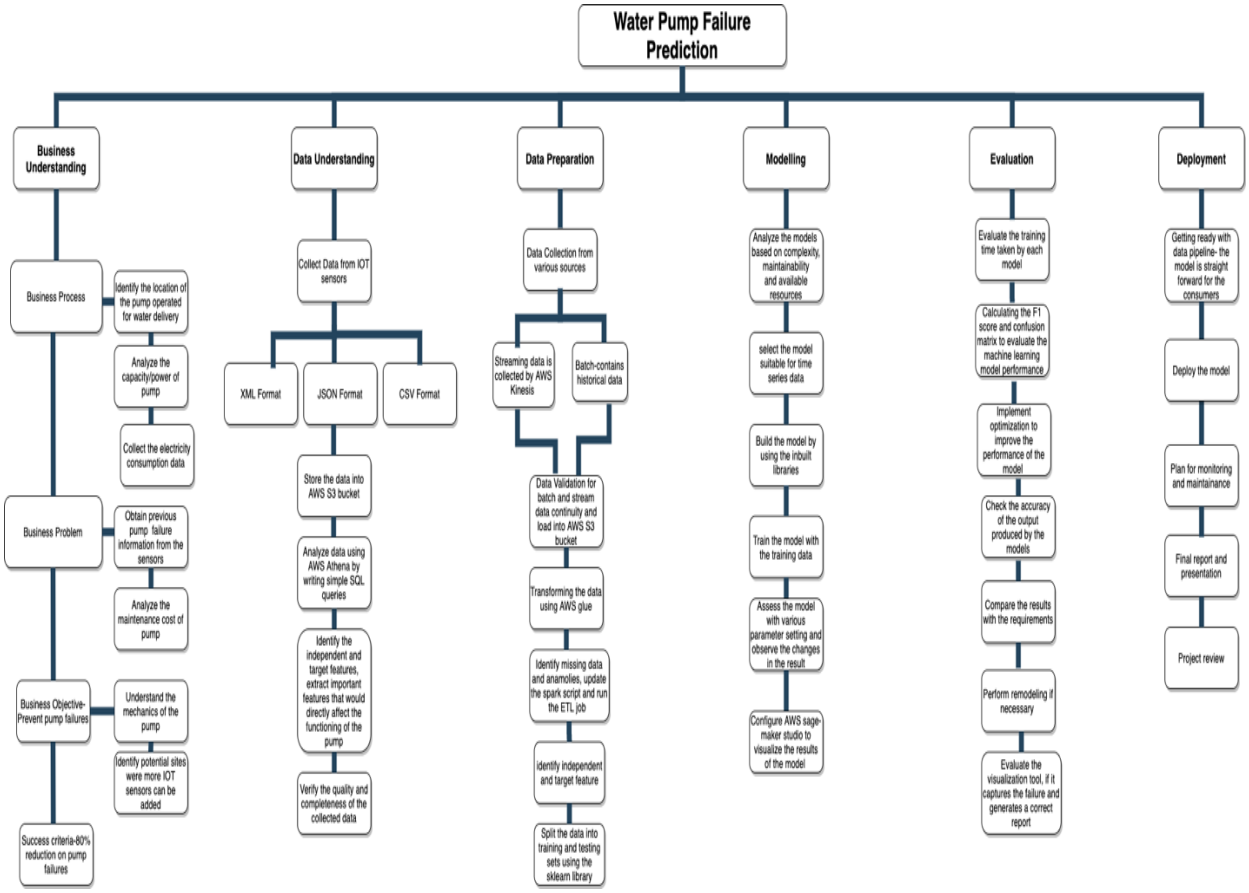
2.3 Project Organization Plan

As shown in the Figure 1, the task required for the completion of each phase of the water pump failure prediction problem domain is listed in the work breakdown structure. The business understanding phase where the business process, problem and objective is analyzed. The data understanding phase, the type of data, storage mechanism, identifying important feature is carried out. The data preparation phase is where the stream and batch data continuity is validated, transformation of data, identifying independent and target features, splitting the data into train and test sets. The modelling phase, the model is selected based on the complexity, maintainability, and available resources. The model is trained to make predictions. In the evaluation phase, the performance and the prediction of the model is evaluated. If the results are

not satisfactory remodeling is done. In the deployment phase, the alerting system is configured along with the sensor monitoring UI and the pipeline is made available for easy use to customers, the entire procedure is documented, and the project is reviewed to check if all the phases are completed.

Figure 1

Work Breakdown Structure



Note. Work Breakdown structure for water pump failure prediction.

2.4 Project Resource Requirement and Plan

The Table 1 lists the resources required for the project. The current system has few sensors already installed, but to capture other important measures additional sensors are required. The cost of one sensor is estimated to be 38 cents (Alsop, 2016). As the data is generated every

second, it must be captured for analysis. The IoT sensors use Wi-Fi to connect to the gateways, which collect data from the sensors, aggregate it, and transfer it to the cloud (Posey & Lavery, 2021). A large volume of data is generated by sensors, to effectively manage cloud platform is required, hence AWS is chosen for this project. The data is received by Amazon Kinesis, which streams the real-time data, that helps in analyzing possible failures on time (Walmsley & Bie, 1983). For streaming the training data, the cost is estimated to be $11.21 for one month (*AWS Pricing Calculator*, n.d.). The data is stored in the Amazon S3 bucket for further processing. It is archived using Amazon Glacier, the cost is estimated as $2.37 for one month (*AWS Pricing Calculator*, n.d.). Data is analyzed using Amazon Athena, the cost for running approximately 100 queries is estimated to be $24 (*AWS Pricing Calculator*, n.d.). Then the data is cleaned before sending to the machine learning model. Amazon Glue transforms the data, by removing the outliers, missing values, and any anomalies in the data and this is crucial as the data is collected in many formats (Harrison et al., 1997). The cost for cleaning the data using Amazon Glue is estimated to be $3.6 per hour (*AWS Pricing Calculator*, n.d.). The Amazon SageMaker has tremendous capabilities for building and training machine learning models, as well as visualizing the model findings (Mishra, 2019). The cost of the Amazon SageMaker is estimated to be $3.8 per hour (*AWS Pricing Calculator*, n.d.). A webserver is configured in Amazon EC2 instance to serve data for the UI developed, using react framework which showcases the health check details of the sensors which aids in sensor monitoring. The estimation for the usage of Amazon EC2 is $20 per month (*AWS Pricing Calculator*, n.d.). The total cost for implementing this project is estimated to be $647.64. The pipeline is run at regular intervals, the AWS will serve the purpose effectively.

Table 1

List of Resources and Estimation

Function	Resource Type	Resource	Time Duration	Cost
Sensors to collect data	Hardware	IoT sensors	08/20/2021- 2/31/2021 (5 months)	$3.8
Streaming data	Software	Amazon Kinesis	09/10/2021-12/31/2021(4 months)	$44.84
Data storage	Hardware	Amazon S3 Bucket	09/10/2021-12/31/2021 (4 months)	$5
To archive data	Hardware	Amazon S3 Glacier	09/10/2021-12/31/2021(4 months)	$10
To run SQL queries	Software	Amazon Athena	09/10/2021-12/31/2021(1 months)	$24
ETL	Software	Amazon Glue	10/01/2021-12/31/2021 (3 months)	$220
ML-Framework	Hardware and Software	Amazon SageMaker with installed scikit-learn and TensorFlow – on Amazon EC2 ml. p3.2xlarge	10/01/2021-12/31/2021 (3 months)	$300
Web server	Hardware and Software	Amazon EC2 Python 3.9 and Django 3.0	11/01/2021-12/31/2021 (2 months)	$40 Free
UI framework	Software	React	11/01/2021-12/31/2021 (2 months)	Free

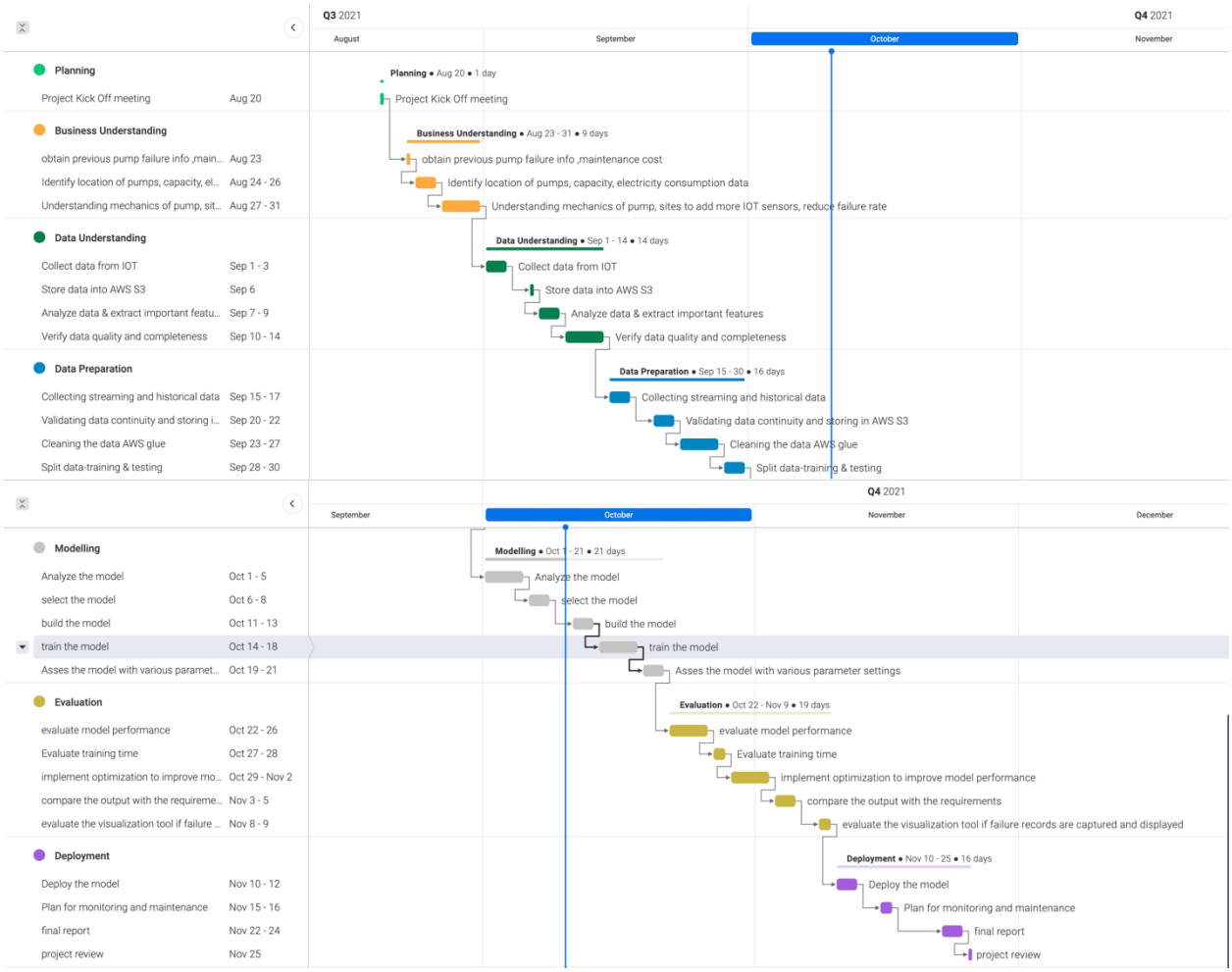
Note. List of Hardware and Software.

2.5 Project Schedule

The Gantt Chart shown in Figure 2, has different phases and timelines for each task involved in predicting the water pump failures. The bars denote the time taken for each activity and the arrows shows the dependencies between the tasks.

Figure 2

Gantt Chart



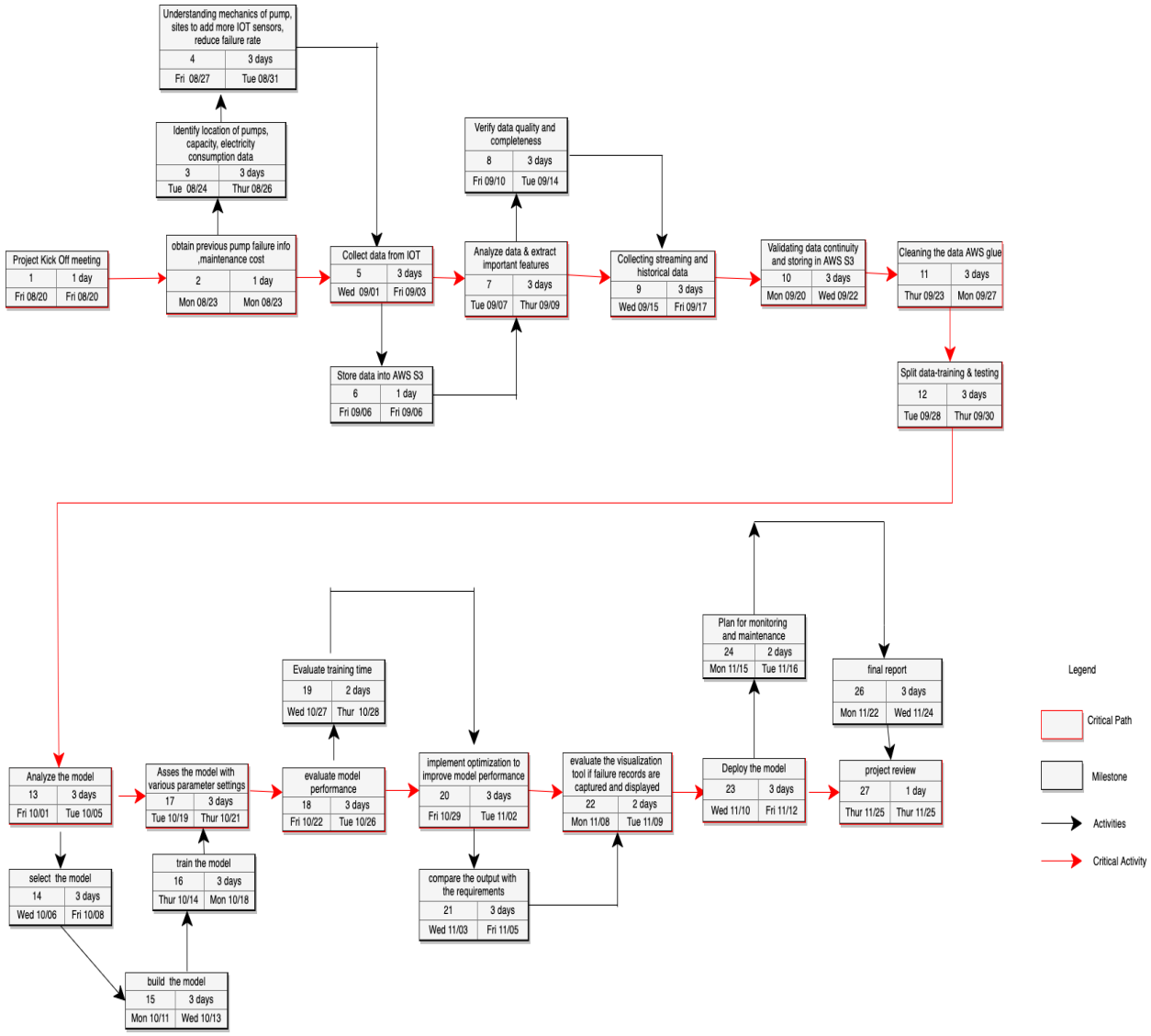
Note. Project phases and timelines.

The Program Evaluation Review Technique (PERT) shown in the Figure 3 illustrates the different activities and their timelines. All the phases of the project are captured in the chart. The

dependencies can be easily visualized and tracked. The critical path is denoted by a red arrow. All other milestones are marked with a black arrow.

Figure 3

PERT Chart



Note. List of project tasks and estimated time for each task.

3. Data Engineering

3.1 Data Process

The failure of a machine can be detected by monitoring its functionality at regular intervals of time. The data is crucial, as it gives valuable insights for the cause of failure. The data for the water pump failure prediction problem is collected from the IoT sensors. The sensors are installed in the water pump machine to record and transfer the data for further processing. The IoT sensors use Wi-Fi to connect to the gateways, which collect data from the sensors, aggregate it, and transfer it to the cloud (Posey & Lavery, 2021). In our problem domain few sensors are already installed, to captured data. The batch data consists of the observations for five months captured for every second. The initial analysis of the batch data reveal that few parameters of high correlation with the water pump failure are not captured. To better assess the cause of failure few sensors are installed. The streaming data with the newly installed sensors are captured and sent to cloud. The data cleaning is crucial as the prediction of the model depends upon the correctness of the data. Often, the sensors may malfunction, therefore there are high chances of missing values or incorrect data which causes outliers. This might seriously affect the model, so these processes are of high importance. The raw data consisting of various parameters from each sensor is measured in different units. The normalization technique is used to convert all the parameters to a common scale, which helps in achieving high accuracy machine learning models. Once the data is cleansed, the training, testing and validation sets must be carefully selected. In the water pump machine failure prediction problem, time series data is recorded, in which each row is dependent on the previous row. There is no independent data as in the non-time series datasets. The data is organized by dates, randomly separating it would result in a lot of distortion because the prediction is based on the prior days data. To keep the data consistent, it

is divided into two sets: training data and test data, with 70% of the data going to training and 30% going to testing. The training data should have equal number of failure and success cases for the model to better understand the data from various angles. Therefore, biased data will produce undesirable outcome and the purpose of having a prediction model is inoperative. By using the validation set the model is tested with various folds of the dataset and the accuracy is measured in each fold and aggregated to get the maximum and minimum accuracy of the model. Once the model is fine-tuned, the test data is used to make the model predict and the accuracy is verified.

3.2 Data Collection

Data collecting is essential for digging further into the problem and analyzing it to get more insights. The first stage is to assess the problem domain and decide the type of data that will be needed to solve it. The data can be from any relevant source pertaining to the domain. In our problem domain, the water pump machine fails for random reasons. Only if the machine is closely monitored by collecting the data of each component in a timely manner the issue can be identified. The primary data and the secondary data are the two types of data collected from the source. In our problem domain, the data is collected for five months recording the behavior of each component every second. The data is collected by the local authority and shared for analysis constitutes the secondary data taken from Kaggle (*Pump_sensor_data,* 2019). As there are a few more critical components to monitor, the primary data will be obtained directly from the water pump. The primary data is further divided into two categories qualitative data and quantitative data. According to the findings of this study, the following characteristics, such as suction pressure, discharge pressure, flow, vibration, temperature, air pressure, pump speed, humidity, and power, are qualitative data for this problem. To analyze the problem accurately the data must

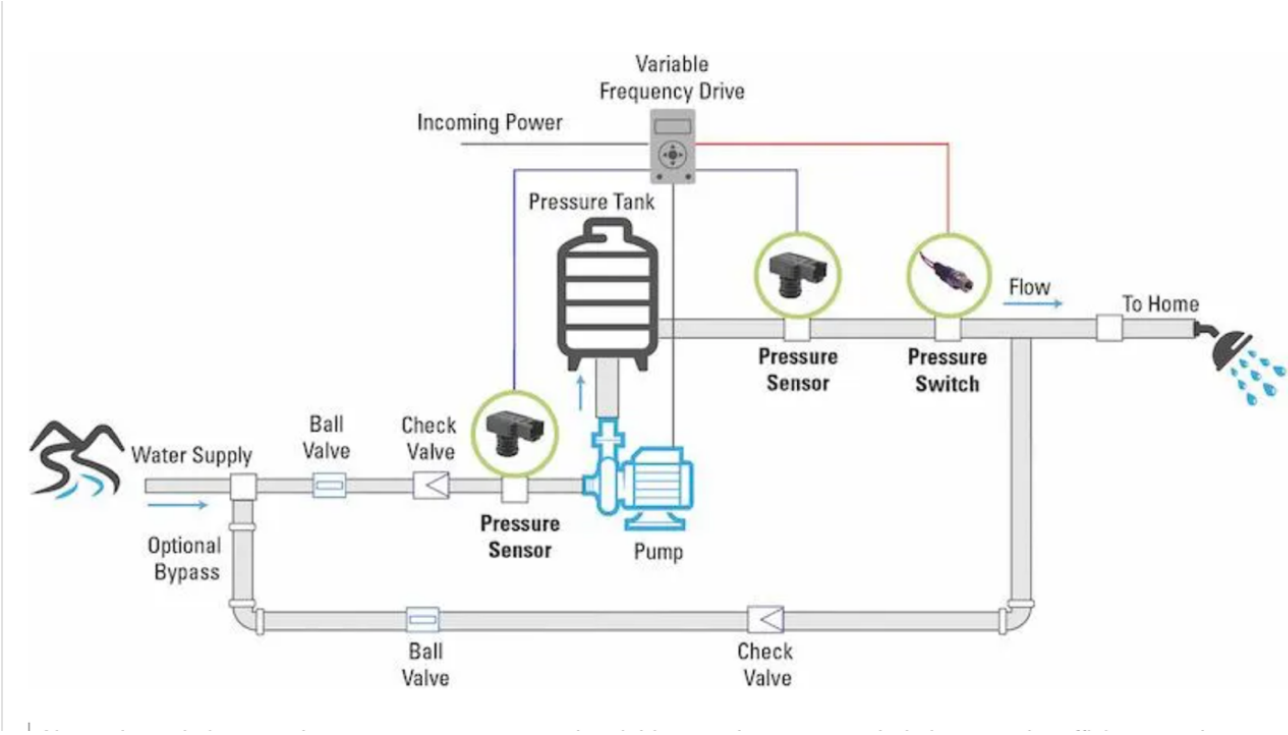
be recorded for every second. By gathering additional information about the operation of a water pump and the various components within the machine, the topic of our research is fully understood.

Understanding the physics of the pump is critical to solving the water pump machine problem. The various components of the pump must be examined in greater depth. A method should be in place to capture these values in real time. The data is collected by installing IoT sensors inside the water pump machine. A variety of sensors are used for different components.

As shown in the Figure 4, the amount of water sucked from the inlet valve is measured by the suction pressure of the pump situated between the check valve and the motor. The pressure sensor captures the value in pounds per square inch (psi) scale. The pressure sensor, measuring the discharge pressure in pounds per square inch (psi) is located between the pressure tank and the pressure switch. The volume of water a pump can discharge is measured in gallons per minute (gpm) by the flow sensor is installed between pressure switch and water outlet. The vibration sensor measuring the vibration is mounted on the bearing housing of the pump, measured in Hertz (Hz). The thermistor sensor measures the temperature of the motor coil in Fahrenheit (^{0}F), this is very important as it may seize the motor if it is over heated. The power sensor placed between the incoming power and motor measures the current entering the system. The pH level of the water is measured by the water quality sensor. This value indicates if the water is safe to drink and ensures that no dangerous substances are present. These are some of the main sensors for motoring the functioning of the water pump.

Figure 4

Image of Water Pump Machine With Installed Sensors



Note. Water pump machine with installed sensors. *"How variable speed pumps & pressure sensors can improve water system | WQP"*, by Mullen, D, 2020. Scranton Gillette Communications. https://www.wqpmag.com/commercial-water/how-variable-speed-pumps-pressure-sensors-can-improve-water-systems. Copyright 2021 by Scranton Gillette Communications.

Table 2

Data Collection Plan

Variable title	Suction pressure	Discharge pressure	Flow	Vibration	Temperature	Power	Machine status
Unit of measurement	Pounds per square inch (psi)	Pounds per square inch (psi)	Gallons per min (gpm)	Hertz (Hz)	Fahrenheit	watts	NA
Data type	float	float	float	float	float	float	string
Collection method	sensor	sensor	sensor	sensor	sensor	sensor	sensor
If manual	No	No	No	No	No	No	No
Gauge/ instrument	Water pressure sensor	Water pressure sensor	Flow sensor	Vibration sensor	Thermistor sensor	Power sensor	sensor
Location	Between check valve and the motor	Between pressure tank and pressure switch	Between pressure switch and water outlet	Mounted on bearing housing of the pump	Placed in the motors coil	Between incomig power and motor	NA
Gauge calibrated?	Yes, calibrated by software	Yes, calibrated by software	Yes, calibrated by software	Yes, calibrated by software	Yes, calibrated by software	Yes	Yes
Measurement system checked?	Yes,	Yes	Yes	Yes	Yes	Yes	Yes
Accuracy adequate?	Yes, verified with zero and full-scale pressure	Yes, verified with zero and full-scale pressure	Yes, verified by using a flow meter	Yes, verified	Yes, verified	Yes, verified	Yes, verified
Historical data exist?	yes	yes	yes	yes	yes	yes	yes
Source of historical data	Kaggle	Kaggle	Kaggle	Kaggle	Kaggle	Kaggle	Kaggle
Historical data reliable?	collected from a pump	collected from a pump	collected from a pump	collected from a pump	collected from a pump	Collect from a pump	Collect from a pump
Mean	41.91	41.47	2.3	590.67	60.7	576.19	576.19

Variable title	Suction pressure	Discharge pressure	Flow	Vibration	Temperature	Power	Machine status
Upper specification limit	60	60	5.5	7.5	200	1100	Normal
Lower specification limit	40	40	4.5	1000	115	250	Broken
Standard deviation	13.06	12.09	0.7	144.02	37.6	225.76	NA
Minimum sample size (MSS)	20% of total data points	20% of total data points	20% of total data points	20% of total data points	20% of total data points	20% of total data points	20% of total data points
Sampling frequency	10000 Hz	10000 Hz	10000 Hz	10000 Hz	10000 Hz	10000 Hz	10000 Hz
Data collector	Analyst	Analyst	Analyst	Analyst	Analyst	Analyst	Analyst
Operational definition exists?	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Data collector trained?	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Start date	08/20/2021	08/20/2021	08/20/21	08/20/21	08/20/2021	08/20/21	08/20/21
Due date	11/20/21	11/20/21	11/20/21	11/20/21	11/20/21	11/20/21	11/20/21
Duration	90 days	90 days	90 days	90 days	90 days	90 days	90 days

Note. Data collection plan to collect data from the newly installed sensors.

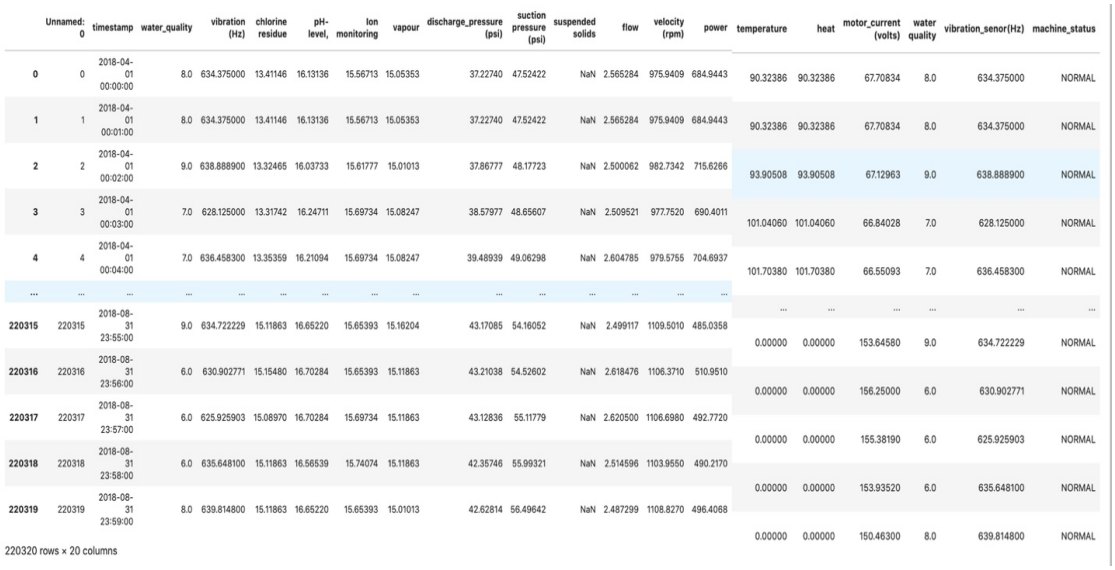
The data collection plan is summarized as shown in the Table 2. The data are collected from the additional sensors which is responsible for capturing the suction pressure, discharge pressure, flow, vibration, temperature, and power are sent to IoT gateway. The sensors and the gateway are connected through Wi-Fi routers. The data collected in the gateway is processed at the edge of the network, as the sensors capture a huge amount of data, the IoT gateway help to filter out the unnecessary information and pass on the critical data to the cloud. The data collected from all the sensors are so large that it cannot be kept in a standard database. Cloud

services are employed to handle the increasing volume. The data is transferred through Amazon Kinesis, with the help of Amazon Kinesis Data Streams, the data is pushed into the Amazon S3 bucket where a folder is created, inside this folder the data from the machine is stored with the current date as the file name. The data which is captured on a particular day will become a batch data the next day. To maintain the batch and stream continuity an Amazon Lambda function is written to validate and version the batch file inside the same folder. Two primary checks must be in place to ensure the data quality. The timestamp integrity check to ensure if the date-time pair is sequential and the range check to ensures if the values fall in specified upper and lower limits. Data archiving is critical, and Amazon Glacier makes the data available for future use. The data in the Amazon S3 bucket is in JSON format, which has been transformed using Amazon Glue. The data is then evaluated by performing queries on Amazon Athena.

As shown in the Figure 5, the primary data collected from the newly installed sensors, has 20 dimensions with 220320 records. The target variable is the machine status which holds records of three types, normal, recovering and broken. The 205836 records in the normal functioning class, 14477 records in the recovering class as shown in Figure 6, and seven records in the broken class as shown in Figure 7. The broken state is followed by the recovering state and hence the recovering state is also considered as the broken state. The machine learning algorithms will lose its predictive power with such a high number of dimensions. This state is generally called as the Curse of Dimensionality. When analyzing the dataset, it was found that many features were captured multiple times for security purpose. This resulted in having duplicate records contributing noise in the dataset. The correlation between the features must be analyzed, features with less correlation can be eliminated. The feature space is reduced using a dimensionality reduction approach.

Figure 5

Raw Dataset With Machine Status as Normal



	Unnamed: 0	timestamp	water_quality	vibration (Hz)	chlorine residue	pH-level,	Ion monitoring	vapour	discharge_pressure (psi)	suction pressure (psi)	suspended solids	flow	velocity (rpm)	power	temperature	heat	motor_current (volts)	water quality	vibration_senor(Hz)	machine_status
0	0	2018-04-01 00:00:00	8.0	634.375000	13.41146	16.13136	15.56713	15.05353	37.22740	47.52422	NaN	2.565284	975.9409	684.9443	90.32386	90.32386	67.70834	8.0	634.375000	NORMAL
1	1	2018-04-01 00:01:00	8.0	634.375000	13.41146	16.13136	15.56713	15.05353	37.22740	47.52422	NaN	2.565284	975.9409	684.9443	90.32386	90.32386	67.70834	8.0	634.375000	NORMAL
2	2	2018-04-01 00:02:00	9.0	638.888900	13.32465	16.03733	15.61777	15.01013	37.86777	48.17723	NaN	2.500062	982.7342	715.6266	93.90508	93.90508	67.12963	9.0	638.888900	NORMAL
3	3	2018-04-01 00:03:00	7.0	628.125000	13.31742	16.24711	15.69734	15.08247	38.57977	48.65607	NaN	2.509521	977.7520	690.4011	101.04060	101.04060	66.84028	7.0	628.125000	NORMAL
4	4	2018-04-01 00:04:00	7.0	636.458300	13.35359	16.21094	15.69734	15.08247	39.48939	49.06298	NaN	2.604785	979.5755	704.6937	101.70380	101.70380	66.55093	7.0	636.458300	NORMAL
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
220315	220315	2018-08-31 23:55:00	9.0	634.722229	15.11863	16.65220	15.65393	15.16204	43.17085	54.16052	NaN	2.499117	1109.5010	485.0358	0.00000	0.00000	153.64580	9.0	634.722229	NORMAL
220316	220316	2018-08-31 23:56:00	6.0	630.902771	15.15480	16.70284	15.65393	15.11863	43.21038	54.52602	NaN	2.618476	1106.3710	510.9510	0.00000	0.00000	156.25000	6.0	630.902771	NORMAL
220317	220317	2018-08-31 23:57:00	6.0	625.925903	15.08970	16.70284	15.69734	15.11863	43.12836	55.11779	NaN	2.620500	1106.6980	492.7720	0.00000	0.00000	155.38190	6.0	625.925903	NORMAL
220318	220318	2018-08-31 23:58:00	6.0	635.648100	15.11863	16.56539	15.74074	15.11863	42.35746	55.99321	NaN	2.514596	1103.9550	490.2170	0.00000	0.00000	153.93520	6.0	635.648100	NORMAL
220319	220319	2018-08-31 23:59:00	8.0	639.814800	15.11863	16.65220	15.65393	15.01013	42.62814	56.49642	NaN	2.487299	1108.8270	496.4068	0.00000	0.00000	150.46300	8.0	639.814800	NORMAL

220320 rows × 20 columns

Note. Dataset with target feature value NORMAL.

Figure 6

Raw Dataset With Machine Status as Recovering

	Unnamed: 0	timestamp	water_quality	vibration (Hz)	chlorine residue	pH-level,	Ion monitoring	vapour	discharge_pressure (psi)	suction pressure (psi)	suspended solids	flow	velocity (rpm)	power	temperature	heat	motor_current (volts)	water quality	vibration_senor(Hz)	machine_status
17156	17156	2018-04-12 21:56:00	9.0	204.725098	3.045428	17.42621	15.740740	16.17477	40.310710	3.730241	NaN	2.448104	951.3060	773.6210	110.52630	110.52630	133.1019	9.0	204.725098	RECOVERING
17157	17157	2018-04-12 21:57:00	8.0	201.137131	7.537616	13.53443	9.324364	16.05179	38.930980	3.816472	NaN	2.410462	948.8156	783.5280	115.34480	115.34480	140.0463	8.0	201.137131	RECOVERING
17158	17158	2018-04-12 21:58:00	9.0	204.030655	7.609953	16.60880	16.203700	16.09520	33.433750	3.860711	NaN	2.429593	950.7607	787.4129	105.04380	105.04380	144.9653	9.0	204.030655	RECOVERING
17159	17159	2018-04-12 21:59:00	9.0	203.567688	7.573785	16.70284	16.160300	16.08796	33.132260	4.496508	NaN	2.361692	949.6810	782.2913	113.03400	113.03400	149.0162	9.0	203.567688	RECOVERING
17160	17160	2018-04-12 22:00:00	9.0	203.567688	7.559317	16.56539	16.239870	16.05179	35.345990	12.028980	NaN	2.311379	950.6841	780.6435	119.86980	119.86980	162.6157	9.0	203.567688	RECOVERING
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
166511	166511	2018-07-25 15:11:00	6.0	152.526031	15.089700	17.62876	18.200230	16.39902	4.347953	0.493511	NaN	2.655957	1088.5370	598.6487	68.15515	68.15515	260.4167	6.0	152.526031	RECOVERING
166512	166512	2018-07-25 15:12:00	6.0	152.178802	15.089700	17.62153	18.243630	16.31944	5.155131	0.008017	NaN	2.500874	1091.2670	563.4313	75.91868	75.91868	244.2130	6.0	152.178802	RECOVERING
166513	166513	2018-07-25 15:13:00	7.0	151.021393	15.125870	17.62153	18.236400	16.35561	5.430405	0.002175	NaN	2.508361	1087.3970	576.0786	64.93620	64.93620	229.1667	7.0	151.021393	RECOVERING
166514	166514	2018-07-25 15:14:00	6.0	152.988983	15.089700	17.66493	18.301500	16.39902	5.286151	0.019133	NaN	2.461950	1092.5990	537.6109	56.60923	56.60923	216.7245	6.0	152.988983	RECOVERING
166515	166515	2018-07-25 15:15:00	9.0	151.252900	15.118630	17.67940	18.127890	16.39902	5.363589	0.000000	NaN	2.699134	1086.3240	563.6975	51.73800	51.73800	211.2269	9.0	151.252900	RECOVERING

14477 rows × 20 columns

Note. Dataset with target feature value RECOVERING.

Figure 7

Raw Dataset With Machine Status as Broken

	Unnamed: 0	timestamp	water_quality	vibration (Hz)	chlorine residue	pH-level,	Ion monitoring	vapour	discharge_pressure (psi)	suction pressure (psi)	suspended solids	flow	velocity (rpm)	power	temperature	heat	motor_current (volts)	water quality	vibration_senor(Hz)	machine_status
17155	17155	2018-04-12 21:55:00	8.000000	202.526031	3.219039	16.890910	16.869210	15.082470	35.530850	3.625588	NaN	2.387357	955.74270	783.6125	114.20790	114.20790	121.527800	8.000000	202.526031	BROKEN
24510	24510	2018-04-18 00:30:00	7.000000	206.038757	12.304690	15.154800	14.185470	13.867190	28.304880	30.434710	NaN	2.481055	944.30770	784.2626	73.31284	73.31284	44.560180	7.000000	206.038757	BROKEN
69318	69318	2018-05-19 03:18:00	NaN	200.115738	13.592300	15.914350	15.147570	14.793110	43.998860	43.623220	NaN	2.539193	997.63640	761.1199	75.20248	75.20248	49.768520	NaN	200.115738	BROKEN
77790	77790	2018-05-25 00:30:00	2.448669	612.152800	14.062500	16.608800	15.943290	15.596060	27.092980	44.793620	NaN	2.557975	982.50690	724.8124	72.70645	72.70645	65.682870	2.448669	612.152800	BROKEN
128040	128040	2018-06-28 22:00:00	8.000000	201.368622	11.335360	15.270540	15.183740	15.118630	2.002474	1.960537	NaN	0.444666	96.38937	109.4662	23.78439	23.78439	29.513889	8.000000	201.368622	BROKEN
141131	141131	2018-07-08 00:11:00	7.000000	500.000000	0.028935	0.036169	0.036169	0.007234	25.977650	23.970270	NaN	2.632338	1091.49300	562.6080	30.04753	30.04753	44.849540	7.000000	500.000000	BROKEN
166440	166440	2018-07-25 14:00:00	7.000000	420.503448	14.185470	16.247110	15.697340	15.053530	36.717480	50.046190	NaN	2.468381	1093.57900	551.2756	44.77601	44.77601	69.733800	7.000000	420.503448	BROKEN

Note. Dataset with target feature value BROKEN.

3.3 Data Pre-processing

The data pre-processing is paramount, as the output of the model dependence upon the correctness of the data. The quality of data is dependent on the accuracy, completeness, consistency, uniformity, interpretability, and timeliness. The sensor functionality determines the accuracy and completeness of the data for this problem domain. If the sensors malfunction it may result in having a value beyond the specific range or sometimes no data. The data from the sensors are collected and aggregated by the IoT gateway, the aggregation must take place in a well-defined manner so that the data is consistent. The timely capturing of data is crucial because the machine must be monitored every second to see any possible abnormality.

In the data pre-processing stage, a series of steps are involved. The data collected from the source is dirty in general, due to many reasons. The data is first analyzed for the missing values. It was found that water_quality, vibration, chlorine residue, pH-level, Ion monitoring, vapor, suspended solids have missing values. Now the missing value percentage is calculated for each feature, based on the percentage the decision is made to drop or impute the feature value. The suspended solids have a 100% missing data; this feature is removed from the dataset. The

vibration feature has 21% of missing data, this feature is noted in the data quality plan, only if it affects the result of the model, it will be imputed. All the other features have less than 5% of missing data which must be imputed by using suitable techniques. Imputation is a method of predicting missing values with the help of a model. The SimpleImputer function from scikit learn is used to impute these features. This function will take the median, mean and mode values for imputation (*Sklearn.Impute.SimpleImputer*, n.d.). The K Nearest Neighbor (kNN) imputation algorithm for predicting the missing values of the vibration feature. The column which captures the vibration of the water pump machine is filled with NaN value. This must be replaced with an appropriate value as this feature is important for our model to predict the future failures. The KNNImputer function from scikit-learn is used to perform kNN imputation (*Sklearn.Impute.KNNImputer*, n.d.). The preprocessed data using these techniques is shown in the Figure 8. In the kNN algorithm, the k value indicates the number of neighbors that the algorithm should consider predicting the missing value. Based on the k value the distance between the missing value and the nearest points are calculated, generally the Euclidian distance is used. The vibration values of the nearest neighbors are aggregated, and the mean value will be predicted for the missing field.

Figure 8

Missing Values Replaced by Using SimpleImputer and kNNimputer

	Unnamed: 0	timestamp	water_quality	vibration (Hz)	chlorine residue	pH-level,	Ion monitoring	vapour	discharge_pressure (psi)	suction pressure (psi)	flow (gpm)	velocity (rpm)	power (watts)	t
220315	220315	2018-08-31 23:55:00	9.0	634.722229	15.11863	16.65220	15.65393	15.16204	43.17085	54.16052	2.499117	1109.501	485.0358	
220316	220316	2018-08-31 23:56:00	6.0	630.902771	15.15480	16.70284	15.65393	15.11863	43.21038	54.52602	2.618476	1106.371	510.9510	
220317	220317	2018-08-31 23:57:00	6.0	625.925903	15.08970	16.70284	15.69734	15.11863	43.12836	55.11779	2.620500	1106.698	492.7720	
220318	220318	2018-08-31 23:58:00	6.0	635.648100	15.11863	16.56539	15.74074	15.11863	42.35746	55.99321	2.514596	1103.955	490.2170	
220319	220319	2018-08-31 23:59:00	8.0	639.814800	15.11863	16.65220	15.65393	15.01013	42.62814	56.49642	2.487299	1108.827	496.4068	

Note. Dataset after adopting the imputation approaches.

Figure 9

List of Features With no Missing Values

```
water_quality               0.0
vibration (Hz)              0.0
chlorine residue            0.0
pH-level,                   0.0
Ion monitoring              0.0
vapour                      0.0
discharge_pressure (psi)    0.0
suction pressure (psi)      0.0
flow (gpm)                  0.0
velocity (rpm)              0.0
power (watts)               0.0
temperature (F)             0.0
heat                        0.0
motor_current (volts)       0.0
water quality               0.0
vibration_senor(Hz)         0.0
machine_status              0.0
dtype: float64
```

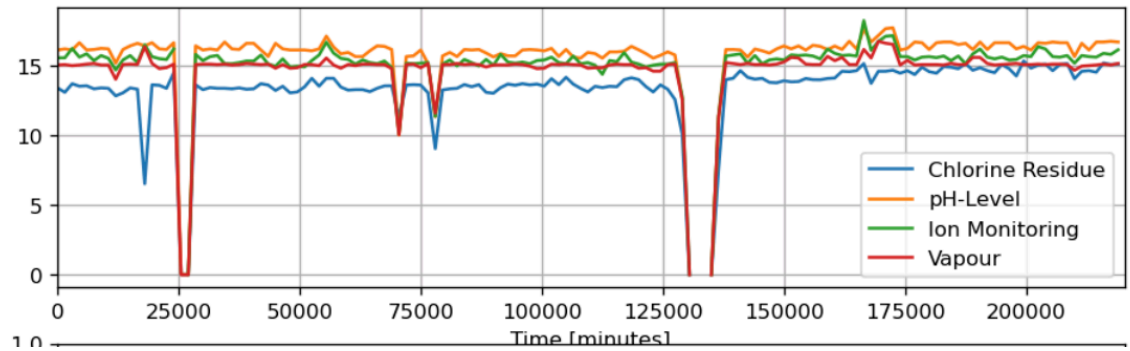
Note. Clean dataset with no missing values.

There are no missing values in the dataset as shown in the Figure 9. To understand abo ut noisy data, a trend graph is plotted for all the features in the dataset. As shown in the Figure 10 , the chlorine residue, pH-level, Ion monitoring, and vapor follows no definite pattern and has a n oisy data. The heat and temperature columns record the same value. Similarly, the water quality data is captured twice which contributes to the noise in data due to duplication. Sometimes dupli

cation of records is for the security purpose. The duplicated data must be removed using the pyth on function.

Figure 10

Features Which Have no Definite Pattern



Note. Features with no definite pattern.

Data integration is the next level of pre-processing; for the water pump failure prediction problem, data from the sensors is collected every day. The data collected on a given day will be combined into a batch the following day. To preserve data continuity, the prior days data and the current data must be blended when new data is streamed through Amazon Kinesis. The data continuity must be validated before it is sent to the model for prediction.

In the data transformation phase, as every feature of the water pump machine is measured in different units, normalizing the data, and aligning to one common scale makes the model very stable. In the data reduction phase, to address the curse of dimensionality problem univariate selection method is used. In this method the feature which has the strong relationship with the target variable will be selected.

3.4 Data Transformation

In the data transformation phase, the daily streaming data must be merged with the previous days data, this is achieved by using Amazon Lambda Architecture. The data collected

by the IoT gateway from the sensors is transferred to the Amazon cloud via Wi-Fi routers. The data from these routers is sent to Amazon Kinesis Data Stream, which is ultimately sent to an Amazon S3 bucket. The batch layer consists of Amazon S3 bucket and Amazon Glue performs all the heavy computation and analysis is done in this batch layer for time-series dataset including the accuracy check and it creates a batch view, in the speed layer the steaming data is obtained through the Amazon Kinesis Data Stream is pushed into the Amazon S3 bucket, this layer creates a real-time view, merged with the batch view in the serving layer by running the merge query and stored in the Amazon S3 bucket for further consumption (*Reference architecture - analytics lens - docs.aws.amazon.com,*n.d.). The incoming data can be of any format namely an XML, JSON or a text file. The Amazon Glue service crawls the data and coverts it into the desirable table format and then creates the real-time view.

Transforming the data is a prime phase, as the accuracy of the model depends upon the correctness of the features. A good prediction model will rely on the most appropriate and few features. The curse of dimensionality is a serious issue in which a dataset has many features, and if all of them are supplied to the model, it will be confused, and its accuracy will suffer. Also, it makes more difficult to visualize the data to detect any patterns, and the model may take longer to train and deliver poor results. This problem must be handled using appropriate techniques. The feature selection is achieved by using the SelectKBest class from the scikit-learn to implement univariate selection, the features are selected by Chi-squared test, takes the descriptive and target values to see the correlation, the feature which gets the high score will be selected by the model (*Sklearn.Feature_selection.SelectKBest*, n.d.).This also reduces overfitting. For our problem the SelectKBest class returned the following features water quality, vibration, discharge pressure, suction pressure, flow, power, and temperature as shown in the Figure 11. Hence, our dataset is

now reduced with the most desirable feature which shows a strong relationship with the target feature.

Figure 11

Dataset After Performing Dimensionality Reduction

	timestamp	suction pressure (psi)	discharge_pressure (psi)	flow (gpm)	vibration_senor(Hz)	temperature (F)	power (watts)	water quality	machine_status
0	2018-04-01 00:00:00	47.52422	37.22740	2.565284	634.375000	90.32386	684.9443	8.0	NORMAL
1	2018-04-01 00:01:00	47.52422	37.22740	2.565284	634.375000	90.32386	684.9443	8.0	NORMAL
2	2018-04-01 00:02:00	48.17723	37.86777	2.500062	638.888900	93.90508	715.6266	9.0	NORMAL
3	2018-04-01 00:03:00	48.65607	38.57977	2.509521	628.125000	101.04060	690.4011	7.0	NORMAL
4	2018-04-01 00:04:00	49.06298	39.48939	2.604785	636.458300	101.70380	704.6937	7.0	NORMAL
...	...	...	...	...	...	...	...	...	...
220315	2018-08-31 23:55:00	54.16052	43.17085	2.499117	634.722229	0.00000	485.0358	9.0	NORMAL
220316	2018-08-31 23:56:00	54.52602	43.21038	2.618476	630.902771	0.00000	510.9510	6.0	NORMAL
220317	2018-08-31 23:57:00	55.11779	43.12836	2.620500	625.925903	0.00000	492.7720	6.0	NORMAL
220318	2018-08-31 23:58:00	55.99321	42.35746	2.514596	635.648100	0.00000	490.2170	6.0	NORMAL
220319	2018-08-31 23:59:00	56.49642	42.62814	2.487299	639.814800	0.00000	496.4068	8.0	NORMAL

220320 rows × 9 columns

Note. Features which show a strong relationship with target variable (machine status).

In the preprocessing stage, the null values and all the inconsistencies in the data are removed using different techniques. The most appropriate features to the machine model, then better the prediction results. Hence, the most contributing feature for the prediction is selected and given to the model. Now, after the features are reduced, there seems to be a different unit of measurement for each feature. The model will not understand these units and hence there much be a mechanism to convert these values to a common scale. The normalization technique is used to convert all the data points to a common scale. The data is normalized to a scale between zero and one, where zero being the lower limit and one the upper limit. To normalize the data, preprocessing Normalizer function from the scikit-learn library is used. The preprocessing Normalizer would convert each row to unit norm (*Sklearn.preprocessing.Normalizer*, n.d). The preprocessing Normalizer library is imported, and the dataset is then transposed to match the function. The dataset will be scaled using the upper and lower boundaries. After reverting to its

original form, the dataset is allocated to the same variable. All the values are in the range of zero to one, and there is a common measurement scale as shown in the Figure 12.

Figure 12

Normalizing the Dataset Using Preprocessing Normalizer

```
normalizer_scaler = preprocessing.Normalizer(norm='max')
X = normalizer_scaler.fit_transform(X.transpose())
X = pd.DataFrame(X.transpose())
X.head()
```

	0	1	2	3	4	5	6
0	0.792070	0.489146	0.526401	0.792969	0.516428	0.467130	0.888889
1	0.792070	0.489146	0.526401	0.792969	0.516428	0.467130	0.888889
2	0.802954	0.497561	0.513017	0.798611	0.536904	0.488056	1.000000
3	0.810935	0.506916	0.514958	0.785156	0.577701	0.470852	0.777778
4	0.817716	0.518868	0.534507	0.795573	0.581493	0.480599	0.777778

```
X.shape
```

```
(220320, 7)
```

Note. Features normalized to a common scale.

Now the target variable, machine_status records the three states of the machine normal, recovering and broken. During the data analysis phase, it was seen that the recovering state is always followed by the broken state. So, the recovering state is also considered to be the broken state. The datatype of this feature is string. But most of the machine models would require all its data to be a numeric value. Therefore, the target variable which is a categorical feature must be converted into a numeric value. The popular technique, one-hot encoding is used to encode these values to a binary value, where zero represents the broken and recovery state and one represents the normal state as shown in the Figure 13.

Figure 13

Converting the Target Variable to Binary Values

```
y = sensors['machine_status']
one_hot = pd.get_dummies(y)
one_hot.head()
```

	BROKEN	NORMAL	RECOVERING
0	0	1	0
1	0	1	0
2	0	1	0
3	0	1	0
4	0	1	0

```
one_hot.shape
```

```
(220320, 3)
```

Note. The machine status is encoded to zero and one.

3.5 Data Preparation

The data has now been cleansed and translated into a format that the machine learning model can comprehend. The modified data must be separated into three sets during the data preparation phase: test, train, and validation. The model built for the water pump failure prediction problem must trained with some sample dataset. As shown in the Figure 14, to train the model, the bulk of the dataset is used as the training set. The target variable in our problem domain is labeled and has three states: normal, recovered, and broken. This problem is categorized as a supervised binary classification problem. When new data is supplied to the model, it is trained with these labels and produces predictions based on the learnings from the training phase. The dataset taken for this analysis consists of 220320 records, in which 70% of the data points are split as the training dataset.

Figure 14

The Training Dataset

X_train

	0	1	2	3	4	5	6
0	0.792070	0.489146	0.526401	0.792969	0.516428	0.467130	0.888889
1	0.792070	0.489146	0.526401	0.792969	0.516428	0.467130	0.888889
2	0.802954	0.497561	0.513017	0.798611	0.536904	0.488056	1.000000
3	0.810935	0.506916	0.514958	0.785156	0.577701	0.470852	0.777778
4	0.817716	0.518868	0.534507	0.795573	0.581493	0.480599	0.777778
...	...	...	...	...	...	...	...
154219	0.788494	0.589806	0.534633	0.793113	0.000000	0.400774	0.888889
154220	0.793368	0.603546	0.523840	0.804543	0.033996	0.382300	1.000000
154221	0.790138	0.601173	0.513251	0.784288	0.074508	0.368495	1.000000
154222	0.794764	0.607897	0.527033	0.786314	0.000000	0.380226	0.777778
154223	0.804393	0.614221	0.522152	0.789352	0.000000	0.363921	0.666667

154224 rows × 7 columns

y_train

	BROKEN	NORMAL	RECOVERING
0	0	1	0
1	0	1	0
2	0	1	0
3	0	1	0
4	0	1	0
...	...	...	...
154219	0	1	0
154220	0	1	0
154221	0	1	0
154222	0	1	0
154223	0	1	0

154224 rows × 3 columns

Note. 70% of the data is split for training the model.

The model is tested, and the accuracy is calculated before exposing the model to a data for which the target value is unknown. The remaining part of the dataset is taken as the testing data. The target value is not revealed to the model in this phase, unlike in the training phase. The model is put to the test by using the values for all the features and predicting the target value. As shown in Figure 15, the testing set represents 30% of the whole dataset, once the model has predicted the target value for all of the testing data, the model accuracy is determined by comparing the anticipated and actual target values.

Figure 15

The Testing Dataset

X_test

	0	1	2	3	4	5	6
154224	0.781739	0.620698	0.525919	0.801939	0.045848	0.383217	1.000000
154225	0.793956	0.624864	0.532775	0.793258	0.115492	0.399363	0.666667
154226	0.789188	0.623893	0.498008	0.789207	0.112261	0.387427	0.777778
154227	0.786440	0.624932	0.515549	0.787037	0.084472	0.371358	0.666667
154228	0.799709	0.625564	0.532747	0.797309	0.080587	0.382372	1.000000
...	...	...	...	...	...	...	...
220315	0.902675	0.567240	0.512823	0.793403	0.000000	0.330793	1.000000
220316	0.908767	0.567759	0.537316	0.788628	0.000000	0.348467	0.666667
220317	0.918630	0.566682	0.537731	0.782407	0.000000	0.336069	0.666667
220318	0.933220	0.556552	0.516000	0.794560	0.000000	0.334327	0.666667
220319	0.941607	0.560109	0.510398	0.799768	0.000000	0.338548	0.888889

66096 rows × 7 columns

y_test

	BROKEN	NORMAL	RECOVERING
154224	0	1	0
154225	0	1	0
154226	0	1	0
154227	0	1	0
154228	0	1	0
...	...	...	...
220315	0	1	0
220316	0	1	0
220317	0	1	0
220318	0	1	0
220319	0	1	0

66096 rows × 3 columns

Note. 30% of the data is split for testing the model

The scikit-learn library has a function, test train split which splits the dataset into the train, test, and validation datasets. This library divides the data into various sets, each of which performs a specific task: the training set trains the model, the validation dataset evaluates the performance, and the testing dataset tests the model accuracy before deployment. In the water pump machine failure prediction problem, time series data is recorded, in which each row is dependent on the previous row. There is no independent data as in the non-time series datasets. Hence this method cannot be used to split the data. Because the data is organized by dates, randomly separating it would result in a lot of distortion because the prediction is based on the prior days data. To maintain the continuity of the data, it is split into two sets the training and the test data where 70% training and the 30% testing.

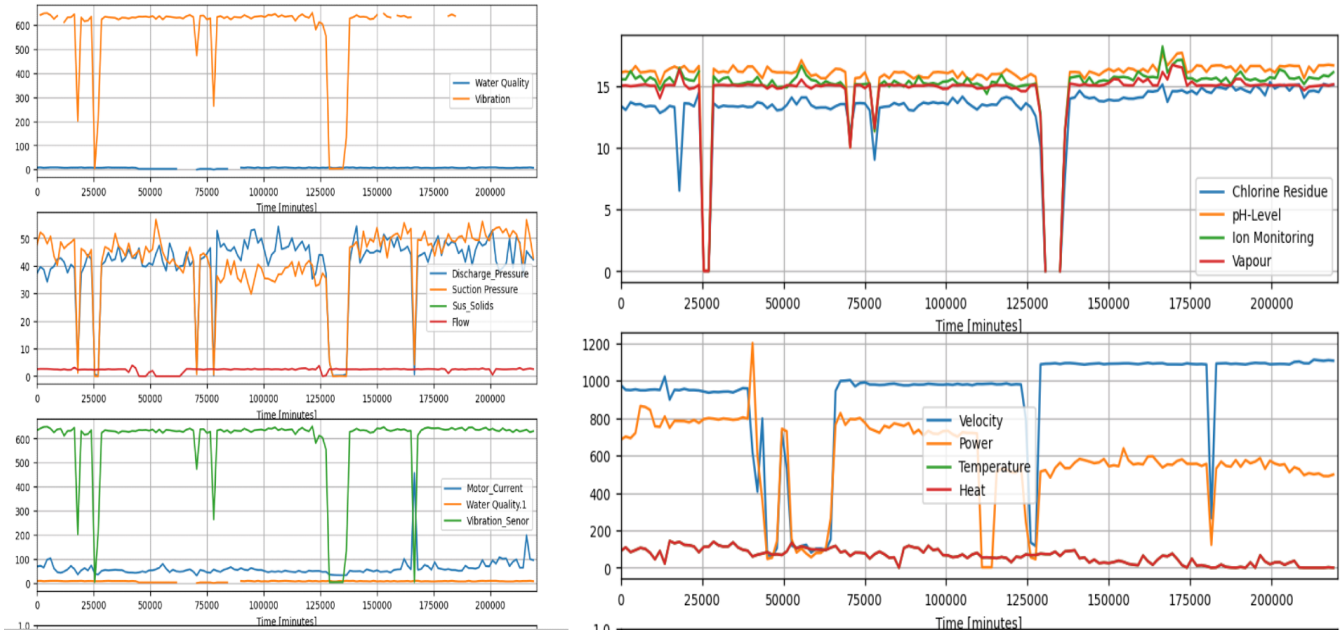
The validation dataset validates the model by testing the model with different sets of values in the dataset in a rotational basis. The dataset is divided into multiple segments by the k parameter in the K-Fold-Cross-Validation technique, all the segments are used to train the model except one which will be used to test the model, this is continued till the model is tested by all the segments (Shrivastava, 2020). The average results of all the folds measures the minimum and maximum accuracy of the model. The water pump machine has a time series data where the cross validation is carried out on a rolling basis. First, a small sample data is selected, and the model is trained using this data. The test data is used for predicting the future points. In the next iteration these predicted points will be added to the training data and the model is trained again. This process is repeated until all the data points are tested. The accuracy score for each iteration is recorded. Once all the iterations are complete, the average accuracy value is calculated. This value will measure the maximum and minimum accuracy and it ensures that the model is trained and tested with all the data in different angles. To perform the validation of the model TimeSeriesSplit function from the scikit-learn library is used. Then the testing data is utilized to test the model, and the confusion matrix, precision score, and F1 score are generated to ensure that the prediction is accurate.

3.6 Data Statistics

The data collected from the sensors for forecasting the problem with the water pump machine has gone through several stages of cleaning, transformation, and preparation. In each stage the dataset is visualized to analyze the patterns which will give more insights to the data and the problem. The visualization in one such tool which helps to identify patterns and relate the functioning of the machine. The data for our problem domain is a time series data and hence it is visualized by trend graphs.

Figure 16

Visualizing the Raw Dataset to Check for Definite Patterns

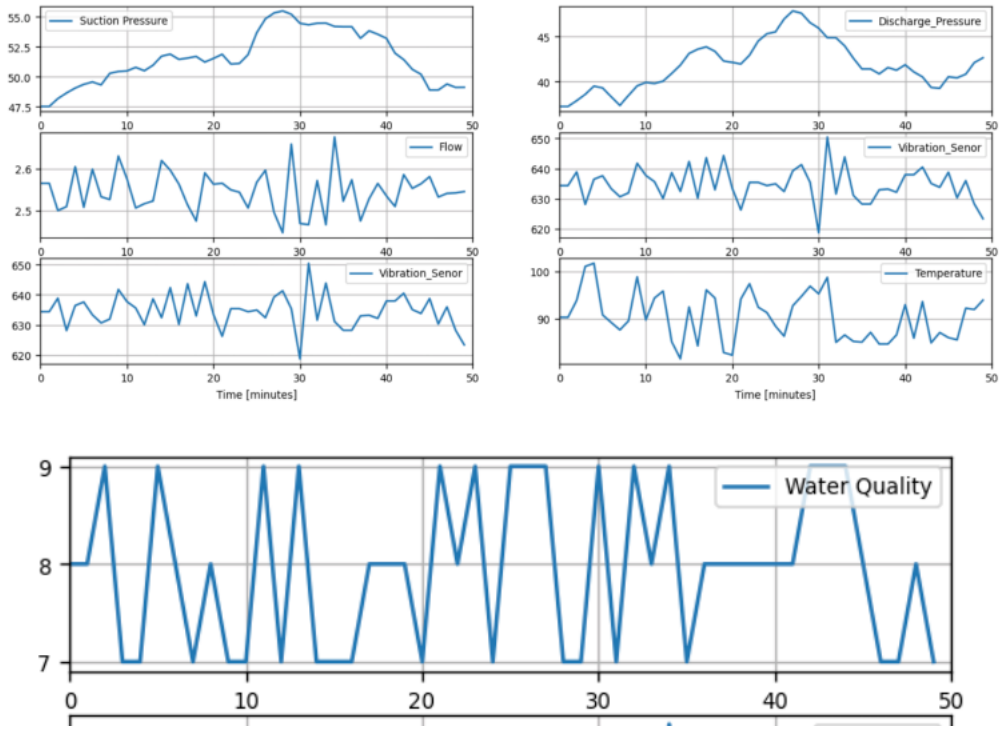


Note. Visualizing the raw data.

As shown in the Figure 16, the chlorine residue, pH-level, Ion monitoring, and vapor fo llows no definite pattern and has a noisy data. The heat and temperature columns record the same value. Similarly, the water quality data is captured twice which contributes to the noise in data du e to duplication. Sometimes duplication of records is for the security purpose. The duplicated dat a must be removed using the python function. The water quality, vibration, discharge pressure, s uction pressure, flow, power, and temperature shows a definite pattern and hence the correlation among these features is calculated to explore the relationship between them.

Figure 17

Univariate Analysis for Selected Feature



Note. Univariate analysis for discharge pressure, flow, vibration, and temperature.

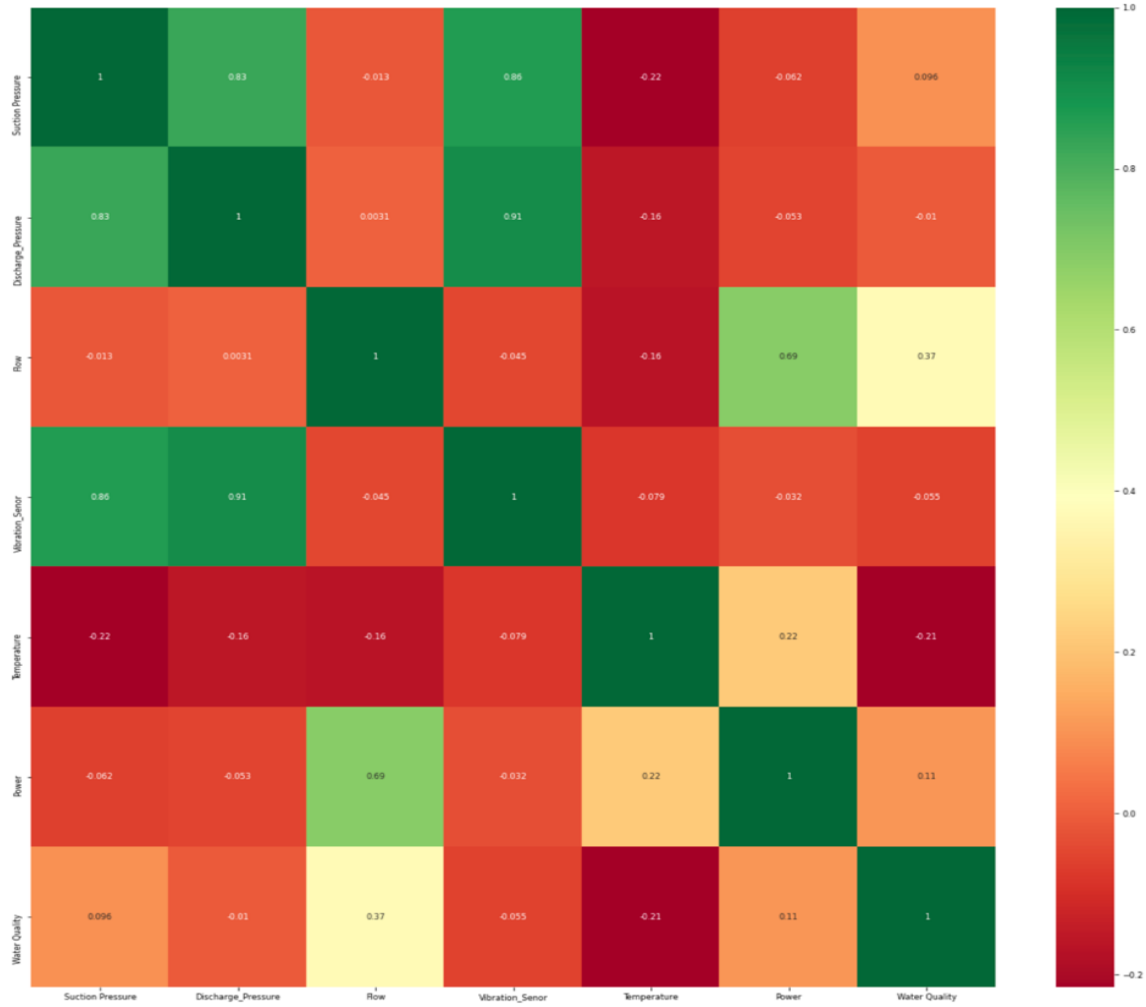
Univariate plots are used to examine individual attributes as shown in the Figure 17, the discharge pressure and suction pressure have a steady increase and fall relationship. If the machine ceases working normally, vibration of the pump suddenly decreases. The temperature rises and falls, and if the heat surpasses a particular threshold, the coil is continuously cooled. Because power fluctuates most of the time, a constant power regulator should be installed within the water pump machine. The water quality fluctuates between seven and nine points, indicating that the water is portable and so safe to drink. The suction and discharge pressures, as well as the power, influence the flow.

The pre-processed data consists of 220320 rows and seven columns. The relationship between these columns can be visualized using a heat map as shown in the Figure 18, the

discharge pressure and vibration have a high correlation, similarly the suction pressure and vibration have a good correlation. The suction pressure and discharge pressure have a very good relationship. Temperature and power have a moderate relationship. The suction pressure and the temperature have no relationship.

Figure 18

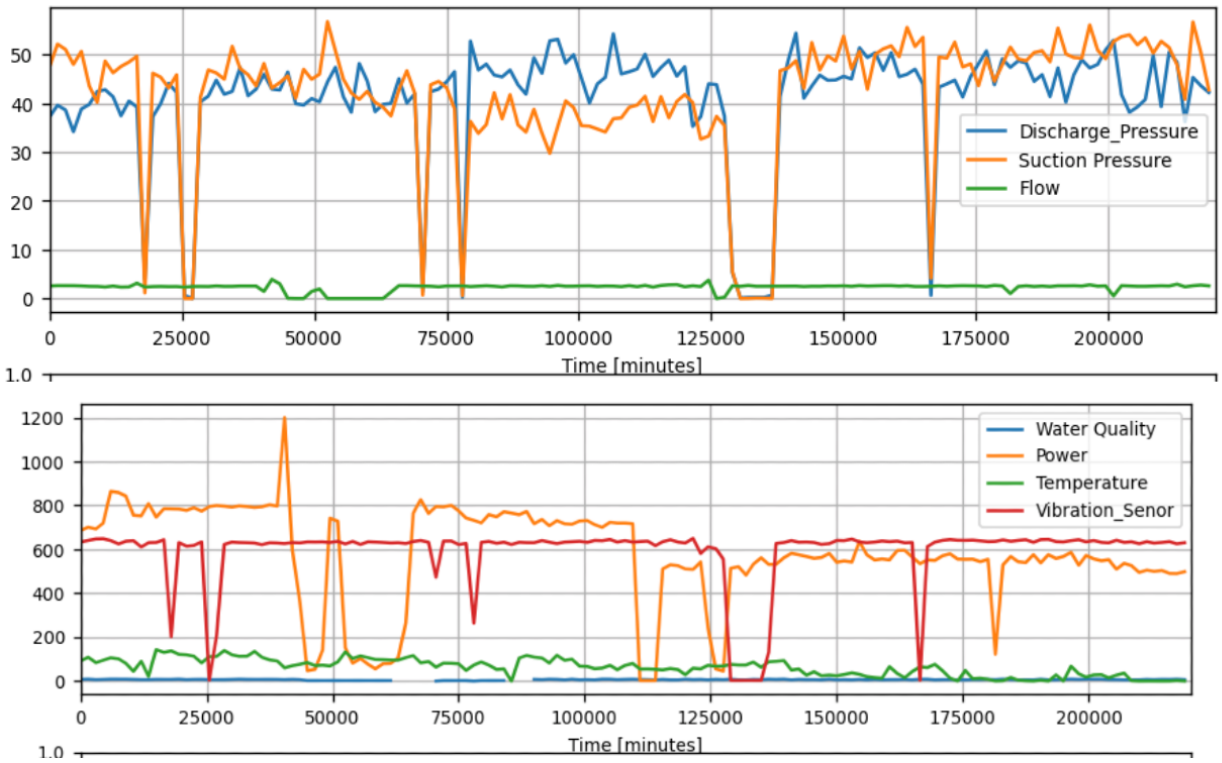
Heatmap for the Selected Feature to Check the Correlation



Note. Heatmap showing the correlation between the features

Figure 19

Visualization of Prepared Dataset

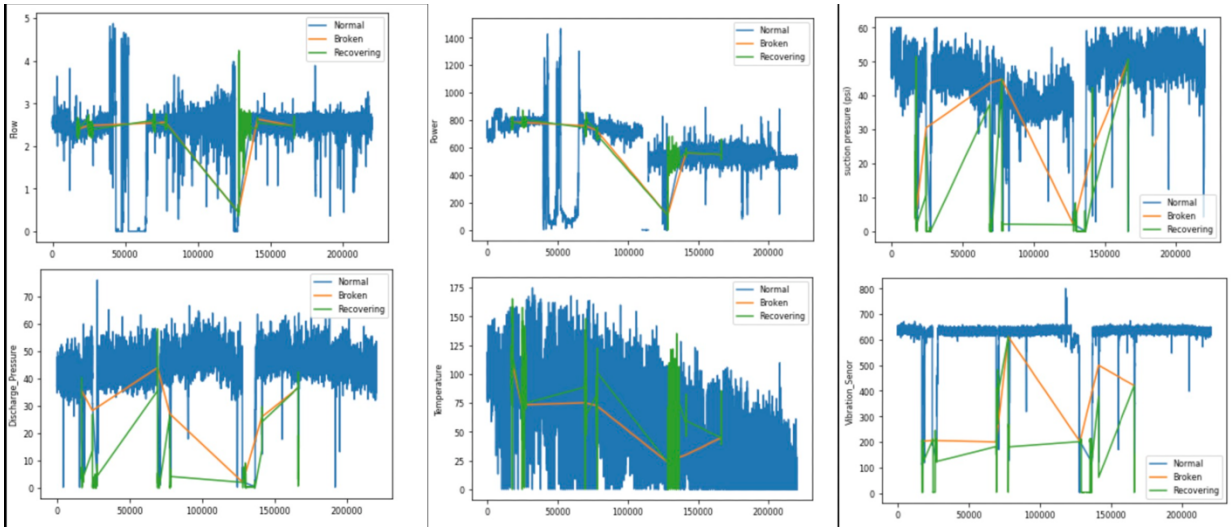


Note. The highly corelated features after data cleaning and dimensionality reduction.

As shown in the Figure 19, the water pump discharges water at a slightly lower pressure than the suction pressure. The coolant fan inside the motor keeps the temperature of the motor coil at a normal level even when the power is raised. Regardless of the fluctuations in pressure, the water flow stays constant. When the power is reduced, the vibration of the pump is reduced as well. The water quality is consistent, thus the water from this pump is always portable.

Figure 20

Visualizing the Three States of Water Pump Machine



Note. The three states normal, broken and recovering for all the selected features.

The normal condition is represented by blue, while the broken and recovered states are represented by orange and green, respectively as shown in the Figure 20. The broken state is followed by the recovering state and hence the recovering state is also considered as the broken state. The graph depicting the feature flow demonstrates that the machine fails when the flow exceeds 3 gallons. Because of the fluctuation, the machine stops working when the power dips from a high voltage to a lower value. The suction pressure is the most crucial feature because water cannot enter the pump if the pressure is too low. Most of the failures happened when the suction pressure was low. Similarly, the water should be discharged at the same pressure; if the pressure drops suddenly in the middle, the water will not be able to escape from the pump. When the coil inside the motor reaches a temperature of more than 125 ^{0}F, the motor enters the recovery mode. The machine resumes normal operation once the coil has cooled. When the machine vibration is modest, the water pump is not discharging any water.

4. Model Development

4.1 Model Proposals

A model is built by a set of algorithms, trained by an enormous amount of data from which it identifies a certain pattern and makes prediction when it is exposed to a new data. The main goal of the machine learning technique is to make the computer learn by itself without human intervention and provide a good decision based on its learnings. Machine learning provides numerous algorithms for various types of data, which learns from training data and predicts based on patterns formed throughout the learning process utilizing artificial intelligence. The models are provided with data and the output, from which a program is built internally by the model. The data has two parts the feature and the target label. The feature describes about the parameters associated to the problem domain and the target is the output which the model is expected to predict.

Based on the inputs given to the machine learning model the algorithms are grouped as supervised and unsupervised learning models. The supervised technique, model is trained with the feature and the target label, on the other hand in the unsupervised technique model is trained on the features without the target labels, it finds the hidden pattern and groups the data into clusters. The supervised learning is further classified as a regression problem in which the target value is a continuous feature and a classification problem where the target is a categorical feature. In the regression problem a value will be predicted from observation. Whereas in the classification algorithm the data is assigned to different classes. In the binary classification the target label must have only two categories and in the multiclass classification the target will have more than two categories (Johnson, 2021).

To anticipate the failure of a water pump machine, we collect characteristics that describe each item that contributes to its appropriate operation. The features are scrutinized thoroughly, and the most correlated feature is given to learn the pattern. As the data acquired in the water pump problem comprises a set of features and a target label, this problem is classified as supervised machine learning. In addition, there are two outputs for the target variable: normal and broken. The model is trained using the features and target variable, and it is supposed to properly predict if the machine is on the edge of failure by classifying fresh machine data into one of two states.

Hence, our problem is a supervised binary classification which will classify the machine status in one of the two classes: normal or broken, evaluated by motoring the machine every second. For water pump failure prediction problem, the model is trained by the historical data collected from the sensors. The target variable is determined by measuring the values of the features crossing a particular threshold, which indicates that something is amiss, and the equipment is about to fail. This is recorded as the broken state of the machine. Hence, further investigation is made on the features contributing to this failure. The decision we make is inextricably linked to the target label. Every feature has a threshold, and if that value is exceeded, it will be notified to the field manager, which aids in the resolution of the problem before the machine is dead. Based on the classification task, the following machine learning models are proposed namely LR, LTSM, ARIMA, and XGBoost.

To solve the problem, the best model must be chosen to obtain accurate predictions. To choose the best model, we must run the prediction on several distinct models, compare the accuracy scores, and choose the best model. The initial stage in modeling is to choose the

simplest model to provide a suitable baseline for the problem, then go to more intricate models if they provide better accuracy.

The LR is one of the simplest binary classification models. The LR model is used if the two classes are linearly separable. The best fit line is calculated using the linear regression hypothesis, which separates the two classes on either side of the plane. As seen in Equation 1 the linear regression hypothesis is calculated where the θ^T is the slope and intercept, and the x is the point above or below the best fit line (*06: Logistic regression,* n.d.).

$$h_\theta(x) = (\theta^T\, x) \tag{1}$$

Firstly, in the LR model, the data points are plotted in the feature space. The best fit line is calculated using the above formula and the points are separated. Now we can determine the distance between the data point and the regression line. If the distance is positive then the data point is correctly classified, on the other hand if the distance is negative then the data is incorrectly classified. To get the best fit line the summation of all the points along with the distance are calculated and the maximum sum decides the best fit line. Suppose if there is an outlier then the value of the best fit line will be either greater than one or less than zero. This may greatly affect the model performance and the summation would result in a negative value. To draw the best fit line in the logistic regression model a cost function called sigmoid is used. This function would remove the effect of outliers in the model (*06: Logistic regression,* n.d.). The logistic regression hypothesis is given by Equation 2 where g is the sigmoid function, θ^T is the slope and intercept and x is the point above or below the best fit line (*Fisseha Berhane, Phd,* n.d.). The maximum summation of the distance between the points and the line are calculated in Equation 3 where the z is the real number and then this value is substituted in Equation 4 which is called the logistic function or the sigmoid function (*06: Logistic regression,* n.d.).

$$h_\theta(x) = g((\theta^T x)) \quad (2)$$

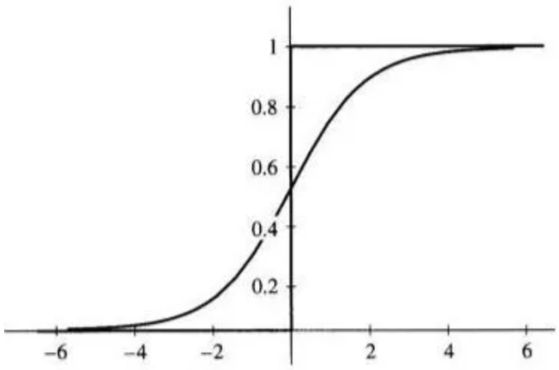
$$g(z) = 1/(1 + e^{-z})) \quad (3)$$

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1} + e^{-}\theta^T x \quad (4)$$

The sigmoid function will transform the distance between the point and the best fit line between zero to one. The predicted probabilities are mapped to values between zero and one and hence it nullifies the effect of outliers. The sigmoid function is denoted by an S shaped curve, where the value lies between zero and one. As shown in the Figure 21, the S shaped curve is generated by the Sigmoid function with threshold 0.5 (Sarkar et al., 2019).

Figure 21

Sigmoid Function S Shaped Curve



Note. "Machine learning: What is logistic regression?", by Sarkar, P., Roy, G. K., Sugandhi, A., & Vora, D. D., 2019. Knowledgehut. https://www.knowledgehut.com/blog/data-science/logistic-regression-for-machine-learning. Copyright 2021 by Knowledgehut.

If the anticipated output probability is larger than a specific threshold value, such as 0.5 in the image above, the predicted probability is categorized as class one, and if it is less than the threshold, it is classified as class zero. To minimize the error in the prediction, the cost function is used in the LR model.

In logistic regression y can take only two value either zero or one. Suppose for one observation if the value of y is equal to zero, and if the predicted value is close to zero, the log of the value will be close to one. Hence the error will be less. If the predicted value is one which is far from the actual value, then the log of the predicted value will be equal to zero and hence the error rate will be high. Similarly, when the y value is one, and if the predicted value is one which is equal to the actual value hence the error will be less, on the other hand if the predicted value is far from the actual value, then the error is high (Sarkar et al., 2019).

Figure 22

Pseudocode for LR Model

```
Step1: Function grad (predictor_attributes, target_attribute, weights)
            {
               Calculate gradient_descent;
               Return weights + learning_rate * gradient_descent;
            }
Step2: Normalize the dataset;
Step3: Repeat
    {
      Weights = grad (params);
      Update weights;
    } until convergence
Step4: z = dot product of predictor variables and updated weights;
Step5: prediction_limit = sigmoid function (z);
Step6: Predict the target class
```

Note. "An intelligent and energy-efficient wireless body area network to control coronavirus outbreak", by Bilandi, N., Verma, H. K., & Dhir, R., 2021. *Arabian Journal for Science and Engineering*, 46(9), 8203–8222. https://doi.org/10.1007/s13369-021-05411-2. Copyright 2021 by King Fahd University of Petroleum & Minerals.

The pseudocode for the LR algorithm is shown in the Figure 22. In our problem domain, the dataset consists of several feature which contributes to the functioning of the water pump machine measured in different units are normalized by the algorithm if the dataset is not normalized. The machine status normal and broken indicated by one and zero respectively are

plotted in the feature space. The weight is calculated by using the value of the predicted attribute and the target attribute. This process is continued until the best-fitting line is found. Once the best fit line is determined the value of the predictor variable and the weights are multiplied, this value is given to the sigmoid function to predict the target class (Bilandi et al., 2021).

The next model, XGBoost algorithm is considered for our problem domain. The XGBoost is a boosting ensemble technique which creates multiple models, learn from the mistake of the previous model and the final model will have a very less error. The XGBoost algorithm is a variation of the gradient boosting technique that considers the expected value as well as the residue. The learning in the gradient boosting algorithm happens by optimizing the loss. The XGBoost is a very popular algorithm as it aids in improved performance and accuracy (Brownlee, 2021).

Figure 23

Pseudocode for XGBoost Model

Input: I, instance set of current node
Input: d, feature dimension
$gain \leftarrow 0$
$G \leftarrow \sum_{i \in I} g_i$, $H \leftarrow \sum_{i \in I} h_i$
for $k = 1$ ***to*** m **do**
  $G_L \leftarrow 0$, $H_L \leftarrow 0$
  for j *in sorted*$(I,$ *by* $\mathbf{x}_{jk})$ **do**
    $G_L \leftarrow G_L + g_j$, $H_L \leftarrow H_L + h_j$
    $G_R \leftarrow G - G_L$, $H_R \leftarrow H - H_L$
    $score \leftarrow \max(score, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda})$
  end
end
Output: Split with max score

*Note. "*A novel systematic and evolved approach based on XGBoost-Firefly algorithm to predict Young's modulus and unconfined compressive strength of Rock"*,* by Cao, J., Gao, J., Nikafshan Rad, H., Mohammed, A. S., Hasanipanah, M., & Zhou, J., 2021. *Engineering with Computers*. https://doi.org/10.1007/s00366-020-01241-2. Copyright 2021, by the Author(s), under exclusive license to Springer-Verlag London Ltd. part of Springer Nature.

The pseudocode shown in the Figure 23, is implemented to calculate the maximum similarity score. The left tree residual values are squared, and all the values are added together to yield the numerator, as shown in the equation above. The number of residual values is equal to the sum of the probability multiplied by one minus the probability gives the number of residual values, this gets added up with the regularization parameter. The regularization parameter can be modified to increase the speed of the model and to control overfitting of the model. Similarly, the similarity score for the right leaf is also calculated (Cao et al., 2021). The information gain of the root node is calculated using the Equation 5 where the left and right trees similarity is calculated and subtracted by the similarity of the root node (Chen, 2020).

$$gain = similarity_{left\ leaf} + similarity_{right\ left} - similarity_{root} \quad (5)$$

The same process is repeated by combining different residual values for all the features to get the best split. The splitting will stop when the leaf node has only one residue. The XGBoost algorithm builds many decisions tree and hence the trees with minimum gain must be pruned to get the best model. If the gain is less than the complexity parameter, then that node will get pruned (Chen, 2020). The final prediction of the model is calculated by the Equation 6, where the Etta is the learning rate which can range from .1 to .3 calculates the updated prediction for that record, as well as the new residue. The process is repeated until the residual is very small, and the model has been properly trained (Chen, 2020).

$$prediction = initial\ prediction\ value + (\in x\ output\ value) \quad (6)$$

In water pump machine problem, the target feature has only two possible outcomes zero and one, broken and normal respectively, therefore, the probability is calculated to be .5. Initial residue is the difference between the target value and probability. One of the features in the training data is taken and made as the root node. The tree is split depending on the similarity

score and the information gain. The next tree is constructed with the new residual value and this continues until the residue is almost equal to the target value. Now, the model is ready to be exposed to a new value. The accuracy of the model is calculated.

The ARIMA is the next model considered for the problem domain. The AR of the model stands for autoregression where the input is taken from the previous observation to predict the next outcome. The I in the model represents the integration, it is calculated by taking the observation from the previous timestamp and subtracting it with the current timestamp. The MA moving average smooths the noisy observations by taking the average of a particular number of observations. The ARIMA model has 3 parameters namely the p, d and q. The p applies to the AR part which holds the number of previous observations given to the model. The d is the order of the integrated part which holds the number of times the difference between previous and current observation are taken. The q is the MA part which holds the average window size. The time series must be a stationary with constant mean and standard deviation. If the time series is not stationary and if it is linear then the I, the integration of the ARIMA model transforms a non-stationary data into a stationary data (Brownlee, 2020).

By using the Equation 7 the non-stationary data is transformed to stationary data, where y_t is the current observation and the y_{t-1} is the previous observation. This method is called differencing. By this method the mean, variance and covariance of the series are made independent of time and free from seasonality trends (*Introduction to arima models,* n.d.).

$$y_t^* = y_t' - y_{t-1}' \tag{7}$$

The foregoing differentiation only applies to the first order, which is concerned with linear trends. The formula in Equation 8 and Equation 9 are used to solve the quadratic trends

using second order differencing where y_t is the current observation and the y_{t-1} is the previous observation and so on (*Introduction to arima models,* n.d.).

$$(y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \quad (8)$$

$$y_t - 2_{yt-1} + y_{t-2} \quad (9)$$

The Autocorrelation function plot (ACF) is drawn to see how the current time series data is correlated with the past values. The Partial Auto Correlation Function (PACF) is used to determine the number of observations to employ in the AR model, as well as to see the correlation between two points at a time and the influence of any other point. The simple ARIMA model with three parameters p, d and q is given by the formula below. The ARIMA model is trying to predict the difference between the values from one time point to previous point. In the Equation 10 the c is the constant, the $\emptyset$ y_{t-1} and y_{t-p} are the AR terms representing the lagged values of y, the theta terms represent the moving average of the lagged errors and the last term is the error term. The integrated part is taken care by the y_t (*Introduction to arima models,* n.d.).

$$y'_t = C + \emptyset_1 y'_{t-1} + \cdots + \emptyset_p y'_{t-p} + \theta_1 \epsilon_{t-1} + \cdots + \phi_q \epsilon_{t-q} + \epsilon_t \quad (10)$$

So far, by using the above formulas we have predicted only the difference from one time to another, but we wanted to predict the future value based on previous observations. The value can be predicted by using the Equation 11, where c is the constant, summation of X_{t-i} where i ranges from one to p and then added to the error term. Solving this equation will accurately predict the future value (*Introduction to arima models,* n.d.).

$$X_t = c + \Sigma_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t \quad (11)$$

Figure 24

Pseudocode for ARIMA Model

```
history = training dataset;
predictions = [empty list];
for value ∈ TestDataset do
    model = ARIMA(history, order=(p,d,q)) ;
    model.fit() ;
    one_step_forecast = model.forecast() ;
    predictions.append(one_step_forecast)
    history.append(value)
end
```

Note. "Prediction of imports of Household Appliances in Ecuador using LSTM Networks", by Tello, A., Izquierdo, I., Pacheco, G., & Vanegas, P., 2019. *Advances in Intelligent Systems and Computing*, 194–207. https://doi.org/10.1007/978-3-030-35740-5_14. Copyright 2020, by Springer Nature Switzerland AG 2020.

The pseudocode of the ARIMA model is demonstrated in the Figure 24. Firstly, two variables are created namely history and predictions. The history variable contains the entire dataset, and the prediction is the empty list. Now, each value is checked if it belongs to the test dataset. If it belongs to the test dataset a variable is created and assigned to the model. The model is fit and the forecast it made for the actual data. The predicted values are added to the prediction variable (Tello et al., 2019).

The LSTM is the next research model, which was created to overcome the inadequacies of the Recurrent Neural Network (RNN). The RNN model iteratively presents the input and previous learning, to assist the model to predict next outcome. However, if the context of a particular input changes, and the output is dependent on the input provided during the beginning stage, the model must go back in time to get the information, but the model does not remember

this knowledge, and the prediction may degrade as a result. The LSTM model was created to address the drawbacks of RNN. The LSTM model has a memory cell for remembering and forgetting the information based on the context of the input. They have the capability of remembering for a long period of time. The RNN has a single neural layer whereas the LSTM has four layers (Sivalingam, 2020).

The main components of the model are memory cell, input gate, forget gate, and the output gate. The forget gate takes the previous states output, and the current input is concatenated and given to the sigmoid function which converts the value between zero to one. The Equation 12 shows that the W_f is the product of the weights of current and previous output. The h_{t-1} and x_t are the previous output and current input respectively. This is now passed to the sigmoid function which translates the value between zero to one. If the previous output is not like the current input, then there is a change of context, the values with zero indicates the mismatch and if there are more ones then there is no change in the context. The pointwise operator multiplies the two vectors and remembers the information indicated by one and forgets the rest of the information (*Understanding LSTM networks,* n.d.).

$$f_t = \sigma\left(W_f . [h_{t-1}, x_t] + b_f\right) \tag{12}$$

The information is now added to the memory cell. Here a tanh operation is performed which converts the input between negative one and positive one. The output from the previous stage and the tanh is connected to a pointwise operator, a comparison is made between two vectors and if there is any new information it gets added to the memory cell. This layer is called the input layer. The below Equation 13 and Equation 14 computes the output values of the sigmoid function represented by i_t is the output of the sigmoid function and the C_t is the output of the tanh function (*Understanding LSTM networks,* n.d.).

$$i_t = \sigma(W_i . [h_{t-1}, x_t] + b_i) \quad (13)$$

$$C_t = \tanh(W_c . [h_{t-1}, x_t] + b_c \quad (14)$$

The last stage is the output layer, here the pointwise operation is applied on the output of the sigmoid function and the values in the memory cell is passed to the tanh function, this will retrieve only the contents which has a meaningful value and passed as a previous output for the next cell (*Understanding LSTM networks,* n.d.).

Before applying the LSTM model to our problem domain, the data must first be checked for stationarity. If the data points are not stationary, it may result in giving a lot of errors. A sample of the entire data is given as the input to the model, after performing all the operations, the memory cell gives the output. This output and the same input expect the first data points is given to the second layer. The test data is passed to check the accuracy, once the results are promising then the model is exposed to a new data to predict the future failure (*Understanding LSTM networks,* n.d.).

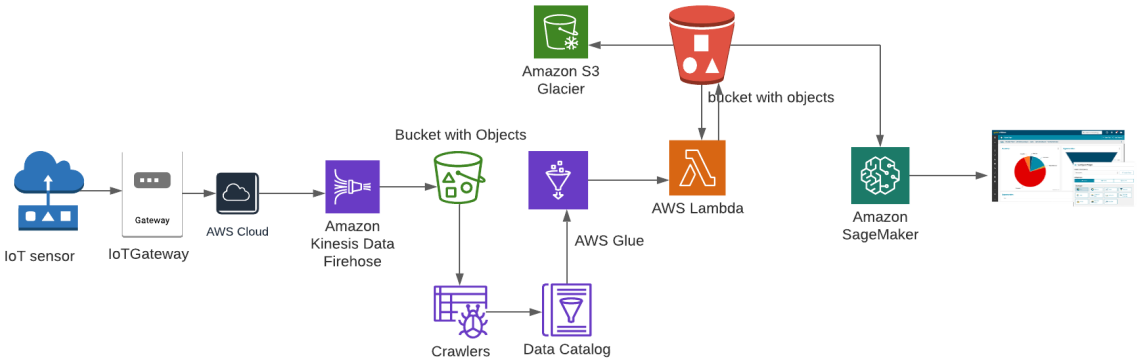
4.2 Model Supports

The architecture for predicting the water pump machine failure is shown in the Figure 25. The data from the IoT sensors are aggregated using the IoT Gateway, and then sent to the cloud services. The sensors are connected to the gateway by Bluetooth, Wi-Fi or an ethernet cable. The IoT gateway can filter the critical information and send it on to the cloud (Admin, 2017). The streaming data from the sensors are streamed by the Amazon Kinesis Data Firehose. The data in a JSON format is converted to a table by using Amazon Glue. Also, various preprocessing of the data is done by using the Amazon Glue services. The Lambda Function ensures the streaming and batch data continuity. The cleaned data is sent to the Amazon S3 Bucket. The machine learning models are implemented in the Amazon SageMaker by taking the data from the Amazon

S3 bucket. The prediction is visualized using the interactive UI designed by using the react framework.

Figure 25

Water Pump Failure Prediction Architecture



Note. Water pump failure prediction data flow architecture

The models are implemented in the Amazon Sagemaker by consuming the data from Amazon S3 Bucket. The architecture of the LR models is illustrated below.

Figure 26

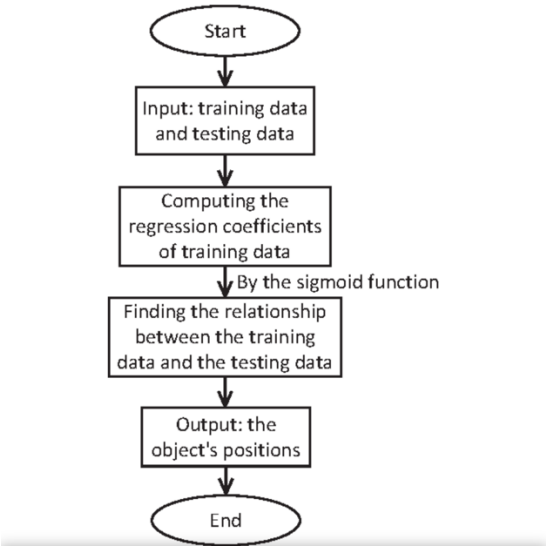
Architecture of LR Model

Note. "Machine learning: What is logistic regression?", by Sarkar, P., Roy, G. K., Sugandhi, A., & Vora, D. D., 2019. Knowledgehut. https://www.knowledgehut.com/blog/data-science/logistic-regression-for-machine-learning. Copyright 2021 by Knowledgehut.

As shown in the Figure 26, the x1, x2 and x3 are the data points multiplied with the slope and intercept calculates the distance between the actual point and the best fit line, then passed to the sigmoid function to transform the probability between zero and one. The gradient descent is used to find the optimal parameters by minimizing the loss function (Sarkar et al., 2019). The below flowchart demonstrates the data flow in the LR model.

Figure 27

LR Model Dataflow Diagram

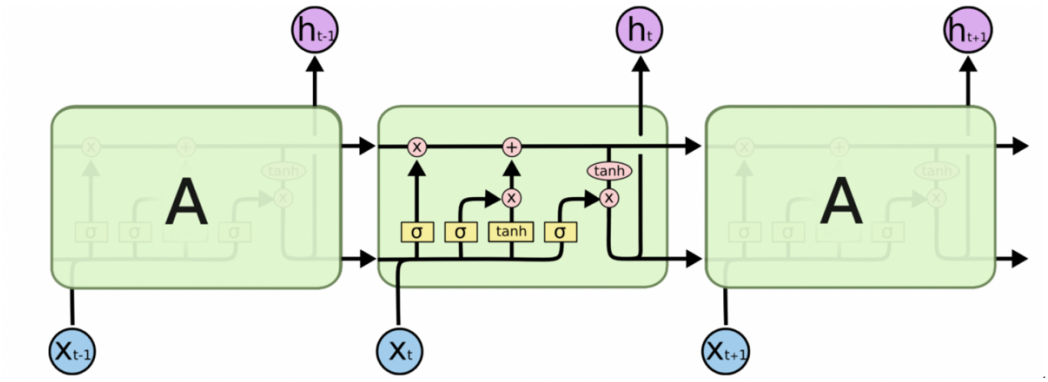


Note. "Logistic regression-based device-free localization in changeable environments", by Lei, Q., Lv, H., Zhang, H., Sun, H., & Tang, L., 2016. *IEEE 13th International Conference on Signal Processing (ICSP)*. https://doi.org/10.1109/icsp.2016.7877992. Copyright 2021 by IEEE.

As shown in the Figure 27**,** the model is fed data from the training and testing phases.The regression coefficient is calculated for the training dataset. By using the sigmoid function, the outliers are handled, and the best fit line is determined. The train and test set relationship must be calculated. Based on the results of evaluation the data point is placed above or below the best fit line indicating if the water pump machine is in normal or broken state (Lei et al., 2016).

Figure 28

Architecture Diagram of LSTM Model

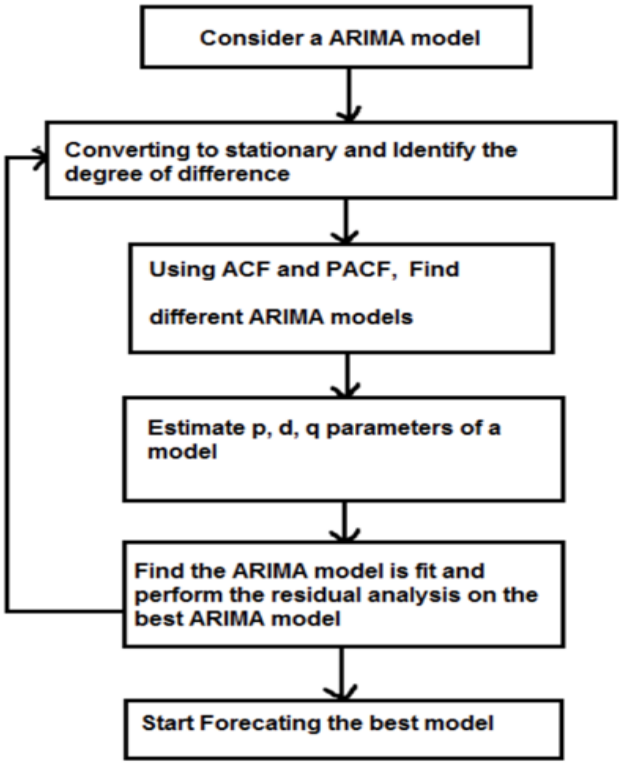


Note. Understanding LSTM Networks. *"Understanding LSTM networks"*, by colah's blog, (n.d.). http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

The LSTM architecture is shown in the Figure 28. The main components are memory cell, input gate, forget gate, and the output gate. The straight line with the pointwise operation at the top is the memory cell, the left sigmoid function along with the pointwise operator is the forget gate, the middle sigmoid function along with the tanh is the input gate and the sigmoid function at the right is the output gate. The rectangular boxes are the neural network layer, the circles are the pointwise operators, the arrows are the vector to transfer the information from one place to another (*Understanding LSTM networks,* n.d.).

Figure 29

Dataflow Diagram of ARIMA Model



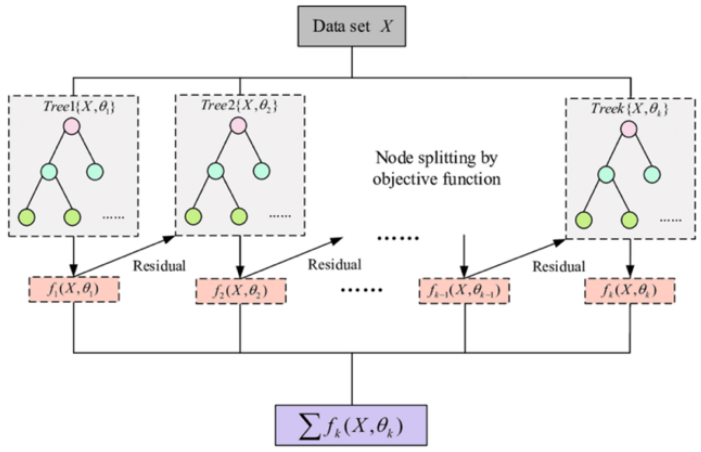
Note. "Annual automobile sales prediction using Arima model", by Shakti, S. P., Hassan, M. K., Zhenning, Y., Caytiles, R. D., & N.Ch.S.N, I., 2017. *International Journal of Hybrid Information Technology*, 10(6), 13–22. https://doi.org/10.14257/ijhit.2017.10.6.02. Copyright 2017, by Science & Engineering Research Support soCiety.

The flowchart shown in the Figure 29, illustrates the flow of the ARIMA model for our water pump failure prediction problem domain. First the dataset is plotted by having the time series at the x axis and the target feature in the y axis. The dataset follows a quadratic trend and not stationary. The next step is to make the data points stationary. The mean, variance and covariance of the series are made independent of time and free from seasonality trends. The ACF is used to visualize the relationship between current time series data and previous values. The

PACF is used to determine the number of observations to employ in the AR model, as well as to see the correlation between two points at a time and the influence of any other point. As a quadratic trend is observed a second order difference must be used to make the trend stationary. Now, the ARIMA model is fit, the residual be analyzed, and the best model is selected. Once the model is ready, the future failures of the water pump machine is predicted (Shakti et al., 2017).

Figure 30

Architecture Diagram of XGBoost Model



Note. "Degradation state recognition of piston pump based on ICEEMDAN and XGBoost", by Guo, R., Zhao, Z., Wang, T., Liu, G., Zhao, J., & Gao, D., 2020. *Applied Sciences*, 10(18), 6593. https://doi.org/10.3390/app10186593. Copyright 2008-2021 by ResearchGate GmbH.

The architecture shown in the Figure 30, explains the working of the XGBoost technique. The algorithm calculates the probability based on the target feature. The residue is the difference between the target feature and the probability. The data is now split based on any one of the features in the dataset. The selected feature is the root node of the tree. The decision tree is constructed by taking the value of the residue. The XGBoost must always have a binary split and hence irrespective of the number of grouping of the root node the tree will have only two leaves. This is continued until the residue becomes very low (Guo et al., 2020).

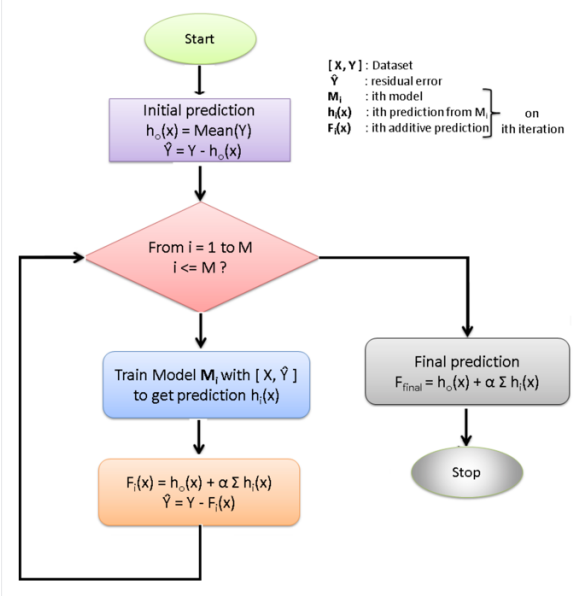
The similarity score is calculated between the two leaf nodes by using the Equation 17 where the residue from each tree is squared and divided by the sum of the number of residual value and the regularization term lambda (Chen, 2020).

$$similarity\ score = \frac{(\Sigma Residual)^2}{number\ of\ Residual + \lambda} \tag{17}$$

The data flow of the XGBoost algorithm is shown in the Figure 31. The dataset is given to the model. The initial probability is calculated based on the target feature. In the below diagram, the X and Y represents the dataset. The y hat value is the residue calculated by taking the difference between target feature and the probability. The decision tree is constructed by taking the residue value. Several decision trees are constructed until the value of the residue becomes low. Finally, the model predicts the output (*How does xgboost work,* n.d.).

Figure 31

Dataflow Diagram of XGBoost Model



Note. How does XGBoost work. *"How does xgboost work",* (n.d.). https://morioh.com/p/3883937008ee.

4.3 Model Comparison and Justification

Machine learning is a technique in which a computer learns patterns in data and then builds a model to predict the future based on those patterns. Four distinct models that support time series data are presented to solve our problem domain. To acquire the optimal output for the problem, these four models must be evaluated in terms of accuracy and performance.

Table 3

Advantages and Disadvantages of LR Model

Model	Advantages	Disadvantages
LR	Easy to train and implement. Easily expanded to support multi-class classification. Good accuracy when the data points are linearly separable. When prone to overfitting, the regularization technique can be used to reduce it. Quick at classification, and whenever it is exposed to previously unseen data, it swiftly classifies it.	It is suitable only if the training data is more than the features considered, otherwise the model will be prone to overfitting. The model seems to perform well with the test data, but it may not work well with the actual data. The data must be distributed linearly else LR model cannot be used.

Note. LR model advantages and disadvantages.

The advantages and disadvantages of the LR are listed in Table 3 (*Advantages and disadvantages of logistic regression,* 2020). The primary reason for choosing the LR model is its ease of training and implementation. The reason for using this model in the water pump failure prediction problem is because the data is linearly separable, simple to implement, and the output is a categorical that forecasts the likelihood of a given instance belonging to one of several classes.

Table 4

Advantages and Disadvantages of XGBoost Model

Model	Advantages	Disadvantages
XGBoost	The overfitting problem is handled by using the regularization technique which is inbuilt in the model. High computational power uses parallel processing. Handles missing values Runs cross-validation in each iteration to get the best possible output. Split a tree to a maximum depth then prune if there is a loss.	The model does not perform well on the unstructured data. The model requires a large training set. It is very sensitive to outliers.

Note. XGBoost algorithm advantages and disadvantages

The advantages and disadvantages of the XGBoost algorithm is listed in Table 4 (Hachcham Aymane et al., 2021). The XGBoost is a boosting ensemble technique that is quickly becoming a popular algorithm, offering several benefits over other models. As the dataset used for this problem domain is a time series, it must first be transformed to supervised learning; the predicted values are binary, hence the model is supervised learning problem. The justification for using this model is that the pump failure must be investigated as soon as possible. Because of its execution speed and performance, XGBoost is used.

Table 5

Advantages and Disadvantages of ARIMA Model

Model	Advantages	Disadvantages
ARIMA	ARIMA is known for time series forecasting. Only requires the previous time series data to predict the next outcome. Good prediction accuracy. Requires minimum parameters for prediction.	Expertise needed to evaluate the model based on the input data. Depends on the past values, hence prediction resembles the past giving inaccurate results. Data must be stationary.

Note. ARIMA model advantages and disadvantages

The advantages and disadvantages of the ARIMA model is shown in the Table 5 (Hayes, 2021). One of the time series forecasting models is the ARIMA. This model is used in our problem domain, as the input to the model is a time series data. The ARIMA model is implementing because the models is more flexible than other statistical models and forecasting in general is tough. ARIMA performs well compared to other forecasting algorithm.

Table 6

Advantages and Disadvantages of LSTM Model

Model	Advantages	Disadvantages
LSTM	Persists past and present information for long time. For inputs with a sequence, the LSTM is a good model. Learning rate, input and output biases are provided in the model.	Stores lot of information in the memory and hence takes lot of time for training. The model must be implemented in a hardware which has a good memory. LSTM is prone to overfitting most of the time.

Note. LSTM model advantages and disadvantages

The advantages and disadvantages of using the LSTM model is shown in the Table 6 (*Understanding of LSTM networks,* 2021). The LSTM is a research model, created to overcome

the inadequacies of the Recurrent Neural Network (RNN). The LSTM model is implemented in our problem domain because the data collected from the sensors are time series; hence this model is appropriate as the previous timestamp data is kept in memory for a long time, the model learns from the data in memory to predict the future value.

The models are compared with the advantages and disadvantages stated above to build the most appropriate model for our water pump failure prediction problem. The LSTM model is compared with the ARIMA model, the LSTM model requires a huge number of data points to train the model whereas the ARIMA model requires a less training dataset. The three parameters p, d and q must be calculated from the data and must be given to the model, on the other hand the LSTM model have some hyperparameters which must be tuned. The LSTM model will take a lot of time to train the model and many parameters to be tuned. The long-term forecasting yields better results with LSTM compared to ARIMA. Based on our time series dataset it is seen that the data has no regular pattern and hence ARIMA model will not be suitable for this kind of data. Also, implementing ARIMA is not easy as many hypotheses must be met. Hence, the LSTM is now compared with XGBoost.

The XGBoost is compared with the LSTM model in terms of the following parameters. First, both the models are implemented, and the results are captured. The predicted accuracy is the same for both the models. In both models the weights are calculated, and it clearly shows that the XGBoost gives a best result when analyzed with the LSTM results. The next parameter is the computational speed, this parameter is very important to get quicker and accurate results, XGBoost is faster compared to LSTM model. Now XGBoost is compared with the LR model.

When compared to the LR model, XGBoost outperforms it for a variety of reasons. Because data in the real world are not linearly separable, the XGBoost method does not require

linearly separable data. In comparison to the LR model, the computational speed and accuracy are extremely high. The XGBoost model handles the overfitting problem by employing regularization, and it also handles missing values. The XGBoost model appears to be the best fit for the water pump failure prediction problem among the three models.

4.4 Model Evaluation Methods

The model must be evaluated on various metrices to measure the accuracy and to know how well the model perform on the new input data. Firstly, the LR model is tested on various evaluation metrics namely the confusion matrix, accuracy, f1 score, ROC/AUC. The confusion matrix shows the count of the correctly and incorrectly predicted values. The terms in confusion matrix, the True Positive (TP) if both new and initial values are yes, the True Negative (TN) if the initial and new values are no, the False Positive (FP) if the new value is yes but the initial output is no, False Negative (FN) if the new output is yes but initial result is no (Narkhede, 2021).

The accuracy score is calculated by adding the true positive and true negative values divided by the other classifications namely TP, TN, FP, and FN as shown in Equation 18. The recall is another measure to evaluate the model. The recall is calculated by dividing TP by TP and FN, it defines the percentage of results correctly predicted by the model as shown in Equation 19. The precision is a matrix which measure the predicted result which are relevant. The precision is calculated by dividing TP and the summation of TP and FN as shown in Equation 20. The F1 score is a method of evaluation that involves multiplying precision and recall and dividing the result by the sum of accuracy and recall as shown in Equation 21. F1 score marks the balance between precision and recall (Jeremy Jordan, 2018).

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{18}$$

$$recall = \frac{TP}{TP+FN} \quad (19)$$

$$precision = \frac{TP}{TP+FP} \quad (20)$$

$$F1 = 2 * \frac{precision*recall}{precision+recall} \quad (21)$$

A threshold is specified in the LR model to determine whether the machine is normal or broken. The machine is broken if the datapoint crosses the threshold, and it is normal if it is below the threshold. When data is improperly identified, shifting the threshold can improve model accuracy, but it is not the best solution. Also, many confusion matrices will be created when shifting the thresholds, hence, to summarize all the information of confusion matrix Receiver Operator Characteristic (ROC) graphs are plotted. A graph with FP rate and TP rate in x and y axis respectively is plotted. The TP rate is identified by calculating the recall metrics. The FP rate is calculated by dividing FP by the summation of FP and TN. The Area Under ROC curve (AUC) compares two ROC curves of different models. The value of the AUC must be higher only then we can conclude the model will accurately predict the output (Narkhede, 2021).

The confusion matrix for each algorithm is calculated by classifying the prediction to be TP, TN, FP, and FN. The LR model classified 54847 records correctly, and incorrectly classified 256 records. Similarly, the ARIMA model classified 54756 records correctly, incorrectly classified 359 records. The XGBoost and LSTM models has a very few incorrect predictions, 60 and 89 respectively. The values were compared between these two algorithms and it seems that the XGBoost algorithm has the highest accuracy.

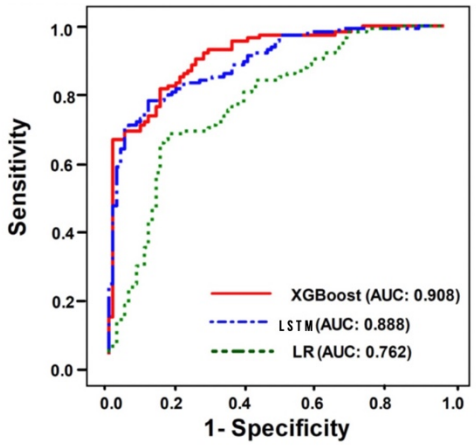
The precision score for the LSTM model is estimated to be 100% and the recall as 99% whereas the XGBoost shows up a high percentage of both the precision and recall. The LR has a precision value estimated to be 75% which is low when compared to XGBoost and LSTM.

Similarly, the ARIMA model has a precision score of 70% and the accuracy is calculated as 80% (Prabhakaralagarsamy, 2021).

The AUC curve for the following models XGBoost, LSTM and LR is shown in the Figure 32. In comparison with the other two the AUC score for the XGBoost algorithm is high. The XGBoost AUC score is nearing one, which states that the model represents good separability. The models with low AUC scores and the ones which is approaching zero means that the separability is very low and hence the model will not show a good accuracy (Lin et al., 2021).

Figure 32

ROC Curves for Models



Note. "Explainable machine learning to predict successful weaning among patients requiring prolonged mechanical ventilation: A retrospective cohort study in central Taiwan", by Lin, M.-Y., Li, C.-C., Lin, P.-H., Wang, J.-L., Chan, M.-C., Wu, C.-L., & Chao, W.-C., 2021. *Frontiers in Medicine*, *8*. https://doi.org/10.3389/fmed.2021.663739. Copyright 2008-2021 by ResearchGate GmbH.

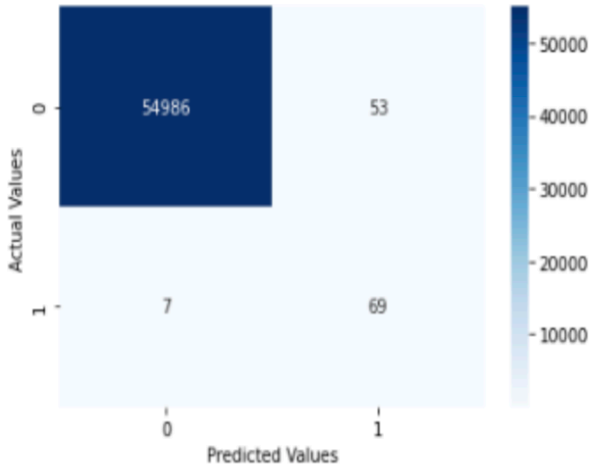
4.5 Model Validation and Evaluation

The images below illustrate the confusion matrix for the water pump machine failure prediction dataset. The top left to right diagonal in the diagram indicates that the model has correctly predicted the values. The other diagonal indicates the number of datapoints incorrectly predicted (Prabhakaralagarsamy, 2021).

The confusion matrix of the XGBoost algorithm is illustrates in the Figure 33, the total number of records which was correctly classified by the model as 55055 datapoints and the number of incorrect predictions as 64 (Prabhakaralagarsamy, 2021). The XGBoost has a small number of incorrect predictions compared to the LR model. Also, the recall, precision, and F1 score are high for the XGBoost model. The ROC curve is calculated between the two algorithms and it was found that the AUC of the XGBoost is greater than the AUC of LR model.

Figure 33

Confusion Matrix of XGBoost Algorithm

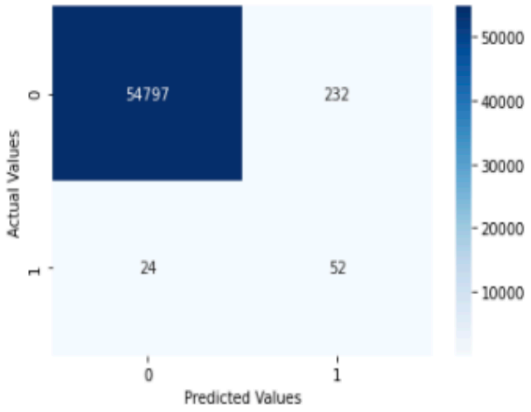


Note. "Predict pump failure before it happens using Deep Learning Model", by Prabhakaralagarsamy, 2021. *Medium*. https://becominghuman.ai/predict-pump-failure-before-it-happens-using-deep-learning-model-dc886bfa073e.

Similarly, the confusion matric for the LR model is shown in the Figure 34. The number of correct predictions were estimated to be 54849 and incorrect predictions were 256 datapoints (Prabhakaralagarsamy, 2021).

Figure 34

Confusion Matrix of LR Model

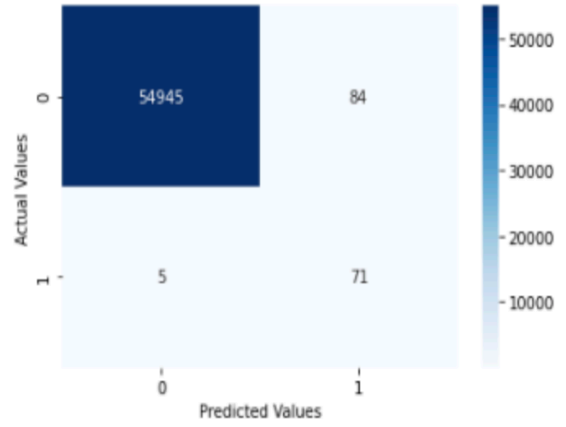


Note. "Predict pump failure before it happens using Deep Learning Model", by Prabhakaralagarsamy, 2021. *Medium*. https://becominghuman.ai/predict-pump-failure-before-it-happens-using-deep-learning-model-dc886bfa073e.

The confusion matrix of the LSTM model is shown in the Figure 35. The sum of 54945 and 71 records are correctly predicted by the model and the sum of 84 and five are the incorrectly classified records (Prabhakaralagarsamy, 2021).

Figure 35

Confusion Matrix of LSTM Model

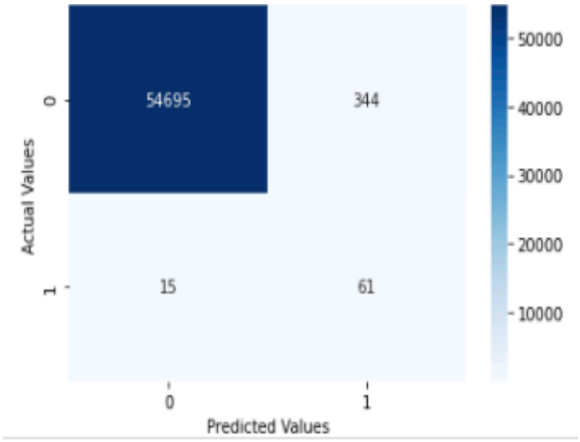


Note. "Predict pump failure before it happens using Deep Learning Model", by Prabhakaralagarsamy, 2021. *Medium*. https://becominghuman.ai/predict-pump-failure-before-it-happens-using-deep-learning-model-dc886bfa073e.

As shown in the Figure 36, the ARIMA models has a similar pattern as that of the LR model. The number incorrect predictions are high. The sum of 54695 and 61 records are correctly classifies by the model whereas the total of 344 and 15 were incorrectly classified (Prabhakaralagarsamy, 2021). The confusion matrix for this model is presented below.

Figure 36

Confusion Matrix of ARIMA Model



Note. "Predict pump failure before it happens using Deep Learning Model", by Prabhakaralagarsamy, 2021. *Medium*. https://becominghuman.ai/predict-pump-failure-before-it-happens-using-deep-learning-model-dc886bfa073e.

By comparing the confusion matrix of all the algorithms, it was found that the LSTM and XGBoost is relatively better than the LR and ARIMA for the water pump failure prediction problem. Hence these two algorithms are further evaluated on the other metrics. The evaluation results show that the accuracy of XGBoost is high compared to LSTM (Prabhakaralagarsamy, 2021). Therefore, XGBoost algorithm is chosen for our problem domain and pump failure must be investigated as soon as possible. Because of its execution speed and performance, XGBoost is used. The comparison between the two selected algorithms is presented below.

Figure 37

Comparison Between LSTM and XGBoost Based on Evaluation Matrix.

LSTM

	precision	recall	f1-score	support
0	1.00	0.99	1.00	55039
1	0.15	0.80	0.25	76
accuracy			0.99	55115
macro avg	0.58	0.90	0.63	55115
weighted avg	1.00	0.99	1.00	55115

XGBoost

	precision	recall	f1-score	support
0	1.00	1.00	1.00	150787
1	1.00	1.00	1.00	14408
accuracy			1.00	165195
macro avg	1.00	1.00	1.00	165195
weighted avg	1.00	1.00	1.00	165195

Note. "Predict pump failure before it happens using Deep Learning Model", by Prabhakaralagarsamy, 2021. *Medium*. https://becominghuman.ai/predict-pump-failure-before-it-happens-using-deep-learning-model-dc886bfa073e.

As shown in the Figure 37, the precision, recall, f1-score is calculated for LSTM and XGBoost models, the accuracy of the LSTM model is estimated to be 99% whereas the accuracy of the XGBoost model is 100%. Hence XGBoost algorithm is used for predicting the water pump machine failures. (Prabhakaralagarsamy, 2021).

References

Admin. (2017, October 10). *IIOT based remote pump monitoring*. Environmental Engineering News Online. Retrieved November 20, 2021, from https://www.environmentalengineering.org.uk/news/iiot-based-remote-pump-monitoring-6216/.

Advantages and disadvantages of logistic regression. GeeksforGeeks. (2020, September 2). Retrieved November 20, 2021, from https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/.

Alsop, T. (2016, September 7). *IOT average sensor costs 2004-2020*. Statista. Retrieved October 9, 2021, from https://www.statista.com/statistics/682846/vr-tethered-hmd-average-selling-price/.

AWS Pricing Calculator. (n.d.). Retrieved October 8, 2021, from https://calculator.aws/#/.

Aymane Hachcham Data Scientist at Spotbills | Machine Learning enthusiast. Follow me on, Hachcham, A., Data Scientist at Spotbills | Machine Learning enthusiast., & on, F. me. (2021, August 12). *XGBoost: Everything you need to know*. neptune.ai. Retrieved November 20, 2021, from https://neptune.ai/blog/xgboost-everything-you-need-to-know.

Bilandi, N., Verma, H. K., & Dhir, R. (2021). An intelligent and energy-efficient wireless body area network to control coronavirus outbreak. *Arabian Journal for Science and Engineering*, *46*(9), 8203–8222. https://doi.org/10.1007/s13369-021-05411-2.

Brownlee, J. (2021, February 16). *A gentle introduction to XGBoost for applied machine learning*. Machine Learning Mastery. Retrieved November 20, 2021, from https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/.

Brownlee, J. (2020, December 9). *How to create an Arima model for time series forecasting in Python*. Machine Learning Mastery. Retrieved November 20, 2021, from https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/.

Business, F. S. of. (n.d.). Introduction to arima models. Retrieved November 20, 2021, from https://people.duke.edu/~rnau/411arim.htm.

Cao, J., Gao, J., Nikafshan Rad, H., Mohammed, A. S., Hasanipanah, M., & Zhou, J. (2021). A novel systematic and evolved approach based on XGBoost-Firefly algorithm to predict Young's modulus and unconfined compressive strength of Rock. *Engineering with Computers*. https://doi.org/10.1007/s00366-020-01241-2.

Chen, S. (2020, July 7). *Simple math about xgboost*. Medium. Retrieved November 20, 2021, from https://towardsdatascience.com/simple-math-about-xgboost-b9a924aaae9f.

Ennomotive. (n.d.). *Industrial IOT is booming thanks to a drop in sensor prices*. 💡Open Innovation Hub for Companies, Startups, and Experts. Retrieved December 9, 2021, from https://www.ennomotive.com/industrial-iot-sensor-prices.

Fernando, J. (2021, December 7). *What is R-squared?* Investopedia. Retrieved December 9, 2021, from https://www.investopedia.com/terms/r/r-squared.asp#:~:text=In%20other%20fields%2C%20the%20standards,would%20show%20a%20low%20correlation.

Fisseha Berhane, Phd. Fisseha Berhane. (n.d.). Retrieved November 20, 2021, from https://datascience-enthusiast.com/Matlab/logistic_regression.html.

Gillis, A. S. (2021, March 24). *The 5 V's of big data*. SearchDataManagement. Retrieved December 1, 2021, from https://searchdatamanagement.techtarget.com/definition/5-Vs-of-big-data.

Guo, R., Zhao, Z., Wang, T., Liu, G., Zhao, J., & Gao, D. (2020). Degradation state recognition of piston pump based on ICEEMDAN and XGBoost. *Applied Sciences*, *10*(18), 6593. https://doi.org/10.3390/app10186593.

Han Ji-Hyeong & Chi Su-Young (2016, July 1). *Consideration of manufacturing data to apply machine learning methods for predictive manufacturing*. IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/7536995

Harrison, P., Chapman, N., & Beaton, C. (1997). *Glue*. Amazon. Retrieved October 9, 2021, from https://aws.amazon.com/glue/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc.

How does xgboost work. How does XGBoost work. (n.d.). Retrieved November 15, 2021, from https://morioh.com/p/3883937008ee.

How variable speed pumps & pressure sensors can improve water systems. Water Quality Products. (2021, May 12). Retrieved October 17, 2021, from https://www.wqpmag.com/commercial-water/how-variable-speed-pumps-pressure-sensors-can-improve-water-systems.

Hayes, A. (2021, November 20). *Autoregressive Integrated moving average (ARIMA)*. Investopedia. Retrieved November 20, 2021, from https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp.

Jeremy Jordan. (2018, August 25). *Evaluating a machine learning model.* Jeremy Jordan. Retrieved November 20, 2021, from https://www.jeremyjordan.me/evaluating-a-machine-learning-model/.

Johnson, D. (2021, November 11). *Supervised vs unsupervised learning: Key differences*. Guru99. Retrieved November 20, 2021, from https://www.guru99.com/supervised-vs-unsupervised-learning.html.

Kanawaday Ameeth & Sane Aditya (2017, November 1). *Machine learning for predictive maintenance of industrial machines using IoT sensor data*. IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/8342870.

Lambda Architecture - analytics lens - docs.aws.amazon.com. (n.d.). Retrieved October 15, 2021, from https://docs.aws.amazon.com/wellarchitected/latest/analytics-lens/lambda-architecture.html.

Lechevalier David, Narayanan Anantha & Rachuri Sudarsan (2014, October 1). *Towards a domain-specific framework for predictive analytics in manufacturing*. IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/7004332

Lei, Q., Lv, H., Zhang, H., Sun, H., & Tang, L. (2016). Logistic regression-based device-free localization in changeable environments. *2016 IEEE 13th International Conference on Signal Processing (ICSP)*. https://doi.org/10.1109/icsp.2016.7877992.

Lin, M.-Y., Li, C.-C., Lin, P.-H., Wang, J.-L., Chan, M.-C., Wu, C.-L., & Chao, W.-C. (2021). Explainable machine learning to predict successful weaning among patients requiring prolonged mechanical ventilation: A retrospective cohort study in central Taiwan. *Frontiers in Medicine*, *8*. https://doi.org/10.3389/fmed.2021.663739.

Logistic regression. 06_Logistic_Regression. (n.d.). Retrieved November 20, 2021, from https://www.holehouse.org/mlclass/06_Logistic_Regression.html.

Mishra, A. (2019). *Machine learning in the AWS cloud: Add intellegence to applications with Amazon Sagemaker and Amazon Rekognition*. Amazon. Retrieved October 10, 2021, from https://aws.amazon.com/sagemaker/.

Narkhede, S. (2021, June 15). *Understanding AUC - roc curve*. Medium. Retrieved November 20, 2021, from https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5.

Narkhede, S. (2021, June 15). *Understanding confusion matrix*. Medium. Retrieved November 20, 2021, from https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62.

Posey, B., & Lavery, T. (2021, April 2). *What is an IOT gateway?* IoT Agenda. Retrieved October 9, 2021, from https://internetofthingsagenda.techtarget.com/definition/IoT-gateway.

Posey, B., Rosencrance, L., & Shea, S. (2021, March 24). *What is iiot? industrial internet of things explained*. IoT Agenda. Retrieved December 9, 2021, from https://internetofthingsagenda.techtarget.com/definition/Industrial-Internet-of-Things-IIoT.

Prabhakaralagarsamy. (2021, August 19). *Predict pump failure before it happens using Deep Learning Model*. Medium. Retrieved November 16, 2021, from https://becominghuman.ai/predict-pump-failure-before-it-happens-using-deep-learning-model-dc886bfa073e.

Predictive analytics: What it is and why it matters. SAS. (n.d.). Retrieved December 9, 2021, from https://www.sas.com/en_us/insights/analytics/predictive-analytics.html.

Pump_sensor_data. (2019, March 4). Kaggle. Retrieved October 9, 2021, from https://www.kaggle.com/nphantawee/pump-sensor-data.

Reference architecture - analytics lens - docs.aws.amazon.com. (n.d.). Retrieved October 15, 2021, from https://docs.aws.amazon.com/wellarchitected/latest/analytics-lens/reference-architecture-3.html.

Sarkar, P., Roy, G. K., Sugandhi, A., & Vora, D. D. (2019, September 23). *Machine learning: What is logistic regression?* Knowledgehut. Retrieved November 15, 2021, from https://www.knowledgehut.com/blog/data-science/logistic-regression-for-machine-learning.

Shakti, S. P., Hassan, M. K., Zhenning, Y., Caytiles, R. D., & N.Ch.S.N, I. (2017). Annual automobile sales prediction using Arima model. *International Journal of Hybrid Information Technology*, *10*(6), 13–22. https://doi.org/10.14257/ijhit.2017.10.6.02.

Shrivastava, Soumya. "Cross Validation in Time Series." *Medium*, Medium, 17 Jan. 2020, https://medium.com/@soumyachess1496/cross-validation-in-time-series-566ae4981ce4.

Sivalingam, A. (2020, July 30). *Why do we need LSTM*. Medium. Retrieved November 20, 2021, from https://towardsdatascience.com/why-do-we-need-lstm-a343836ec4bc.

sklearn.feature_selection.SelectKBest. (n.d.). Scikit-Learn. https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html

sklearn.impute.KNNImputer. (n.d.). Scikit-Learn. https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html

sklearn.impute.SimpleImputer. (n.d.). Scikit-Learn. https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html#sklearn.impute.SimpleImputer

sklearn.preprocessing.Normalizer. (n.d.). Scikit-Learn. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.Normalizer.html

Tello, A., Izquierdo, I., Pacheco, G., & Vanegas, P. (2019). Prediction of imports of Household Appliances in Ecuador using LSTM Networks. *Advances in Intelligent Systems and Computing*, 194–207. https://doi.org/10.1007/978-3-030-35740-5_14.

Understanding of LSTM networks. GeeksforGeeks. (2021, June 25). Retrieved November 20, 2021, from https://www.geeksforgeeks.org/understanding-of-lstm-networks/.

Understanding LSTM networks. Understanding LSTM Networks -- colah's blog. (n.d.). Retrieved November 15, 2021, from http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

Walmsley, G., & Bie, J. (1983). *Kinesis*. Amazon. Retrieved October 9, 2021, from https://aws.amazon.com/kinesis/.

What is analytics? Oracle. (n.d.). Retrieved December 9, 2021, from https://www.oracle.com/business-analytics/what-is-analytics/.

Wu, D. (2017, July 1). A comparative study on machine learning algorithms for smart manufacturing: Tool wear prediction using random forests | J. Manuf. Sci. Eng. | ASME Digital Collection. https://asmedigitalcollection.asme.org/manufacturingscience/article/139/7/071018/454654/A-Comparative-Study-on-Machine-Learning-Algorithms