IoT based Generator Fuel Level Severity Prediction Using Machine Learning Method for Cell Towers

By: SOLOMON TESHOME REGASSA

A thesis submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing.

Presented in Partial Fulfilment of the Requirement for the Degree of

Masters in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

JUNE 2022

Adama, Ethiopia

IoT based Generator Fuel Level Severity Prediction Using Machine Learning Method for Cell Towers

Solomon Teshome Regassa

Advisor(s): Bahiru Shifaw (PhD)

A thesis submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing.

Presented in Partial Fulfilment of the Requirement for the

Degree of Masters in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

JUNE 2022

Adama, Ethiopia

APPROVAL SHEET

I, the advisor of the thesis entitled IoT based Generator Fuel Level Severity Prediction Using Machine Learning Method for Cell Towers" and developed by Solomon Teshome, hereby certify the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Advisor	Signature	Date

The undersigned, members of the Board of Examiners of Solomon Teshome Regassa, have read and evaluated the thesis entitled " IoT based Generator Fuel Level Severity Prediction Using Machine Learning Method for Cell Towers " and examined the candidate during the open defense. As a result, this is, therefore, to certify that the thesis is accepted for partial fulfilment of the requirement of the degree of Master of Science in Computer science and Engineering.

Cahir person	signature	date
Internal Examiner	signature	date
External Examiner	signature	date

Finally, approval and acceptance of the thesis are contingent upon submitting its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate.

Council (DGC) and School Graduate Committee (SGC).

Department Head	signature	date
School Dean	Signature	date
OfficeofPostgraduatestudiesDean	Signature	date

Declaration

I, at this moment declare that this MSC thesis is my original work and has not been presented as a partial degree requirement in other university. In addition, all sources of materials for the thesis have been duly acknowledged.

Name: Solomon Teshome Regassa

Signature: 

This MSc thesis has been submitted for examination with my approval as a thesis advisor.

Name: Bahiru Shifaw (PhD.)

Signature: _____

Date of Submission: _____

Acknowledgement

I am grateful to the Almighty for blessing me and providing me with good health and strength throughout my thesis journey. My heartfelt thanks go to my advisor, Dr Bahiru Shifaw, for his willingness to offer research advice and encouragement, guidance, support, and critical comments, which helped improve this thesis work.

I would like to thank my classmates for their support until the completion of the thesis.

Finally, yet importantly, I would like to thank my family for their unending support and blessing, as well as all of my friends who helped me finish the thesis.

Table of Contents

Declaration i
Acknowledgement ii
Table of Contents iii
LIST OF FIGURES vii
List of Tables ix
LIST OF LIST OF ABBREVIATIONS x
Abstract xi
CHAPTER ONE INTRODUCTION 1

1.1 Background of the study 1
1.2 Motivation 4
1.3 Statement of the problem. 4
1.4 Objectives of the Study 5

1.4.1 General Object 5
1.4.2 Specific Objective 5

1.5 Scope and Limitations of the Study 5

1.5.1. Scope of the Study 5
1.5.2. Limitations of the Study 5

1.6 Significance of the study 6
1.7 Thesis Organization 6

CHAPTER TWO LITERATURE REVIEW 7

2.1 Introduction 7
2.2 Internet of things 7
2.3 Architecture of IoT 7
2.4 Fuel level monitoring and IoT 8
2.5 IoT in BTS Diesel Generators fuel level monitoring. 8
2.6 achine Learning 9

2.6.1 Supervised Machine Learning 9
2.6.2 Supervised machine learning algorithms 9

2.7 Supervised Learning Application 10
2.8 Kinds of supervised Machine learning 11

2.8.1 Regression 11
2.8.2 Classification 11
2.8.3 Advantages of Supervised learning 14

2.9 Related works 14

CHAPTER THREE METHODOLOGY 17

3.1 Overview 17
3.2 Research Procedure 17
3.3 Tools Used 17
3.3.1 Hardware tools 17

3.3.2 Software tools 18

3.4. Data collection 18
3.5. Modeling for Predicting Fuel Level Severity 19

3.5.1. Preprocessing 19

3.6 Dataset Balancing Methods. 19
3.7 Clustering Algorithm based on Machine Learning 20

3.5.3. Algorithm for Machine Learning Classification 21

3.8 Model evaluation 22

3.8.1 Accuracy 22
3.8.2 Recall 22
3.8.3 Precision 23
3.8.4 F1 score 23
3.8.5 Confusion Matrix 24
CHAPTER FOUR PROPOSED APPROACH & IMPLEMENTATION 25

4.1 Proposed Approach for BTS fuel level severity prediction Framework. 25
4.2 Proposed BTS Fuel Severity Prediction Preprocessing.....26

4.2.1 Dataset Cleaning.....26
4.2.2. Method for Dataset Balancing.....27
4.2.3 Applied Machine Learning Clustering Method.....27

4.3. Developing Machine Learning Models 28
4.4. Models Evaluation.....29
4.5. Implementation and Experimentation 29
4.6 Implementation Environment.....31
4.8 Description of the data set 31
4.9 Implementation of preprocessing 31
4.10 Deployment for cleaning the data set 32
4.11 Dataset Balancing Implementation.....32
4.12 k-means clustering execution 33
4.13 Implementation of a Machine Learning Model.....35
4.14 Model Assessment and Testing.....37

CHAPETR FIVE RESULT And DISCUSSION..... 38

5.1 Introduction 38
5.2 Result of dataset balancing.....38
5.3 K-means clustering result.....41
5.4 Model Evaluation Result 45
5.5 Discussion.....54

Conclusion..... 56
Recommendation and Future Work..... 57
references..... 58
Appendix A:1 Sample Source Code..... 63
Appendix A.2: A sample code for data visualization..... 67
Appendix B: Tools used..... 75
Appendix C..... 77

LIST OF FIGURES

FIGURE 2. 1 THE IOT ELEMENTS 8
FIGURE 2. 2 EXAMPLE OF SUPERVISED LEARNING (JAISWAL, 2021) 10

FIGURE 3. 1 RESEARCH METHODOLOGY FRAMEWORK. 18
FIGURE 3. 2 INPUT/OUTPUT DATA COLLECTED AND CONVERTED TO SPREADSHEET 20

FIGURE 4. 1 BTS FUEL LEVEL SEVERITY PREDICTION FRAMEWORK 26
FIGURE 4.2 DATASET PREPROCESSING 27
FIGURE 4.3 FLOW DIAGRAM OF THE K-MEANS CLUSTERING FOR THE DATASET 29
FIGURE 4.4 DIAGRAM FOR MODEL DEVELOPMENT 30
FIGURE 4.5 THE DATASETS ARE LOADED USING PYTHON CODE. 33
FIGURE 4.6 PYTHON CODE TO STANDARDIZE DATASETS. 33
FIGURE 4.7 IMPLEMENTATION CODE FOR UNDER SAMPLING AND OVERSAMPLING 34
FIGURE 4.8 CLUSTERING BY IMPORTING NECESSARY PACKAGES 34
FIGURE 4.9 PYTHON CODE TO LOAD A DATASET WITH ONLY SEVERED CLASS 35
FIGURE 4.10 THE ELBOW METHOD IS IMPLEMENTED IN PYTHON CODE. 35
FIGURE 4. 11 PYTHON CODE FOR K-MEANS CLUSTERING 35
FIGURE 4. 12 IMPORTING THE NECESSARY SOFTWARE PACKAGE 36
FIGURE 4.13 SPLITTING A DATASET INTO A TRAINING AND TESTING SET USING PYTHON 36
FIGURE 4.14 FIT THE MODEL AND BUILD RF WITH PYTHON CODE 37
FIGURE 4.15 FIT THE MODEL AND BUILD DT WITH PYTHON CODE 37
FIGURE 4.16 FIT THE MODEL AND BUILD SVM WITH PYTHON CODE 38
FIGURE 4.17 FIT THE MODEL AND BUILD KNN WITH PYTHON CODE 38

FIGURE 5. 1 INITIAL BTS1 DATASET 39
FIGURE 5. 2 BTS1 DATA AFTER BALANCING 39
FIGURE 5. 3 INITIAL BTS2 DATASET 40
FIGURE 5. 4 BTS2 DATA AFTER BALANCING 40
FIGURE 5. 5 INITIAL BTS3 DATASET 40
FIGURE 5. 6 FIGURE 5. 6 BTS3 DATA AFTER BALANCING 40
FIGURE 5. 7 INITIAL BTS4 DATASET 41

FIGURE 5. 8 BTS4 DATA AFTER BALANCING 41
FIGURE 5. 9 INITIAL BTS5 DATASET 41
FIGURE 5. 10 CLASS DISTRIBUTION AFTER BALANCING BTS5 DATASET 41
FIGURE 5.11 BTS5'S ELBOW POINT 42
FIGURE 5.12 BTS5YES CLUSTER GROUP. 42
FIGURE 5.13 BTS1'S ELBOW POINT 43
FIGURE 5.14 BTS1YES CLUSTER GROUP 43
FIGURE 5.15 BTS2'S ELBOW POINT 44
FIGURE 5.16 CLUSTER GROUP FOR BTS2 44
FIGURE 5.17 BTS3 BTS3'S ELBOW POINT 45
FIGURE 5. 18 BTS3 CLUSTER-GROUP 45
FIGURE 5.19 ELBOW POINT FOR BTS4 46
FIGURE 5.20 BTS 4 CLUSTER-GROUP 46
FIGURE 5.21 MODELS COMPARED DEPENDING ON THEIR ACCURACY. 48
FIGURE 5. 22 MODELS COMPARED DEPENDING ON THEIR F1-MEASURE. 49
FIGURE 5.24 A CONFUSION MATRIX FOR RF MODELS 50
FIGURE 5.25 FIGURE 5: CONFUSION MATRIX FOR DECISION TREE MODEL 51
FIGURE 5.26 FOR EACH DATASET, A CONFUSION MATRIX FOR K-NEAREST NEIGHBOR MODELS 52
FIGURE 5.27 CONFUSION MATRIX FOR SVM MODELS FOR EACH SELECTED DATASETS 54

List of Tables

Table 2. 1 Pros and Cons of algorithms. (Johnson B., 2021) 13
Table 3. 1 Confusion matrix 25
Table 4. 1 Implementation Environment 31
Table 4. 2 Dataset attribute 32
Table 5. 1 Cluster group for BTS5 42
Table 5.2 Cluster group for BTS1 43
Table 5.3 Cluster group for BT2 44
Table 5. 4 BT3 Cluster group 45
Table 5.5 Cluster group for BTS 4 46
Table 5.6 RF Performance result. 47
Table 5.7 KNN Performance result 47
Table 5. 8 SVM Performance result 48

LIST OF LIST OF ABBREVIATIONS

AC	Alternating Current
ANN	Artificial Neural Network Current
BTS	Base Transceiver Station
CDMA	Code Division Multiplexing Access
DT	Decision Tree
ET	Ethio Telecom
GSM	Global System for Mobile Communication
IDE	Integrated Development Environment
IoT	Internet of Things.
KNN	K-Nearest Neighbor
LCD	Liquid Crystal Display
LED	Light Emitting Diode
LTE	long-Term Evolution
ML	Machine Learning
MS	Microsoft
MW	Micro Wave
NLP	Neural Language Processor
PCB	Printed Circuit Board
PDF	Portable Document Format
PLX-DAQ	Parallax microcontroller data acquisition
PR	Pattern Recognition
QOS	Quality of Service
RAM	Random-Access Memory
SVM	Support Vector Machine
UMTS	Universal Mobile Telecommunications System
WCSS	within Cluster Sum of Square
WCDMA	Wideband Code Division Multiplexing Access

Abstract

The major infrastructure element in mobile networks is the Base Transceiver Station (BTS), connecting client equipment to the cellular network. As the number of mobile customers and base stations increases, the demand of fuel for the BTS standby generator and installing reservoir fuel tank is also too. However, unexpected fuel running out from BTS is common once the reservoir tank is filled. It exposes the telecom service providers to network interruption when the generator finish fuel. After sensors detect severity levels knowing the magnitude is very important,The severity level of the fuel in the reservoir indicates the effect of unexpected fuel runout and BTS generator shut down. Predicting the stage of fuel level severity is mandatory. Refueling the critical cell towers before it goes to shut down and then can visit the less severed sites after. Different studies have been done on monitoring the fuel level of a BTS, but it is critical to understand the severity state. Fuel level monitoring has been proposed in the past, but not the severity level threshold.

Based on a dataset obtained from five BTS fuel tanks, this work proposes BTS fuel level severity prediction which has 500,1000,2000,3375 & 5000 litter volume capacity with various manual gage levels on it,which defines the amount & severity level of fuel in the tank. Data is collected using hardware tools HC-SR04,Arduinomega2560,PS,& data logger and software which integrate the hardware.The datasets were labelled to suitable severity levels using a clustering machine-learning method. The algorithms are then applied to five datasets, namely called BTS1, BTS2, BTS3, BTS4 & BTS5 which has 4884,5208,5208,5112 & 5112 rows of datasets respectively to construct a model and the performance of each algorithm is measured and compared to select a superior model. Based on the accuracy and f measure, the result shows that the RF model outperformed the DT, KNN, and SVM models on each dataset with a maximum testing accuracy of 92-97.4 per cent and f-measure 92 -97 per cent scored. Furthermore, RF performed better in all datasets than DT, KNN and SVM. The study's findings suggest that supervised machine learning can predict BTS fuel level severity stage. Therefore RF model is recommended for BTS fuel level severity prediction.

Keywords: Fuel level Prediction, Machine learning, Base Transceiver station.

CHAPTER ONE INTRODUCTION

1.1 Background of the study

Telecommunication tower sites are classified into two types; sites in accessible zones and sites in remote zones. The national grid, power generator sets, solar panels, or batteries usually power Base Tranciver Stations in accessible zones as the secondary source. In contrast, sites in isolated zones depend on power generators as the primary source and solar panels or batteries as secondary sources. In India, a study by Intelligent Energy showed the irregular provision of electricity from the national grid to telecommunication tower sites in accessible zones, similar to most African countries. It gives rise to the need for diesel fuel to power the sites as a standby power source (Bhoraskar, (2012)).Ethio telecom was established in November 2010 by replacing Ethiopian Telecommunications Corporation (ETC). The company has had different names since 1894 E.C., during the governance of Emperor Menelik II, when the 407 Km telegraph line between the cities of Harar and the capital Addis Ababa was constructed. Nationally there are about 7556 BTS sites in rural and urban areas. Above 700 of them are found in the capital city, Addis Ababa (ethiotelecom, 2016).ETC holds various telecom devices that need a consistent power source to operate. The company got this power mainly from the national grid and uses this power source alone or with generators and backup batteries. In addition, it uses a generator commonly with a fuel tank, which has 500-5000 litter capacity and backup batteries for off-grid or remote areas. Protecting the power source of telecommunication equipment enables no interruption of power source & communication; increases client satisfaction; reduces revenue loss & ensures the quality of service.

Site operators in Telecommunication Network cell sites monitor generators manually, and monitoring the fuel levels of each generator deployed in a cell site is inefficient. There have been reports of sites being shut down for hours because of the site owners' irresponsible and unethical behavior causing a significant loss to the telecommunications firm. This issue has persisted for a long time and should be taken seriously. The management of mobile base station cell sites is inefficient, resulting in continuous losses for telecommunication network owners. They cannot keep managing cell sites one at a time and expect successful performance. Human beings (Site Operators) are exposed to making mistakes and are not good to be employed to resolve such issues. According to past data, they were responsible for most site failures due to response time delays and inadequate diesel fueling (Edem Donald, 2018).A similar problem exists in Ethiopia's

only telecommunication operator, which uses a generator as an alternative and standby power source in accessible & remote BTS. It also faces high revenue loss due to the manual prediction method of the Base Transceiver Station (BTS) generator's fuel severity level (ethiotelecom, 2021). The study aimed to survey predicting the generator fuel severity level in the sub-regional branch maintenance center of ethiotelecom found in Bale Robe, which has 92 BTS out of which 50 BTS installed with the standby generator. These BTS have a 500 -5000-litter volume capacity fuel tank according to site distance from fuel distribution to the center and alternative power installed(AppendixC).

Unexpected fuel runout from the fuel tank results in power and communication service interruptions that cannot be tackled using the manual prediction method. In addition, the problem will be more difficult as the expansion of BTS and installation of standby genertor increases. Managing the severity level of fuel of a BTS generator can prevent telecom owners from high loss revenue and client complaints (AppendixC). Relatively handling the severity level improves the refueling action and sustainable power to the BTS and communication service for the clients. It is cost-effective and adds value to a reliable power source for BTS.

Fuel level detection using a sensor is useful to collect fuel height in the tank that helps identify the amount of fuel in terms of litter or percent. Fuel level severity prediction involves using two metrics, fuel consumption and Height of fuel or fuel amount in percent (AppendixB).

After the Height of the fuel in the tank is detected, predicting the severity of the fuel level in the resrvior requires classifying and clustering the treshold level. Otherwise, the paper categorizes the treshold level based on their severity height level or fuel volume percent. It is concerned about determining fuel severity level risk based on their treshold level and the impact they cause on the BTS power and communication service.The following are classifications of BTS generator fuel tank severity levels based on the assessment of questionnaires & physical visit made about the manual method (AppendixC).

- Normal: fuel level in height ≥ 3/8
- Minor:¼<of the tank'scapacity ≤3/8 .
- Major: when 1/8 < of the tank's capacity ≤ ¼ indicating immediate refueling.
- Critical: When the generator's fuel level falls below 1/8 of the tank's capacity, the BTS
- is on the verge of shutting down.

Research had been done to monitor the fuel level in time intervals to know fuel level status, but after detecting the level, it is necessary to predict the severity level. The severity level enables to refuel before volume drop to critical stage, prioritization of the BTS which has a high impact on service coverage and the revenue it brings. By studying severity stages and classifying them as Critical, Major, Minor, and Normal, the study aims to provide a way to predict BTS standby generator fuel level severity

This research work is based on collected data from five different BTS generator fuel tank namly called BTS1, BTS2, BTS3, BTS4 & BTS5 which has 4884,5208,5208,5112 & 5112 rows of datasets respectively by using developed prototype, interview and physical observation. The dataset contains features BTS ID, Elapsed hour, Date, BTS distance from the center, the Fuel level in the litter, the Fuel level in percent, the height of fuel level(cm) in the tank, & Class features. which measures the fuel's severity level and suggests a selection of machine-learning algorithms for comparing and selecting the best prediction modelThis research is motivated by observing inefficient fuel level severity monitoring and refuelling activities by the Bale Robe sub-branch local telecom company. This company uses a manual labelled gage on the fuel tank & hand touching method to measure and determine the fuel level of the diesel generator tank installed in every BTS as depicted in figure 1.1 below .

Figure 1. 1 Manual fuel level determining of BTS diesel generator fuel tank

1.2 Motivation

The company recently lost birr 567,941 in an hour after the site shut down due to grid power off & standby generator running out of fuel from its tank. This causes network interruption, which affects the day-to-day activities of different business clients and mobile users (ethiotelecom, 2021). Again the company had lost 1.5 ml. birr in 14hour due to the site shut down on date 11/07/2020 due to similar unexpected fuel runs out of the reservior (ethiotelecom, 2021). The problem had occurred because of the absence of a site guard who is expected to inform or warn centres about refuelling the generator. Therefore, this problem is inspiring to study sites to determine the severity of the fuel level of the power generator tank using prediction techniques, thereby recommending its implementation by the company.

1.3 Statement of the problem.

Unexpected fuel runout from BTS fuel tank expose telecom operators to high revenue loss and service interruption. Early detection and refuelling of the fuel reservior before it reaches to severe level is essential. Data is collected by sensors and analyzed to handle the severity & making it appropriate for prediction. On the other hand, refuelling all critical BTS simultaneously is difficult. Therefore, fuel level prediction is important to identify the severed BTS that should be prioritize refuling of its fuel tank. It helps the BTS refuelling administrators to give attention to cell towers which are in a state of critical fuel level. This work helps to predict the magnitude of the severity level of fuel in four levels through machine learning algorithms.The study addresses the problem by answering the following questions:-

- How can data be automatically detected and collected from the fuel tank of a BTS fuel reservior ?
- How can the collected dataset's biased classes be balanced?
- How to prepare a data set for predicting fuel level severity from collected data sets that do not have a treshold level?
- Which machine learning algorithm better in predicting severity?

1.4 Objectives of the Study

1.4.1 General Object

The general objective of this work is to predict the severity of fuel level in BTS using machine-learning techniques.

1.4.2 Specific Objective

The specific objectives are:-

- To examine the literature on fuel level predictions and the use of machine learning in prediction.
- To collect data from BTS generator fuel tank using hardware and software tools.
- Preprocessing data sets to make them appropriate for solving the research questions.
- To create models based upon the type of machine learning algorithms.
- To assess the model's efficency.

1.5 Scope and Limitations of the Study

1.5.1. Scope of the Study

The study focuses on predicting the extent of the severity of fuel level BTS using collected data through sensors. The dataset with severity class level is clustered into Minor, Major and Critical. The dataset consists of BTS ID, Elapsed hour, Date, BTS distance from the center, the Fuel level in the litter, the Fuel level in percent, the height of fuel level(cm) in the tank, & Class features. The paper did not consider the first six-dataset features, as they are not necessary for solving the research questions formulated. The model is trained and tested based on these class labels. That means the model's performance is evaluated with Accuracy, Precision, recall and f-measure metrics.

1.5.2. Limitations of the Study

Selected branch of the local telecom operator Environmental and power section staff have provided with the current manual trend in the company. However, samples were taken after expressing the great benefit of the result of the research. All BTS diesel generators with fuel tankers are installed in different areas that cannot be covered simply to get installed fuel tank size and distance from center , because they have been installed across the country's wide area and are

costly. Therefore, the scope of this study is restricted to a sub-regional branch maintenance center.

1.6 Significance of the study

Observing the problem in one of the local telecom operators' sub-branch areas, namely Bale-Robe, this paper attempts to develop solutions to the problem that exposed the company to lose significant revenue because of the cell tower shutdown due to unexpected running out of fuel from BTS fuel reservoir.

1.7 Thesis Organization

This thesis work consists of five chapters. Chapter 1 presents the thesis work's introduction. Chapter 2 goes through a literature review of related jobs, reviews related literatures regarding BTS, IoT, Machine learning, research gaps and summarizes the critical parameters and diesel generator fuel monitoring in BTS. Chapter 3 is concerned about methodology of the study. Chapter 4 states the implementation of the proposed system. In this chapter, topics such as modeling the solution, its result, discussion and evaluation are covered. Chapter 5 presents conclusion, recommendations and future works.

CHAPTER TWO LITERATURE REVIEW

2.1 Introduction

This chapter examines relevant literature related to the issue of investigation or the research problem. The Internet of things (IoT) idea is described and how machine learning can determine fuel sevrity levels in the tank.

2.2 Internet of things

The concept "Internet of Things (IoT)" has lately gained popularity in communication technology. It has been developed in various methods and is known as the next frontier. The IoT device is set to transform many aspects of our life. Several characteristics of our existence alter our world. The number of IoT devices is likely to skyrocket in the following years. IoT has a present reach of more than 12 billion items that can connect to the Internet, but by 2020, it is expected that there will be 26 times more linked things to the Internet than people (K.R. Darshan and K. R. Anandakumar, 2015).

After the Internet, the Internet of Things (IoT) is regarded as a technology and economic wave in the global information industry. The Internet of Things (IoT) is an intelligent network that connects all things to the Internet to exchange information and interact via information detecting devices that adhere to agreed-upon protocols. Its purpose is to intelligently identify, locate, track, monitor, and manage things (; J. A. Stankovic ". D.–9., Feb. 2014.).

It is an extension and expansion of an Internet-based network that extends communication from humans and humans to humans to things or things to things. In the Internet of Things (IoT) concept, many of the objects in our environment will be linked together in networks. RFID, sensor technology, and other intelligent technologies will be used in various applications**.**

2.3 Architecture of IoT

According to the traditional definition of IoT, computers, sensors, and objects communicate and process data; hence, we can say that IoT is a new technology system that combines many information technologies. The Internet of Things is a semi-autonomous network that incorporates many technologies. It links individual devices to the network and one another.

There are also controller systems (software and services) in the network that operate as brains for data processing by analyzing and utilizing data acquired by connected devices to make decisions and initiate actions from the same or other instruments (J. A. Stankovic, 2014.).The central objective of IoT is to enable us to uniquely identify, signify, access and control things anytime and anywhere by using the Internet (Miller, 2015). Furthermore, the interconnected device networks can result in many intelligent and autonomous applications and services, bringing significant personal, professional and economic benefits (Li, 2019).

Figure 2. 1 The IoT Elements

2.4 Fuel level monitoring and IoT

IoT-based technology has now found its way into many consumer gadgets. Many businesses rely on diesel generators as a backup power source. One of them is the telecommunications industry. This entails a massive quantity of fuel consumption for their power needs. Fuel level monitoring solutions use IoT-enabled sensors to allow users to remotely fuel tank levels and identify issues, like leaks or excessive idling, and maximize fuel efficiency. Furthermore, it can alter several parts of the manual refueling operation. Many fuel-monitoring programs are now available, allowing refueling employees to check the fuel status in the tank and plan refueling via mobile phones and smart devices, eliminating the need to call the site operator or guard (Li, 2019).

2.5 IoT in BTS Diesel Generators fuel level monitoring.

IoT transforms the sensed data into insights for more brilliant follow up of fuel levels, current, voltage & other parameters that exist in distributed BTS. For example, diesel generators are commonly used as a temporary source of power in domestic and industrial applications. However, in the case of domestic systems, we do not have any monitoring systems for the fuel level except the analogue value that would be placed outside the diesel generator. Therefore, IoT is essential in monitoring diesel generator fuel level & another parameter, which is found in distributed BTS. By leveraging devices like connected sensors and other types of things that information can be placed in the cloud or own server, refueling personnel or site operator can easily monitor the real-time status of the distributed BTS generator fuel level & other power parameters (Jaiswal, 2021).

2.6 achine Learning

IoT transforms the sensed data into insights for more brilliant follow up of fuel levels, current, voltage & other parameters that exist in distributed BTS. For example, diesel generators are commonly used as a temporary power source in domestic and industrial applications. However, in the case of domestic systems, we do not have any monitoring systems for the fuel level except the analogue value that would be placed outside the diesel generator. Therefore, IoT is essential in monitoring diesel generator fuel levels & another parameter, which is found in distributed BTS. By leveraging devices like connected sensors and other types of things that information can be placed in the cloud or own server, refueling personnel or site operator can easily monitor the real-time status of the distributed BTS generator fuel level & other power parameters (Jaiswal, 2021).

2.6.1 Supervised Machine Learning

Machine learning is supervised learning if the instances are given with known labels. The way to be managed learning is learning the function or algorithm to map an input to output without explicitly programming. The dataset contains labelled results in supervised machine learning (SML), and an algorithm learns from the training dataset (Jaiswal, 2021). SML includes different machine learning techniques such as regression and classification. The supervised machine learning technique focuses on classification and regression problems. Classification algorithms classify data objects into other classes based on the points' characteristics with their labels. For example, the genotype first clustering algorithms are implemented in the group, and the result of clustering algorithms called labels are used for the classification algorithm training. Some popular classification algorithms include logistic regression, naïve Bayes, K-nearest neighbour (KNN), support vector machine (SVM), and random forest(RF).

2.6.2 Supervised machine learning algorithms

May utilize labelled examples to apply what they have learned in the past to new data to predict what will happen in the future. Based on the analysis of a given training dataset, the learning algorithm generates an inferred function to expect the output values after a suitable amount of training, and the system is ready to use. I will provide objectives for any new input. The learning algorithm can also discover errors by comparing its output to the proper, intended outcome and adjusting the model.

Unsupervised machine learning approaches, on the other hand, are used when the Data being taught is not supervised. It isn't classified or labeled in any way.Unsupervised learning is the study of how computers may deduce a function from unlabeled data in order to explain a hidden pattern. The system does not determine the correct output but examines and infers private networks from unlabeled data using datasets .(Jaiswal, 2021).

2.7 Supervised Learning Application

In supervised learning, models are trained using a labelled dataset, where the model learns about each category of input. After the model has been introduced, it is assessed with test data (a subset of the training set) and then predicted the outcome. The following diagram and illustration will assist us in comprehending how supervised learning works:

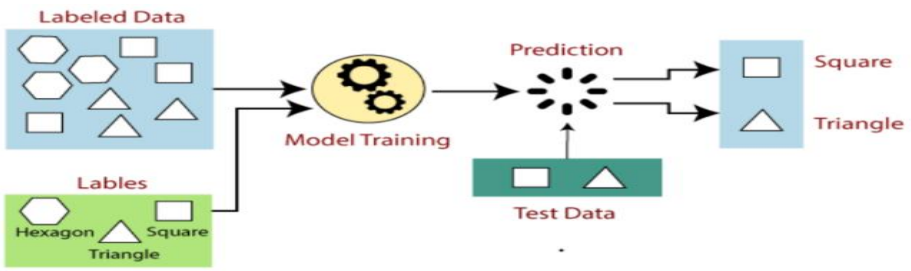


Figure 2. 2 Example of supervised learning (Jaiswal, 2021)

Assume we have a dataset with various forms, such as squares, rectangles, triangles, and polygons. The model must now be trained for each shape as the initial phase.

- If a given form has four sides and all are equal, it is referred to as a Square.
- If a particular shape has three sides, it is a triangle.
- When a form has six equal sides, it is a hexagon.

The test set is used to put our model to the check after it has been trained, and the model's goal is to recognize the form. The system has already been trained on various forms, so when it encounters a new one, it classifies it based on several sides and predicts the outcome.

2.8 Kinds of supervised Machine learning

Supervised machine algorithms can be further subdivided into two categories of issues:

2.8.1 Regression

If there is a relationship between the input and output variables, regression procedures are applied. It's used to forecast continuous variables like weather, market trends, and so on.Some popular Regression algorithms that fall within the category of supervised learning are listed below: (Jaiswal, 2021)

- a Linear regression
- Regression tree
- Non linear regression
- bayesian linear regression
- polynomiyal regression

2.8.2 Classification

When the output variable is categorical, classification procedures are used. That is, there are two classifications. : Yes-No, Male-Female, True false, etc.

Logistic Regression. - The outcomes are classified into one of two groups. The dependent variable is always categorical, regardless of whether the independent components are categorical or numerical.

P(Y=1|X) or P(Y=0|X) is written in this format.

Given the independent variable X, it determines the probability of the dependent variable Y.This might help you figure out whether a word has a positive or negative connotation. (zero, 1, or on a scale between). It can also be used to discern the object in a photograph (tree, flower, grass, etc.), with each thing assigned a probability between zero and one (Jaiswal, 2021).

Naive BayesThe probability of a data point falling within a given range is evaluated using Naive Bayes.Does or does not belong to a specific category.Words or phrases can be classified as belonging to a predetermined "tag" (classification) or not using in-text analysis.Consider the following example:

$$P(A/B) = \frac{P(B/A) \times P(A)}{P(A)}$$

P (A) = Class Prior Probability

P (B/A) = Likelihood

P (A/B) = Posterior Probability

P (B) = Predictor Prior Probability

Alternatively, if B is true, the probability of A is equal to the possibility of B; if A is true, times the chance of a being true, divided by the probability of B being actual (Wolff, 2020).

K-nearest neighbors (k-KNN). - Is it a pattern recognition system that uses training datasets to predict the k closest relatives in future cases.When you use k-KNN for classification, you calculate to place data in the category of its nearest neighbor neighbor. If k = 1, It will be assigned to the class that is the nearest to it.A majority of its neighbors classify k in a survey. (Rachel, 2020).

The decision Tree. - A decision tree is a supervised learning technique that is ideal for classification issues because it can organize classes precisely. It works like a flow chart, splitting data points into two related categories at a time, starting with the "tree trunk" and progressing to "branches" and "leaves," where the categories become more finitely similar. This produces sorts, allowing for organic classification with minimal human oversight (Rachel, 2020).

Support Vector Machines. - A support vector machine (SVM) employs methods to train and classify input within degrees of polarity, going beyond X/Y prediction. We'll utilize red and blue tags, together with two data features: X and Y, to provide a basic visual description and then train our classifier to output an X/Y coordinate as red or blue. The SVM then determines which hyperplane best separates the tags. This is just a line in two dimensions. Everything on one side of the line is red, while everything on the other is blue. This would be favorable and negative in sentiment analysis; for example, the ideal hyperplane for maximizing machine learning is the one with the greatest significant distance between each tag: The more complicated the data, the more accurate the predictor will become when using SVM. . For example, imagine the above in three dimensions, adding a Z-axis, which turns into a circle. When the best hyperplane is used to map it back to two sizes, it appears to be as follows: SVM enables more precise machine learning because it is multidimensional.

Table 2. 1 Pros and Cons of algorithms. (Johnson B., 2021)

Alg.	Advantage	Disadvantage	Application area
LR	Simple to apply, extend several classes and quickly identify unknown records.	If the number of observations exceeds the number of features, overfitting occurs. Linearity for both dependent and independent variables.	It is utilized to assess patterns and make projections or forecasts in business.
KNN	There is no assumption of data, easy to implement, no training data period, & high accuracy.	It does not work well with high dimensional data, Needs feature scaling, sensitivity to the implanted dataset, poor interpretability.	Text mining, Agriculture, Finance, Medical, Facial recognition, Recommendation systems (Amazon, Hulu, Netflix, etc.)
DT	Easier to understand, requires less code, does not require scaling, models non-linear decision boundaries.	Higher time for modeling minor modification in the data can result in a massive shift in the structure of the data accessible to overfitting instability.	applied in real life in numerous fields, including engineering, civil engineering, law, and business
SVM	Effective with high dimensional spaces and handle non-linear, imbalanced data, and result in increased accuracy	It does not work well when there is a lot of traffic. Noise and overlapping data, which is not suitable for extensive data.	Applications,like. Handwriting,recognition, intrusion,detection,face detection,email classification, gene classification, and web pages.
Naïve Bayes	Easy to implement, the training period less, work with small to extensive data.	The assumption of independent predictors assumes all attributes are mutually independent.	Predict the probability of different classes based on various attributes. This algorithm is mainly used in text classification and has multiple Categories problems.
R.F.	Reduce overfitting & improve accuracy, work in high dimensional data, automate missing values in the dataset.	Complexity happen because R.F. creates many trees, requires more time for modelling.	They are used in various applications, for example, to provide recommendations of different products to customers in e-commerce. A random forest algorithm can identify the patient's disease by analyzing the patient's medical record in medicine.

2.8.3 Advantages of Supervised learning

In addition, the model can predict the outcome based on previous experiences using supervised learning.A supervised learning model helps us solve real-world problems such as fraud detection, spam filtering, etc. (Johnson, 2021).

Disadvantages of supervised learning

- For dealing with complex tasks, supervised learning methods are ineffective.
- If the test data differs from the training data, supervised learning will be unable to predict the correct outcome.
- Training required a lot of computing time. Enough understanding of object classes is required for supervised learning. (Johnson, 2021)

Based on the problem statement and the research question, the method is intended to measure and predict the real-time fuel level range in the fuel tank. In the case of that classification, supervised machine learning is suitable for prediction purposes. On the other hand, regression led machine, learning is used for casting continuous variables.

2.9 Related works

Google Scholar, Scopus, IEEE Explore, and Digital Library searches were used to find related articles. IoT based fuel level monitoring,' 'IoT based load monitoring,' 'water level monitoring,' 'prediction of a defect in load Transformer,' 'neural network for water level prediction and algorithm for fundamental IoT architecture were among the search phrases utilized. This section presents earlier works done by different authors.

(Sugumar, 2020) Developed a system that would monitor the fuel level of a diesel generator's external tank and send a warning message to mobile phones through GSM when the fuel level drops below the base value. It also provides a time value for the fuel level on the website for future reference that all information be saved in the cloud. The author Used IoT to create a load monitoring system for diesel generators. They present a system that can monitor the diesel generator's fuel and battery levels by monitoring several characteristics such as current, voltage, operation hours, and fuel level in their research. In addition, intelligent sensors based on hydrostatic capacitive measured principles are included.

(M. Sokolova, 2018) Also proposed a system based on IoT to control the fuel and battery level of diesel generators installed in the Base Transceiver Station (BTS). Observing inefficient and manually maintaining power and battery level of diesel generators of telecommunication cell sites

exposed to shut down four hours because of carelessness and unprofessional training. Their proposed system comprises both hardware & software like transmitter, ultrasonic sensor, battery sensor, and IoT technology.

Mohammad Asif Khan, (2017)) Design a system to monitor solar plants for higher performance. According to these authors, solar plants are situated in remote access locations, according to their research. Therefore, they are vulnerable to malfunctioning solar panels, connections, and dust-covered panels, reducing output and influencing solar performance. They demonstrate a hardware-based technique with an AT mega microcontroller, current and voltage sensors, a Wi-Fi module, and algorithms. Mohammad et al. conducted a work entitled Context-Aware Fuel Monitoring System for Cellular Sites. The authors' report indicates that manual fueling and supervision of sites exposed fuels purchased to theft which increased the fuel purchase by 25 per cent. To tackle this problem, a system is proposed which uses a microcontroller.As the authors explained, a report on power quantity is generated monitoring continuously the gasoline level. The site engineer can remotely keep an eye on the gasoline level in the tank and manage the generator runtime. Their research report shows that this system's prototype was developed utilizing an Arduino Mega 2560 (microcontroller), an ultrasonic sensor (volumetric sensor), relays (for remotely starting and stopping the generator), and a GSM SIM900 module (used to communicate with site engineer).

(Hiremath, 2015)Also conducted a study titled “Intelligent fuel theft detection using microcontroller” and proposed a system that monitors fuel theft in generator sets due to the global rise in fuel prices. The approach the authors proposed used GSM, a technology for data transfer in areas where the Internet is not available. A flow sensor was utilized to collect data on the amount of fuel consumed in the engine, GSM was used to report theft, and GPS (Global Positioning System) was used to follow the vehicle. Based on the research report, this study also sought to resolve the problem of stolen vehicles alongside fuel pilferage. As the result shows, the GSM and GPS technology allows users to monitor fuel levels and receive notifications for stealing fuel. A vehicle SIM900A was used to create a prototype (GSM module), microcontroller ARM7, and GPS module, LCD for display and keypad to enter the device's password.

(Sayali Barde, Pakhal, & Josh, 2017)In their paper “GSM-based fuel level monitoring system” stated that fuel theft is a growing problem in today's vehicles.It has an impact on financial loss.We're installing a wireless fuel level monitoring system to avoid this.The Hall Effect sensor and GSM modem are used in this system.The Hall effect sensor detects the amount of fuel in the

vehicle as well as the amount of fuel utilized.The PIC 18F458 in this system informs thowner of the amount of fuel remaining in the tank.Microcontroller, GSM module, LCD, and keypad are all part of the system. If there is a significant difference in the fuel level, the fuel level sensor will send a signal to the microcontroller, and the user will receive information in text format via the microcontroller.This smart technology allows users to obtain gasoline 24 hours a day, seven days a week, and sends out alerts when the tank's fuel level drops unexpectedly.

According to (Edem Donald, 2018)to address the unprofessional behavior of the cell site operator monitoring numerous generators, our work established a GSM-based remote monitoring procedure. Both hardware and software are included in the proposed system. The hardware part (transmitter section) includes ultrasonic sensors, battery sensors, microcontrollers, GSM modules, power supplies, and other components, whereas the reception section includes the site operator's mobile phone. The algorithm and program code are written in the C++ programming language in the software section. The Trans mitter for the system was meticulously designed and tested. The results demonstrate that the created system perfectly monitored the fuel level and generator battery health in the targeted cell site and wirelessly transmitted the status (parameters) to the site operator's mobile phone.Cell sites can be maintained by remotely monitoring the generators that power the Base Stations and alerting the site operator.

Therefore, all of the above stated researches have focused on fuel & related parameter level monitoring. The suggested method will deal with predicting the severity of fuel levels in BTS, using a standard classification of BTS fuel severity that includes critical, major, minor, and Normal stages

CHAPTER THREE METHODOLOGY

3.1 Overview

This chapter discusses the methodology, tools employed, data collection methods, data preparation strategies, and data assessment techniques. To meet the study's objective, we go over the dataset in detail, model selection, and evaluation methodologies for the algorithms.

3.2 Research Procedure

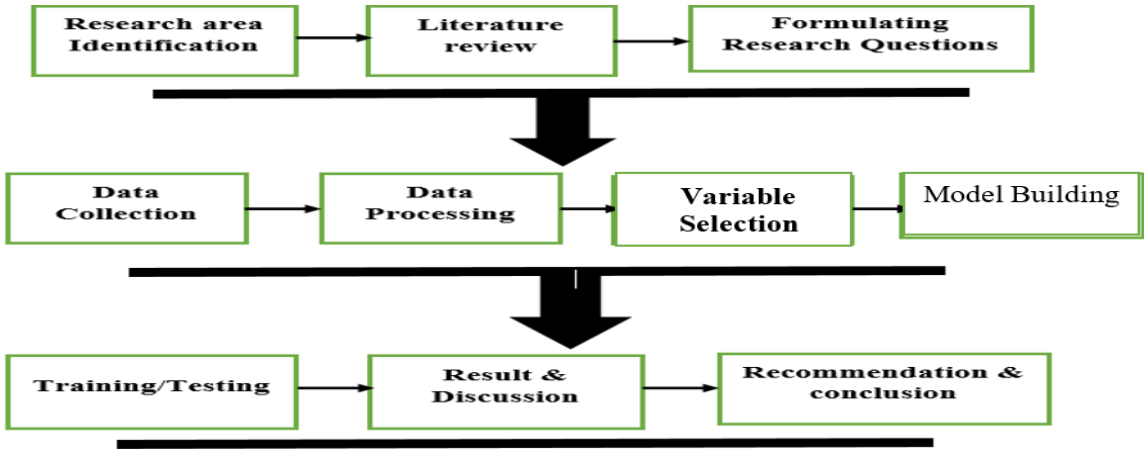


Figure 3. 1 Research Methodology Framework.

3.3 Tools Used

Throughout this research, study, hardware and software resources were employed to accomplish the suggested work's goal of implementing data preprocessing and generating diagrams quickly and precisely.

3.3.1 Hardware tools

- P.C. Intel core i7
- RAM 8 G.B.
- Hard Disk: 1T
- HC-SR04 fuel sensor.
- SD card module
- Dual band 900Mhx
- Atmega2560 Microcontroller

3.3.2 Software tools

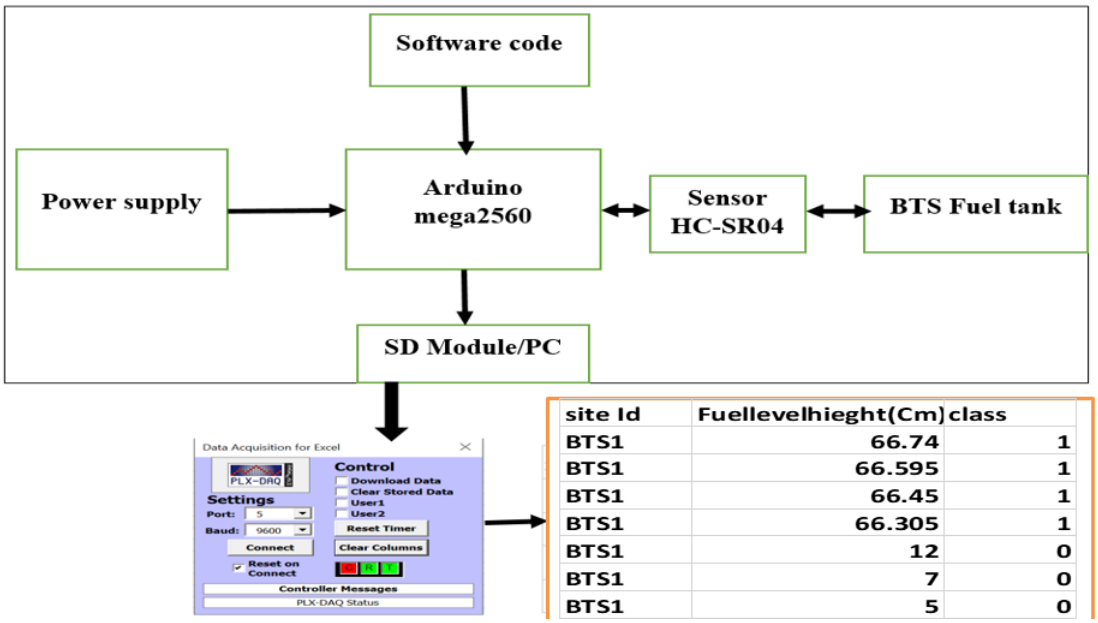
- Arduino version1.8.13
- Proteus 8.11
- E-draw max: version 7.9
- Anaconda navigator python version 1.9.1.2
- Jupyter notebook version 6.0.1
- Microsoft Excel 2010 Used for data preparation
- Scikit-learn version 0.21.3
- Numbly Version .16.1
- Matplotlib 3.1.1
- Seaborn 0.9.0

3.4. Data collection

Questionnaire and observation methods, as well as data from sensors, are used in this study. On the other hand, data for this study is collected from two sources. One is from an employee of Ethiotelecom Company working at the Bale Robe branch. This data is required to determine employees' methods and procedures in refuelling BTS generator fuel tankers. Hence, according to the questionnaire respondents, whose number is twelve (12), the company has 92 BTS, out of which 50 are installed with a power generator. These BTS fuel tank are five types based on their holding volume in a litter. These include 500,1000,2000,3375 & 5000 litter capacity. In addition, the respondents confirmed that they use manual BTS fuel severity determination by measuring the height of fuel volume in the tank. The informants further indicated that when the height of fuel in the tank is below 1/8, the severity level is assumed to as critical; when the height of fuel in the tank is between 1/8 and ¼, the severity level is Major, and when the height of fuel in the tank is between 1/4 and 3/8, the severity level is Minor, and when the height of fuel in the tank is greater than 3/8, the severity level is called as Normal.

The second one is using hardware and software tools like Arduino mega2560 integrated with HC-4 ultrasonic to detect the height of fuel in the tank, SD-card module to record data, GSM-module sensor to transfer data and 5V power supply connected to the microchip to power the system and software Arduino version 1.8.13 & c++ code to integrate the hardware. The physical parameter detected by the sensors and quantitative data collected then changed into an analogue signal. The

values of sensors are captured into a personal computer and SD card module through the Arduino interface. The input/output data collected through these sensors are stored in a spreadsheet using Arduino IDE and PLX-DAC application, as shown figure 3.2 below. The collected data as a spreadsheet file is converted to .csv files as a training dataset and test data set. 4884,5208,5208,5112 & 5113 data is collected from BTS1, BTS2, BTS3, BTS4 and BTS5 respectively. The following figure shows the sample of data collected from the generator's fuel tank within above four months.



site Id	Fuellevelhieght(Cm)	class
BTS1	66.74	1
BTS1	66.595	1
BTS1	66.45	1
BTS1	66.305	1
BTS1	12	0
BTS1	7	0
BTS1	5	0

Figure 3. 2 Input/output data collected and converted to spreadsheet

3.5. Modeling for Predicting Fuel Level Severity

3.5.1. Preprocessing

Feature engineering of data is necessary for making data appropriate for a machine-learning model. It increases the performance of a machine-learning model. Preprocessing methods have been used to clean the dataset, handle missing values, balance the biased classes, and cluster the dataset to the appropriate group of class labels for prediction.

3.6 Dataset Balancing Methods.

The majority of resampling procedures are used to balance biased classes for classification. When the quantity of values in one Class is significantly less than those in another, class imbalance issues

arise. With unbalanced data, prediction models are skewed toward the majority. As a result, the model will generate erroneous findings. (O. Arbelaitz, 2019). The data set employed in this research had a significant imbalance between normal and severe level cases. There are examples at both the normal and severe levels. The majority class is depicted as normal, whereas the severity classes are depicted as severity classes. To balance the normal and severe classes of the data set in this investigation, a combination of under-sampling and oversampling strategies were applied.

Under-sampling: is among the most often utilized approaches for dealing with unbalanced data.The algorithm uses this strategy to try to achieve balance by deleting the bulk of class instances from the datasets at random. As a result, the majority of the samples in the dataset will be removed. This method will be performed until each Class's number of instances is equal. (R. Mohammed, 2020) (Pikes, 2020).

Oversampling: To balance the data set, this strategy replicates minority classes at random. It adds many duplicates of the minority class instances to the training data by selecting random examples of the minority class with replacement. Oversampling is a non-heuristic method that randomly replicates examples in the minority class; this study uses a combination of both methods, which results in better performance; the concept is to apply some oversampling to the minority class to improve bias, while applying some under-sampling to the majority class to reduce bias.

3.7 Clustering Algorithm based on Machine Learning

There is no information about the severity of the data sets used. The model requires labeled data to train in order to predict severity levels. A goal value that describes the severity level of a BTS generator fuel tank should be included in the data. Only a few classes are required from the dataset fuel levels.Clustering algorithms based on severity metrics values are used to categorize datasets into distinct severity levels.The Cluster technique is used to group data points with similar characteristics together.The program analyses the data and finds the groups of data points that exist in the data. Instances within the same clusters are similar, while those found on other clusters are different.By examining our data and applying the algorithm, we can see which groups the data points may be classified into. (Johnson D. , Guru99, 2022), (Self, 2018) (Jemal & Abebe, 2021).

K-means clustering algorithm in this study, it was utilized to group the data into appropriate classifications. It is the most commonly used clustering algorithm because it is simple to construct, comprehend, and present. The K-means method divides the data into k clusters, each with its own set of k values. It divides each data point into a single group, with data points in the same group

being similar. A single location called a centroid will be defined for each cluster, and these centroids must be placed properly. When there are no points pending, early grouping is used to associate each data point with the nearest centroid. After that, new centroids will be calculated as the centers of the clusters formed in the previous stage. As a result, a new relationship between data points and new k centroids is established. This will result in an iterative cycle in which those centroids will shift their positions until no changes are made or they cease moving. (Dabbura, 2018), (board, 2019).

3.5.3. Algorithm for Machine Learning Classification

The study's goal is to use machine-learning techniques to forecast the severity level of BTS generator fuel.Multiple supervised machine-learning algorithms have been employed to achieve this goal, and their performance has been compared.The prediction model is built using the following machine learning algorithms.The algorithms were chosen for their high performance and widespread use in prediction issues. (Upasana, 2022) (Jemal & Abebe, 2021).

Decision Tree: The output of the decision is mapped using a branching structure in decision trees. The decision node, branches, and leaves are the three main elements of the tree. An internal node represents the dataset's attributes, while leaves indicate decision rules and outcomes.

Starting with the root node, the algorithm compares the node to actual dataset features, and then follows a branch based on the comparison. The algorithm will compare the qualities with the sub nodes until it reaches the leave nodes, at which point it will stop. (Hurwitz & Kirsch, 2018).

Random Forest: is a type of supervised machine learning that builds and combines numerous decision trees to generate forests. When constructing the tree, it employs bagging. It chooses the best feature from a random subset of features while splitting the tree, rather than searching for the most significant characteristics. It is simple to calculate the relevance of each feature in the prediction in RF; it is quick and can handle a high number of attributes in datasets. (Shin, 2021).

Support Vector Machine: In classification issues, the support vector machine is commonly utilized. The method classifies the data by creating optimal hyperplane, which are n-dimensional decision boundaries that divide the data points into two categories. The technique selects extreme data points known as support vectors to form a hyper-plane. The algorithm's main goal is to find a hyper-plane that can help split those vectors such that one group of the target variable is

categorized on one side of the plane, while the other side has the second target group. Because SVM is a binary classifier, and the dataset we are using has multiple labels, we employed a multiclass implementation of SVM dubbed one versus the rest to classify our dataset in this study. The number of class's c will be generated as a binary SVM classifier in the one-versus-the-rest technique; the classifier will reduce to a two-class problem by discriminating one class from the other. (Ray, 2017).

K-nearest Neighbor: The KNN method allocates data points to the nearest neighbors, which are the most labeled instances, in order to categorize new instances. The KNN algorithm is non-parametric, makes no assumptions about data, and is a lazy learner algorithm, acting on the data set at the time of classification rather than learning from the training set. The procedure calculates the Euclidean distance between the k-neighbors after allocating the number of k-neighbors.The k closest 26 neighbors will be chosen based on the distance result.The number of data points in each category will be counted from these neighbors, and a new instance will be issued to the category with the most k neighbors. (Jain, 2022).

3.8 Model evaluation

At last, the models and feature selection techniques are compared to determine the best model and feature selection technique for the datasets. The models in the analysis are evaluated using four metrics: Precision, Recall, F1-score, and Accuracy. All four are given equal weight rather.

3.8.1 Accuracy

One of the most extensively used categorization performance indicators is accuracy .It is defined as a ratio of correctly identified samples to the total number of cases, as shown below..

$$\mathbf{ACC} = \frac{\mathbf{TP+TN}}{\mathbf{TP+TN+FP+FN}}$$

The letters P and N stand for positive and negative samples, respectively. The misclassification rate (ERR) = 1-ACC= (FP+FN) / (TP+TN+FP+FN) is the complement of the accuracy metric.. The metric measures the number of samples that have been misclassified into both positive and negative categories, and it is calculated as follows*:* (A.P. Bradley).

3.8.2 Recall

The recall rate, also known as the actual positive rate, is the percentage of accurately detected positive examples among all positive models. The RAM is computed earlier with the contingency table defined.

$$\textbf{Recall} = \frac{\textbf{TP}}{\textbf{TP+FN}}$$

The recall score is also known as sensitivity in anomaly detection or defect identification scenarios, with the positive class denoting abnormal examples. The recall setting in the detection setting, intuitively, measures how sensitive the model is in detecting irregularities or defects among samples with normal functioning.

3.8.3 Precision

The accuracy score assesses how well the classifier accurately recognizes positive samples while misclassifying negative ones as positive.As a result, the precision score is derived as the percentage of models accurately predicted to be positives out of all the expected positive cases (correctly or incorrectly).Thus, using the contingency table, the precision score is calculated as

$$\textbf{Precision}=\frac{\textbf{TP}}{\textbf{TP+FP}}$$

3.8.4 F1 score

There are two types of scores: recall and precision.The F1 score, sometimes known as the F1 score, is an average of these two scores.The harmonic mean of precision and recall is used to calculate it.

$$\text{F1}=\left(\frac{\boldsymbol{Recall}+\textbf{Precision}}{\textbf{2}}\right)^{-\textbf{1}} \qquad = \frac{\textbf{2}\boldsymbol{Precision\ Recall}}{\boldsymbol{Recall}+\boldsymbol{Precision}}$$

We acquire a single score to report and evaluate the model's performance by combining both measures.In binary classification issues with imbalanced classes, for example, it is preferable to think about the model in terms of its F1 score rather than its accuracy score.

3.8.5 Confusion Matrix

The classification model's prediction results are displayed in a confusion matrix, which is a table.It summarizes the right and faulty prediction values by comparing the actual values to the predicted values by the classifier.It also shows how the classifier became confused during prediction, as well as the errors that were made, the categories of errors that were made, and how the classifier became confused.For four severity level class models, a confusion matrix is used in this study (Critical, Major, Minor, and Normal).In a sample, the confusion matrix for such classes is presented in the table below. (Jemal & Abebe, 2021, p. 27).

Table 3. 1 Confusion matrix

	Predicted				
		Critical	Major	Minor	Normal
Actual	Critical	TPC	FP	F Critical	Formal
	Major	F Critical	TN critical	F MINOR	Formal
	Minor	F Major	F Minor	TN Major	Formal
	Normal	F Minor	F Minor	Critical	TN Normal

CHAPTER FOUR PROPOSED APPROACH & IMPLEMENTATION

4.1 Proposed Approach for BTS fuel level severity prediction Framework.

The recommended architecture predict BTS fuel tank treshhold level into Critical, Major, and Minor.It takes the input from collected dataset stated in figure 3.2 and then preprocessed it by cleaning and removing not usable records from the dataset. A combination of the under-sampling and oversampling method is used to balance the severed and non-severed classes of the dataset. After balancing the dataset the severed classes of the dataset have been clustered using k-means clustering machine learning algorithm to group the severed classes of the dataset into severity level (Critical, Major,Minor). The resulting group of the fuel's severity level mixed with the rest of the non-severed fuel level, which has '0' target class indicating non-severe fuel level.Models constructed utilizing RF,KNN, SVM, and DT Machine Learning algorithms after we have the training set available with the necessary class labels.The best model is selected after the models have been evaluated using the various assessment metrics outlined in the previous chapter.

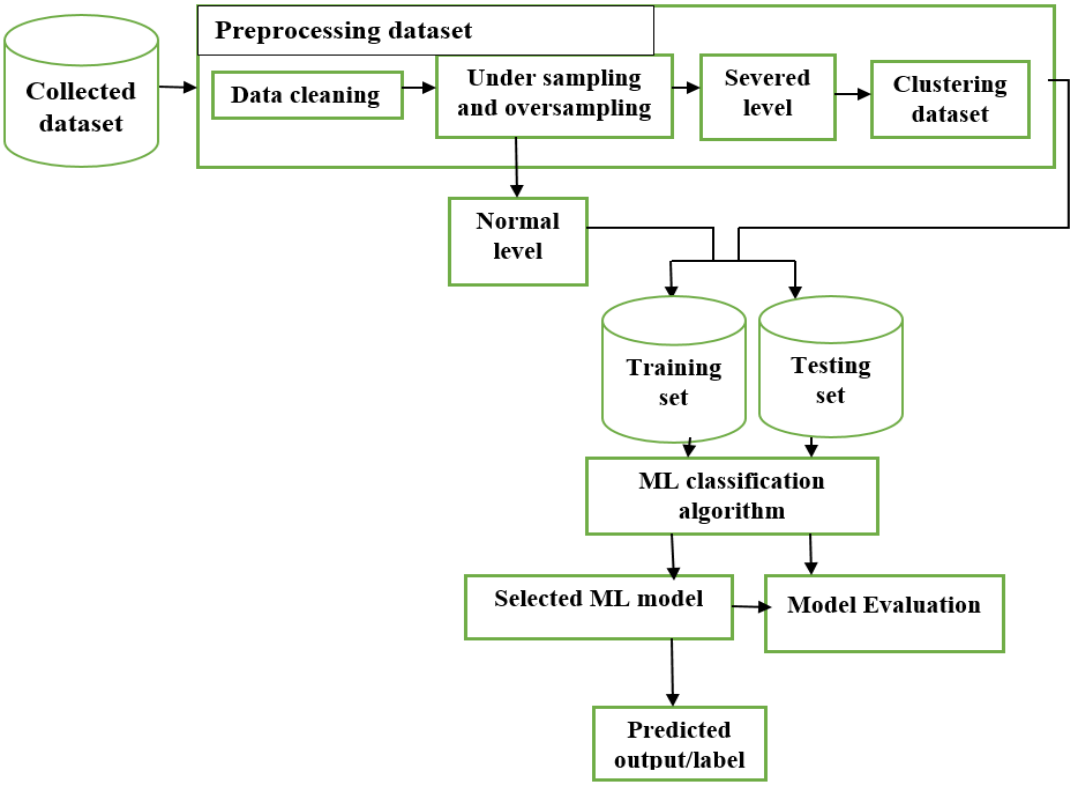


Figure 4. 1 BTS fuel level severity prediction framework

4.2 Proposed BTS Fuel Severity Prediction Preprocessing

The recommended BTS fuel level Severity prediction preprocessing component cleans the dataset by removing unnecessary records and handling missing values, after which sampling techniques are employed to compensate for the dataset's imbalanced character. The dataset is severed and non-severed classes are balanced using a combination of under- and oversampling approaches. Following that, the severed classes are clustered using the k-means clustering technique and grouped to appropriate class labels for prediction using the k-means clustering algorithm. The standard class datasets are then blended with the clustered severed classes.

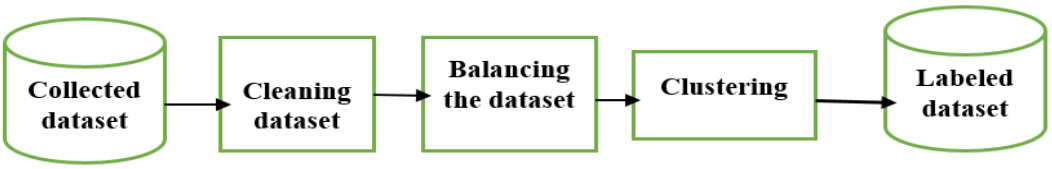


Figure 4.2 Dataset Preprocessing

4.2.1 Dataset Cleaning

Cleaning data involves deleting missing values or filling in blank spaces. Smooth noisy data is data that has no meaning and thus does not provide information. Outlier detection is a technique for removing odd observation data from large datasets. Inconsistencies and improper data are resolved by data, which reduces the accuracy of the data.

Input: Records in the dataset

As a result, the dataset has been cleaned.

1. Begin by reading the dataset's records.

2. If the data contains a Nan (not a number) value, drop the record with Nan (! end of records in dataset). If there are any null values in the data, Remove the record that has a null value.

3. Provide a clean dataset

4.2.2. Method for Dataset Balancing

To balance the severed and Normal classes of the dataset, this study proposes a mix of under-sampling and over-sampling strategies. This approach decreases the class to balance with the severed '1' (majority class) by randomly removing Normal or non-severed'0' (majority class) from the data (Minority class). To balance the Normal classes, which are the majority class, choose at random severed classes (minority class) and duplicates. This technique is used to boost performance by eliminating bias toward both classes. The procedure, as illustrated in Figure 4.3, takes the number of severed and normal classes and randomly reduces the number of majority classes while increasing the number of minority classes (severed classes) to balance the classes.

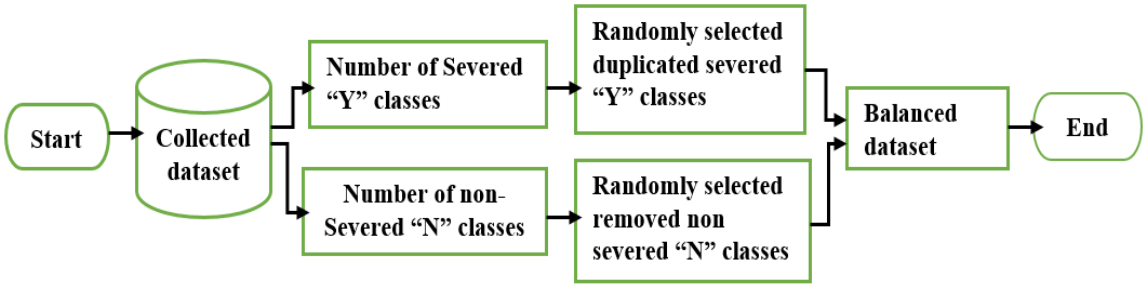
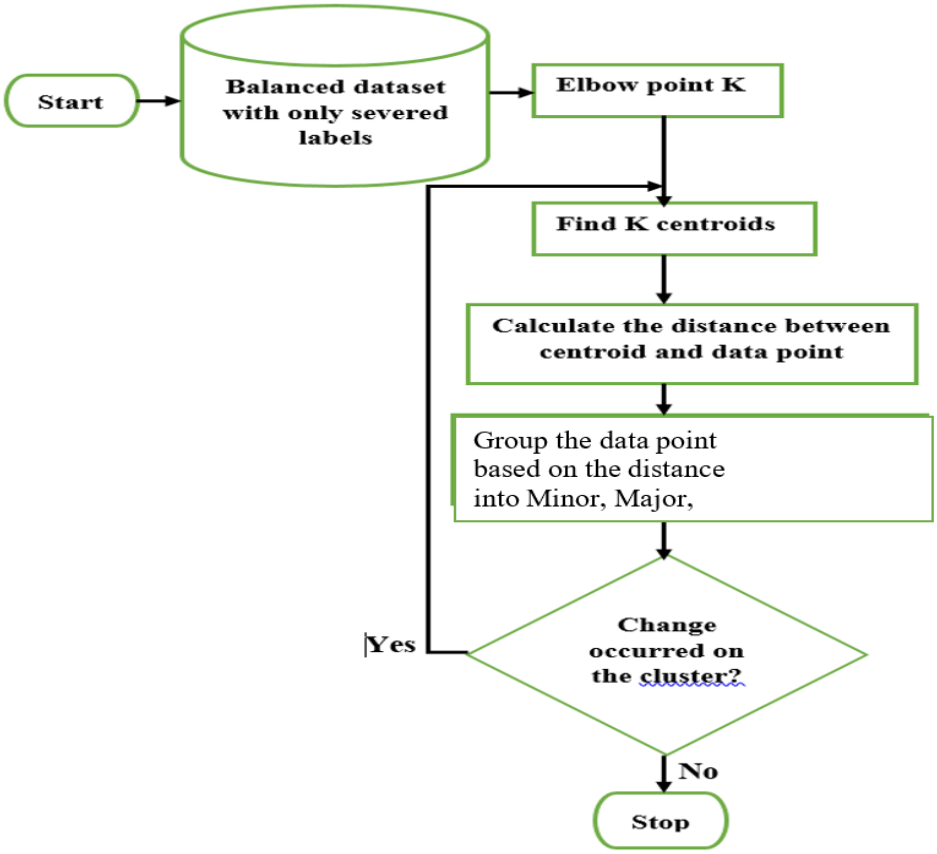


Figure 4.3 Flow diagram for balancing dataset

4.2.3 Applied Machine Learning Clustering Method

The k-means clustering approach was proposed in the study to acquire acceptable severity labels for prediction.The data is divided into k cluster groups using the algorithm.It categorizes items into clusters based on how similar they are.As indicated in the diagram below.After balancing the dataset, a dataset with only severed classes is grouped into severity categories in this study.The algorithm starts by counting how many k clusters there are.Elbow approaches are utilized to find a number of clusters in this study.Then choose the centroids and work out the distance between the instance and each centroid.The data points are categorized according to the computed range



.

Figure 4.3 Flow Diagram of the K-means clustering for the dataset

4.3. Developing Machine Learning Models

This part of the proposed framework for BTS fuel severity level prediction uses a machine learning classifier on the dataset's clustered fuel severity and non-fuel severity levels. In this study, machine learning methods such as DT, RF, SVM, and KNN are employed to create a classification model from which one will be chosen to predict BTS fuel severity levels.From the dataset, instances containing metrics (features) and labels are extracted.To evaluate the model, the dataset is separated into a training set and a test set.Using the learning algorithm, the modeling process detects patterns in the training set of the dataset and maps cases with their metrics (features) to the target class.Using the trained prediction model, new BTS fuel severity level metrics may be used to predict severity level.

The reaserch developed a decision tree classifier, which uses a branching structure to map the output of the decision tree classifier, which generates a model based on decision rules derived from data features to predict a value of the target variable.

To solve the multiclass problem, the research suggests a support vector machine classifier, the

One-vs-Rest approach of support vector machine. It used to be used to divide one class from another.

In the study, a random forest classifier was presented, which is a Meta predictor that fits number of decision tree classifiers on different sub-samples of the data set.Several decision trees are constructed and merged using the bagging method.The report provided a k-nearest neighbor classifier, which allocates a number of k-neighbors before computing their distance.

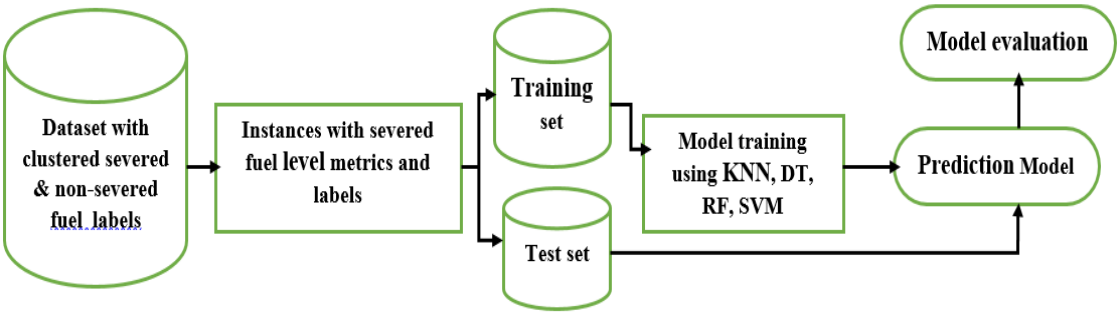


Figure 4.4 Diagram for Model Development

4.4. Models Evaluation

The evaluation of models is an essential element in identifying the best strategy for our data.It helps us assess how well our algorithm will perform in the future.For evaluating the learning ability of the model trained on the dataset for fuel severity level prediction, this study offered performance assessment metrics such as accuracy, precision, recall, F-measure, and confusion matrix, as mentioned in chapter three.

4.5. Implementation and Experimentation

Using the methods and recommended solutions outlined in the preceding section, implement BTS generator fuel level severity prediction.This research takes a unique experimental approach, utilizing the most advanced machine learning algorithms.We do five trials with five datasets and four machine-learning algorithms, comparing the results and recommending the best algorithm.

Table 4. 1 Implementation Environment

Tools		
Tools	Version	Description
Anaconda Navigator	2021.11	This command is used to start the development environment and packages.
Jupyter notebook	6.4.6	Allows us to develop and share data visualization, equations, statistical modeling, data cleansing, and machine learning-related documents.
Python		Is one of the most simple and intuitive programming languages for developing machine learning algorithms.
Microsoft Excel	2016	Used in the processing of data
Python package		
Scikit-Learn	1.0.2.	Sklearn is a set of Python modules for clustering, training, and testing models.
NumPy	1.20.3	A multi-dimensional array, matrix data structure is included in this numerical Python module.
Pandas	1.3.5	Allows us to import data from a variety of file formats, including CSV, merge data, and reshape it.
Matplotlib	3.4.3	It was applied in this research to visualize the results and data.
Seaborn	0.11.2	In this study, graphs were plotted in
Imblearn	0.7.0	The method for creating a data collection with an equal ration class, also known as a balanced class.

This study employed a variety of development tools and packages to implement BTS generator fuel level severity prediction.The python programming language is used to implement each proposed solution.Python is used due of its ease of use, consistency, and ability to access a variety of libraries.Python is used for everything from preprocessing to modeling to performance

evaluation. The tools and Python packages that were utilized are shown in the table above.

Table 4.1 Description of Tools and Python Packages Used in the Implementation.

4.6 Implementation Environment

The applications tools described in section 4.5 were installed on a personal computer with an Intel® CoreTM i7-6500UM processor, CPU 2.50GHz, 2 Core(s), 8 Gigabytes of physical memory, and a 1 terabyte hard disk storage capacity.Windows 10 Pro 64-bit is the operating system.

4.8 Description of the data set

BTS1, BTS2, BTS3, BTS4, and BTS5 are the five BTS data sets used in this research.This dataset was compiled from BTS that can be used to represent the rest of the BTS.The following table lists the attributes in the dataset, along with their descriptions, that were not included in the study.

Table 4. 2 Dataset attribute

NO	Attribute	Attribute
1	BTS ID	Base station ID for identification from others.
2	Elapsed hr.	Time taken when fuel is consumed.
3	Date	Date status of fuel occurred
4	BTS distance from center	BTS location from center of monitoring.
5	Height Fuel level in tank	Height of the fuel level in the tank which determines the amount of fuel level in terms of cm.
6	Fuel level in litter	Amount of fuel in the tank in terms of litter
7	Fuel level in percent	Amount of fuel in the tank in terms of percent
8	Class	Dependent variable severity and normal state of fuel level

4.9 Implementation of preprocessing

Filtering the dataset, dealing with missing values, resampling the dataset to balance the biased classes, and assigning the dataset to the right group of class labels for prediction have all been done.The Python computer language is used to implement BTS fuel severity level prediction in this study.To import library packages and load the dataset used in the implementation, we used the python code shown in figure 4.6..

```
import csv
import pandas as pd
d1 = pd.read_csv("BTS1.csv")
d2 = pd.read_csv('BTS2.csv')
d3 = pd.read_csv('BTS3.csv')
d4 = pd.read_csv('BTS4.csv')
d5 = pd.read_csv('BTS5.csv')

```

Figure 4.5 The datasets are loaded using Python code.

This method scales the values for each numerical characteristic in the dataset using the StandardScaler () method, which is also known as standardization.

```
from sklearn.preprocessing import StandardScaler
x=StandardScaler().fit_transform(d)
```

Figure 4.6 Python code to standardize datasets.

4.10 Deployment for cleaning the data set

Various python methods are used to examine and clean the dataset.is Null ().values are used to display the rows count and data kinds, describe () methods are used to present the primary statistics for each numerical column in our dataset, and info () methods are used to show the rows count and data types.To check if the dataset contains Nan values and missing values, any () methods are employed.The all () method is used to see if the dataset and the table have the same values.Those methods return True if there is a value or False if there is none. The drop() function is used to clean data by removing true values with missing or nan values.

4.11 Dataset Balancing Implementation

The study employed the imblearn python package, which provides access to resampling techniques, to balance the classes of severed and non-severed classes of selected datasets using the suggested method.The imblearn package's RandomUnderSampler sub-module is used to implement under-sampling techniques, whereas RandomOverSampler is used to develop over-sampling techniques. The RandomUnderSampler () class selects samples from the dataset atrandom in order to under-sample the majority class, which is not faulty.The majority class

employs a sampling strategy to specify the ratio of under-sampling approaches; in this study, we utilized 0.8 ratios to under-sample the dataset, which implies the majority class is under-sampled by 80% as the minority class. By randomly duplicating the minority class, RandomOverSampler () over-samples the minority.To oversample the dataset, we utilized 0.6 ratios to the majority class.This signifies that the minority class is 60 percent overrepresented in the majority class.To fit and apply the changed data to the dataset, the fit resample () function is used.

```
import imblearn
from imblearn.under_sampling import RandomUnderSampler
from imblearn.over_sampling import RandomOverSampler
from collections import Counter
```

```
]:  1 over =  RandomOverSampler(sampling_strategy=0.6)
    2 under = RandomUnderSampler(sampling_strategy=0.8)
    3 X_over,y_over=over.fit_resamplle(X,y)
    4 X_combine_sampling,y_combined_sampling = under.fit_resample(x_over,y_over)
```

Figure 4.7 Implementation code for under sampling and oversampling

4.12 k-means clustering execution

The proposed approach was used to cluster the data into severity categories, We initiated by importing the crucial library package for modeling using a python ski-kit learning library package.

```
import csv
import panda as pd
from pandas import DataFrame
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.cluster import Kmeans
import sklearn.cluster as cluster
import numpy as n
```

Figure 4.8 clustering by importing necessary packages

Only severed classes are clustered to severity levels after the dataset has been balanced. Only severed fuel level classes are required for severity level assessment.

```
# Only severed fuel levele data is being loaded.YES
d1 = pd.read_csv("BTS1YES.csv")
d2 = pd.read_csv("BTS2YES.csv")
d3 = pd.read_csv("BTS3YES.csv")
d4 = pd.read_csv("BTS4YES.csv")
d5 = pd.read_csv("BTS5YES.csv")
```

Figure 4.9 Python code to load a dataset with only severed class

In this work, the elbow method was used to select an appropriate number of clusters; the method is implemented using the KMeans class from the scikit learn sklearn.cluster package library; the class receives parameter numbers of clusters to locate by iterating across the range.Kmeans.fit () was used to determine the cluster in scaled data.The number of clusters is then chosen, and WCSS (Within Cluster Sum of Squares), a measure of the variability between data points within each cluster, begins to lay off, but does not diminish with each iteration.

```
from sklearn.cluster import Kmeans
wcss = []
for i in range(1, 11):
    kmeans = kmeans(n_cluster = i, int = 'k-means++',max_iter = 500,n_init=10,random_state=0)
    kmeans.fit(y)
    wcss.append(kmeans.inertia_)
#Plotting the results onto a line graph, allowing us to observe the elbow.
plt.style.use("fivethirtyeight")
plt.plot(range(1,11)wcss)
plt.title('The elbow method')
plt.xlable('Number of clusters')
plt.ylabel('wcss')#within cluster sum squares
plt.show()
```

Figure 4.10 The elbow method is implemented in Python code.

The number of clusters acquired by the elbow approach is used to implement K-means clustering, which yields three clusters. The KMeans class from the sklearn.cluster package library of scikit learn is used to build the Kmeans clustering algorithm. For each data, the Kmeans.fit predict () method computes cluster centers and predicts cluster groups. Kmeans. Cluster center displays the centroids' most recent location. Kmeans. Labels returns a clustered set of labels.

```
kmeans=clustering.Kmeans(n_clusters=3,int="k-means++",max_iter=500,n_init=10,random_state=0)
y_kmeans=kmeans.fit_predicict(x)
kmeans.cluster_centers_
data['Severity'] =kmeans.labels_
```

Figure 4. 11 Python code for K-means clustering

4.13 Implementation of a Machine Learning Model

We used a python ski-kit learning library package to develop machine-learning models using the suggested technique, followed by a mechanism for importing the key library package for modeling and metrics for model evaluation.

```
import csv
import pandas as pd
from panadas import DataFrame
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from  sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from sklearn.svm import LinearSVC
from sklearn.multiclass import OneVsRestClassifier
from sklearn.multiclass import KNeighborsClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.datasets import make_multilabel_classification
```

Figure 4. 12 Importing the necessary software package

The new-grouped datasets are utilized to estimate the severity degree of fuel level once the dataset is clustered into severity categories.The dataset with the severity label is loaded, and we split it into 80 percent training and 20 percent test sets using the train test split() method, X _train, y_train to train using the fit()with the appropriate set parameter for each classifier instantiate the class.To test the model, use x _test and y _test.

```
d1 = pd.read_csv("BTS1cluster.csv")
d2 = pd.read_csv("BTS2cluster.csv")
d3 = pd.read_csv("BTS3cluster.csv")
d4 = pd.read_csv("BTS4cluster.csv")
d5 = pd.read_csv("BTS5cluster.csv")
```

```
x = data.drop('Severity',axis=1)
y = data['Secerity']
X_train,x_test,Y_train,y_test_split(X,y,test_size=0.2,random_state-0)
```

Figure 4.13 Splitting a dataset into a training and testing set using Python

RandomForestClassifier () from the sklearn ensemble package is used to train the classifier for the RF classifier.GridSearchCV () from sklearn model selection is used to produce candidates from a grid of parameter values supplied with the parameter, while the best combination of parameters is evaluated and retained. As specified in the n estimator's parameters, this classifier fits many decision tree classifiers.

```
from sklearn.ensemble import RandomForestClassier
from sklearn.model_selection import GridSearchCV
param = {"n_estimators":[100,120,150,1000],
        "max_depth":[3,5,10]
        "min_samples_split":[2,5,10]}
#instatnciate random forest
clf=RandomForestClassifier()
forest=GridSearchCV(clf,param)
#Fit model to train the model
forest.fit(X_train,y_train)
```

Figure 4.14 Fit the model and build RF with Python code

The DecisionTreeClassifier () from the sklearn tree package is used to train the classifier for the DT classifier.GridSearchCV() from the sklearn model selection library was used to generate candidates from a grid of parameter values supplied with the param argument, and then build a decision tree classifier using the training data.

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import GridSearchCV
param = {"max_depth":range(1,10),
        "min_samples_leaf":range(1,9),
        "criterion":["gini,"entropy"]}
#initializing tree classifier
clf=DecisionTreeClassifier()
DT = GridSearchCV(clf,param)
#fit the model to predict
DT.fit(X_train,Y_train)
```

Figure 4.15 Fit the model and build DT with Python code

To create the SVM model, the researchers employed the sklearn package's Linear SVC() classifier.For multi-class classification, this classifier uses the one-vs-rest parameter.

```
from sklearn.svm import LinearSVC
#inistanciate the training model
svclassifier = LinearSVC(multi_class='ovr',loss='squared_hing',dual=True
                         max_iter=10000,class_weight='balanced',c=0.01,
                         verbose=0)
#fit the model to train
svclassifier.fit(X_train,y_train)
```

Figure 4.16 Fit the model and build SVM with Python code

To train the classifier for the KNN model, we used KNeighborsClassifier () from the sklearn neighbors package.GridSearchCV() is a sklearn model selection function that generates candidates from a grid of parameter values supplied with the param argument. It implements learning for each data point based on the grid search method's nearest neighbors.

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import GridserachCV
param={"leaf_size":range(3,10),
       "n_neighbors":range(1,10),
       "p":[1,2]
       }
#instanciate the model
clf = KNeighborsClassifier()
clfk=GridSearchCV(CLF,param)
#fit to train the model
clfk.fit(x_train,y_train)
```

Figure 4.17 Fit the model and build KNN with Python code

4.14 Model Assessment and Testing

A 20% test dataset is used to assess the performance of the training model.To evaluate the performance of the created models using predict the value of the test set, the study employed the classification report (), accuracy score (), and confusion matrix () techniques from the Sklearn metrics package.These approaches calculate the accuracy, precision, recall, and F1-score value of the model under test.

CHAPETR FIVE RESULT And DISCUSSION

5.1 Introduction

The results of the proposed BTS fuel severity level prediction experiment utilizing a machine-learning system are covered in this section.For each dataset, the outcomes of dataset balancing approaches, clustering, and classification models are discussed.The results of each experiment in this study are discussed in the concluding section.

5.2 Result of dataset balancing

The study used a mix of under-sampling and over-sampling strategies to balance the dataset's selected class. Five datasets are used to test the methods: BTS1, BT21, BTS3, BTS4, and BTS5.The actual and balanced dataset is shown in the following Figures, which were created using a combination of oversampling and undersampling approaches. As shown in Figures one is the severed class and zero is the non-severed class of the dataset. Figure 5.1 shows the BTS1 actual dataset with 4884 records, from which 3568 records are non-severed and 1316 records are severed. After balancing the dataset as shown in Figure 5.2 the severed class records are 3568 class and non-severed class are also too.

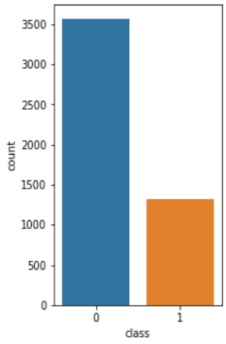


Figure 5. 1 Initial BTS1 dataset

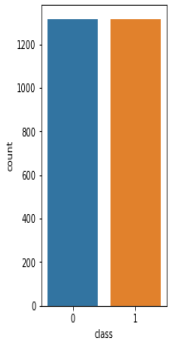


Figure 5. 2 Bts1 Data after balancing

Figure 5.3 indicates the BTS2 intial dataset with 5208 records, from which 3848 records are non-severed and 1360 records are severed. The severed class is 3848 data, and the non-severed class is 3848 data, for a total of 7696 data after balancing the dataset as illustrated in Figure 5.4.

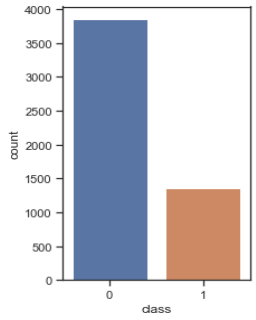


Figure 5. 3 Initial BTS2 dataset

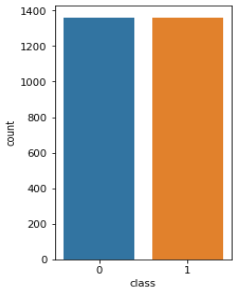


Figure 5. 4 Bts2 Data after balancing

Figure 5.5 indicates the BTS3 intial dataset with 5208 records, from which 3374 records are non-severed and 1834 records are severed. The severed class is 3374 data, and the non-severed class is 3374 data, for a total of 6748 data after balancing the dataset as illustrated in Figure 5.6.

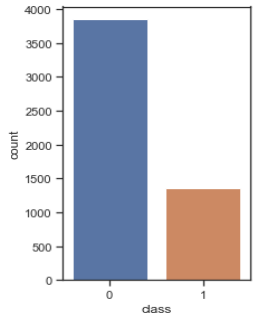


Figure 5. 5 Initial BTS3 dataset

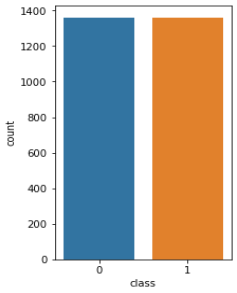


Figure 5. 6 Figure 5. 6 Bts3 Data after balancing

Figure 5.7 indicates the BTS4 intial dataset with 5112 records, from which 3743 records are non-severed and 1369 records are severed. After balancing, the dataset in Figure 5.8 shows that the non-severed class has 3743 data and the severed class has 3743 data, for a total of 7486 datasets.

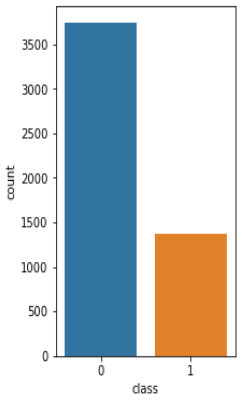


Figure 5. 7 Initial BTS4 dataset

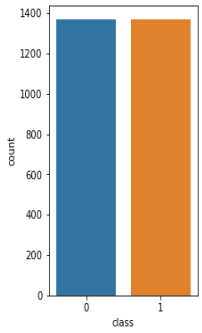


Figure 5. 8 Bts4 Data after balancing

Figure 5.9 indicates the BTS5 intial dataset with 5112 records, from which 3161 records are non-severed and 1951 records are severed. After balancing the dataset, as shown in Figure 5.10, the severed class consists of 3161 data, while the non-severed class consists of 3161 data, for a total of 6322 data.

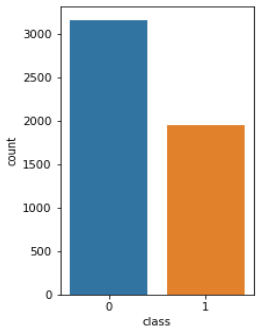


Figure 5. 9 Initial BTS5 dataset

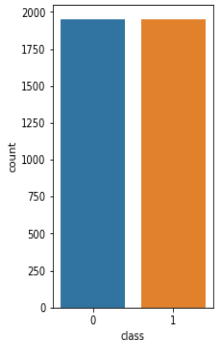


Figure 5. 10 class distribution after balancing BTS5 dataset

5.3 K-means clustering result

Each BTS dataset's severed classifications are classified into critical, major, and minor cluster groups, which are useful for predicting fuel level severity.Each dataset's clustering result is shown as follows.Using the elbow approach, the suitable number of clusters is chosen; the number is chosen where the elbow point began to lay off.In Figure 5.11, the elbow point for BTS1 is displayed, with three clusters where the elbow begins to lay down.

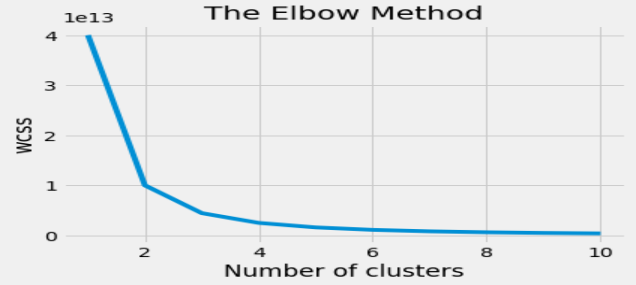


Figure 5.11 BTS5's elbow point

BTS5 cluster groupings are shown in Table 5.1.The 1951 severed classes in the BTSYES dataset are clustered into three groups, yielding 402 items on cluster 0, 802 items on cluster 1, and 743 data on cluster 2.

Table 5. 1 Cluster group for BTS5

Domain group	Items	Levelof severity
Cluster- 0	402	Critical
Cluster -1	806	Minor
Cluster -2	743	Major

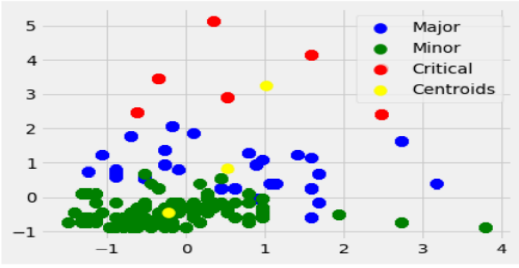


Figure 5.12 BTS5YES cluster group.

The elbow point for BTS1YES is shown in Figure 5.13. The number of clusters where the point started lying off is three in the elbow result.

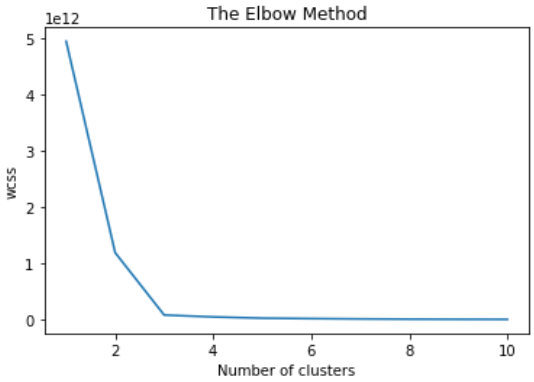


Figure 5.13 BTS1's elbow point

BTS1 cluster groups are shown in Table 5.2.1316 disconnected classes from the BTS1 dataset are clustered into three groups, yielding 210 items on cluster 0, 640 items on cluster 1, 466 data on cluster 2, and 210 items on cluster 0, 640 items on cluster 1.

Table 5.2 Cluster group for BTS1

Domain group	Items	Levelofseverity
Cluster 0	210	Critical
Cluster 1	640	Major
Cluster 2	466	Minor

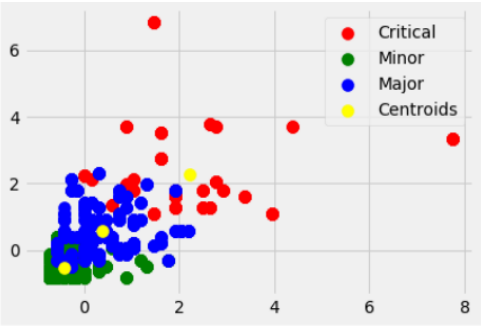


Figure 5.14 BTS1YES Cluster group

The elbow point for BTS2 is shown in Figure 5.15.The number of clusters where the point started lying off is three in the elbow result..

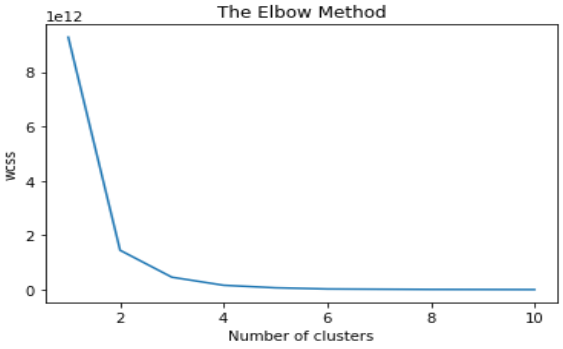


Figure 5.15 BTS2's elbow point

BTS2 cluster groups are shown in Table 5.3.The 3848 severed classes in the BTS2 dataset are clustered into three groups, yielding 840 items in cluster 0, 1653 items in cluster 1, and 1355 data in cluster 2.

Table 5.3 Cluster group for BT2

Domain group	Items	Level of severity
Cluster -0	840	Critical
Cluster -1	1653	Major
Cluster -2	1355	Minor

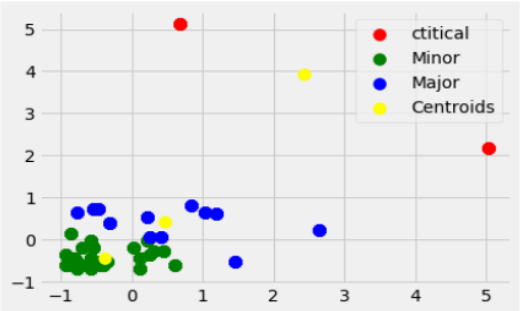


Figure 5.16 Cluster group for BTS2

The elbow point for BTS3 is shown in Figure 5.16.The number of clusters where the point started lying off is three in the elbow result.

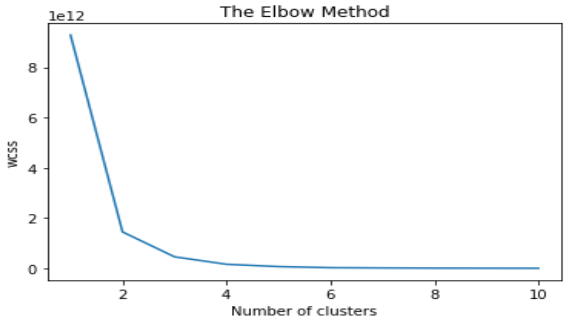


Figure 5.17 BTS3 BTS3's elbow point

Table 5.4 shows cluster groups of BTS3, From the BTS3 dataset 3568 severed classes are clustered into 3 groups,which result in 1213 items on cluster 0,1460 items on cluster 1,895 data on cluster 2.

Table 5. 4 BT3 Cluster group

Domain group	Items	Level of severity
Cluster 0	1213	Critical
Cluster 1	1460	Major
Cluster 2	895	Minor

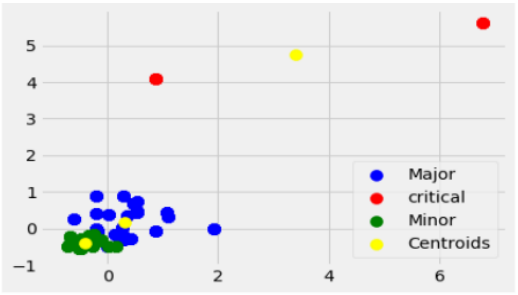


Figure 5. 18 BTS3 Cluster-group

Figure 5.16 displays the elbow point for BTS4. The elbow result indicates three as the number of clusters where the point began lying off.

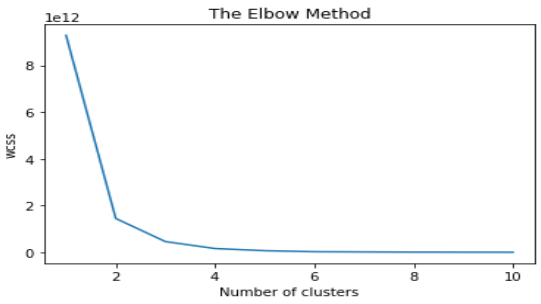


Figure 5.19 Elbow point for BTS4

Table 5.5 shows cluster groups of BTS4, from the BTS4 dataset 3743 severed classes are clustered into three groups, which result in 948 items on cluster 0, 1236 items on cluster 1, 1559 data on cluster 2.

Table 5.5 Cluster group for BTS 4

domain group	items	level of severity
cluster- 0	1213	Critical
cluster -1	1236	Major
cluster -2	1559	Minor

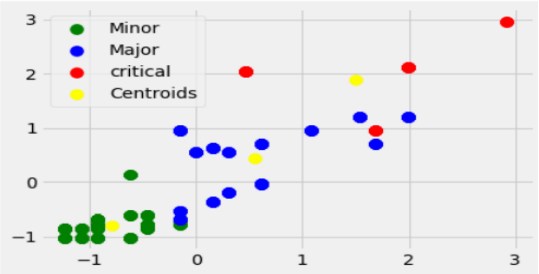


Figure 5.20 BTS 4 cluster-group

5.4 Model Evaluation Result

We used the clustered datasets from BTS1, BTS2, BTS3, BTS4, and BTS5, which result in four models for each dataset. The performance of the trained models is tested using 20% test set. The result of each trained model testing accuracy, precision, recall, and F1-score are presented for, DT, RF, KNN, and SVM respectively are shown in the tables below. The result in Table 5.6 shows the performance score of RF model on each dataset. The higher accuracy recorded is between 92.1 & 97.4percent and the F1 score is between 92-97percent.

Table 5.6 RF Performance result.

Algorithm	Dataset	Accuracy	Precision	Recall	F1-score
RF	BTS 1	96.5	98	97	97
	BTS 2	92.1	92	92	92
	BTS 3	95.2	96	95	95
	BTS 4	97.4	98	97	97
	BTS 5	95.8	96	96	96

The performance score of the DT model on each dataset is indicated Table 5.7.The better accuracy gained ranges from 84.8 to 94.3 percent, and the F1 score ranges from 85 to 94%.

Algorithm	Dataset	Accuracy	Precision	Recall	F1-score
DT	BTS 1	94.3	95	94	94
	BTS 2	84.8	85	85	85
	BTS 3	91.8	94	92	92
	BTS 4	93.7	93	91	91
	BTS 5	90.3	92	90	90

The KNN model's performance result on each dataset is shown in Table 5.8.The better accuracy gained ranges from 89.3 to 77.1 percent, while the F1 score ranges from 77 to 89 percent.

Table 5.7 KNN Performance result

Algorithm	Dataset	Accuracy	Precision	Recall	F1-score
KNN	BTS 1	86.1	90	86	86
	BTS 2	84.8	85	85	85
	BTS 3	77.1	84	77	77
	BTS 4	89.3	91	89	89
	BTS 5	86.2	89	86	86

The performance result of the SVM model on each data set is seen in Table 5.9.The greater accuracy gained ranges from 56.2 to 73.4 percent, while the F1 score ranges from 56 to 74%.

Table 5. 8 SVM Performance result

Algorithm	Dataset	Accuracy	Precision	Recall	F1-score
SVM	BTS 1	63.5	71	64	66
	BTS 2	58.1	66	58	59
	BTS 3	56.2	57	56	56
	BTS 4	73.4	77	73	74
	BTS 5	64.8	62	65	63

Figure 5.21 shows a visual depiction of the accuracy of each model based on the datasets in the previous four tables Blue represents the classification accuracy of DT models, brown represents the classification accuracy of RF models, silver represents the classification accuracy of KNN models, and yellow represents the classification accuracy of SVM models.On the x-axis, the selected datasets are presented, and the accuracy result is based on the accuracy result, which shows that the RF models are more accurate on each dataset than the DT, KNN, and SVM models.

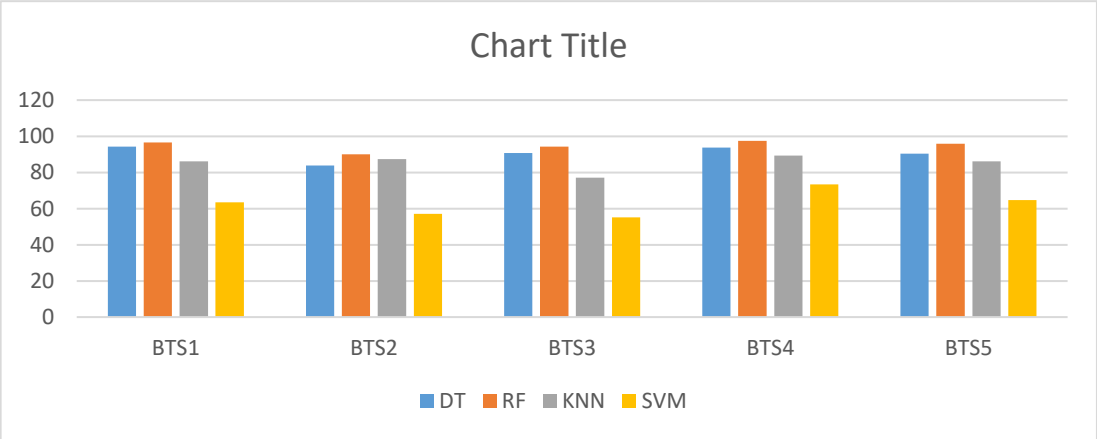


Figure 5.21 Models compared depending on their accuracy.

The F-measure is used to compare models. To compare the four classifiers' recall and precision, we employed F-measure values. The F1 measure values for the used algorithms in the five datasets are shown in Figure 5.22.The selected datasets are depicted on the x-axis, and the F-measure result is represented on the y-axis. In all datasets, the RF model has the highest f1 score, followed by DT, KNN, and finally the SVM classifier, as shown in the picture.

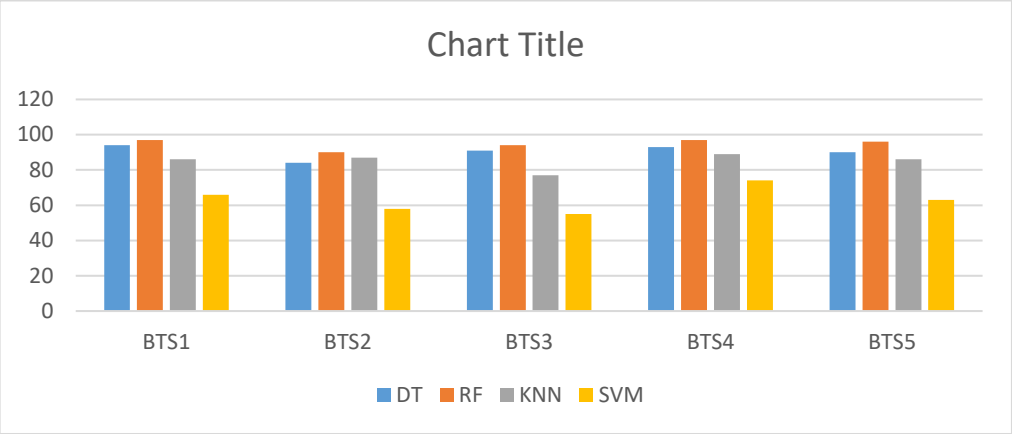
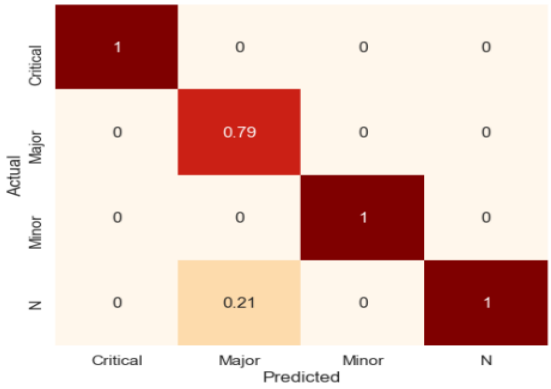
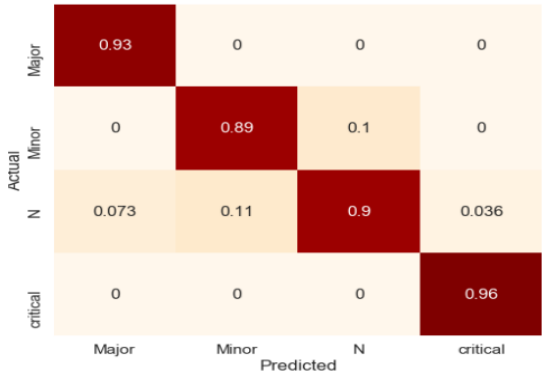


Figure 5. 22 Models compared depending on their f1-measure.

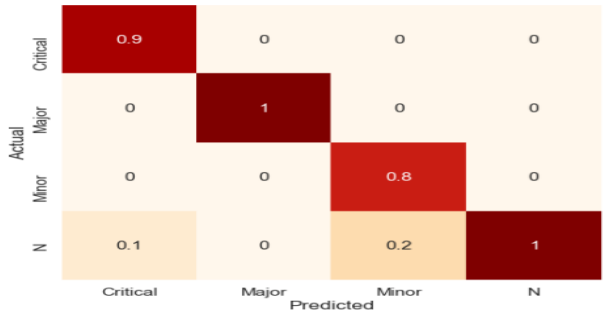
BTS1



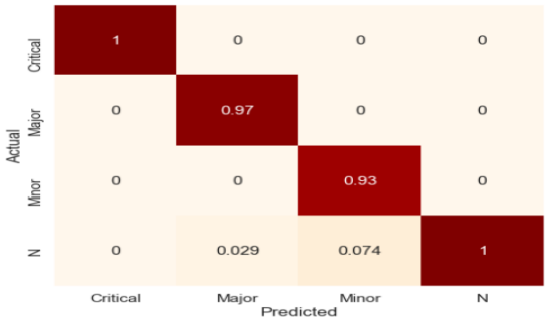
BTS2



BTS3



BTS4



BTS5

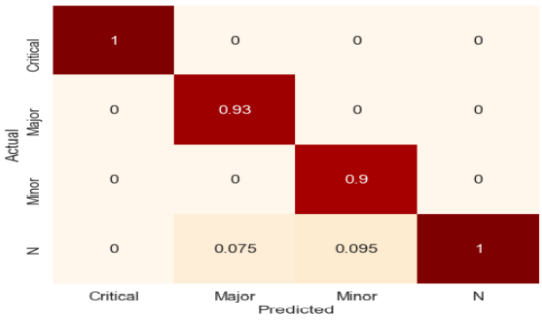
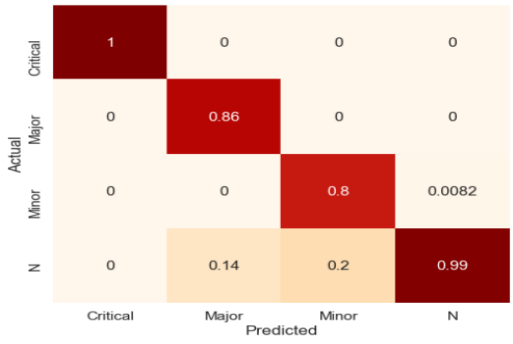


Figure 5.24 a Confusion Matrix for RF models

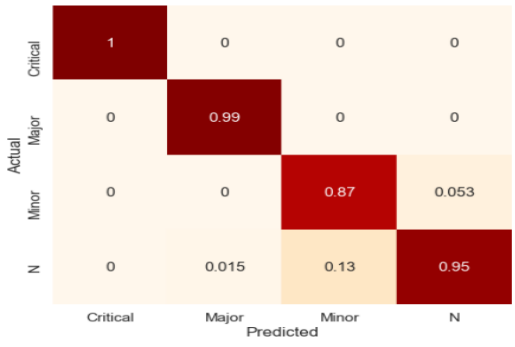
The confusion matrix for RF models is shown in Figure 5.23.On BTS1, RF accurately identifies 100percent of Critical, 79 percent of Major, 100 percent of Minor, and 100 percent of '0' classes; there is a 21percent misclassification between Major and N. On BTS2, 93 percent of major, 89 percent of Minor, 90 percent of '1', and 96 percent of crucial classes were correctly classified, but 7.3 percent of Major classes were misclassified as '0' classes, and 11 percent of Minor and '0' classes were misclassified. On BTS3, 90perecent of critical, 100 percent of major, 80 percent of Minor, and 100percent of critical were correctly identified, but 20perecn of Minor was misclassified as '0', resulting in a 10% misclassification between critical and '0' class. On BTS4, 100percent of Critical, 97 percent of Major, 93 percent of Minor, and 100% of '0' classes are correctly classified, however 3% of Major and 7.5 percent of Minor classes are misclassified as '0'. On BTS5, 100percent of Critical, 93 percent Major, 90percent Minor, and 100% of '0' class were correctly identified; however 9.5 percent of Minor and 7.5 percent of Major were misclassified as '0'.

In the picture below, the normalized confusion matrix result for the DT model for each dataset is presented.

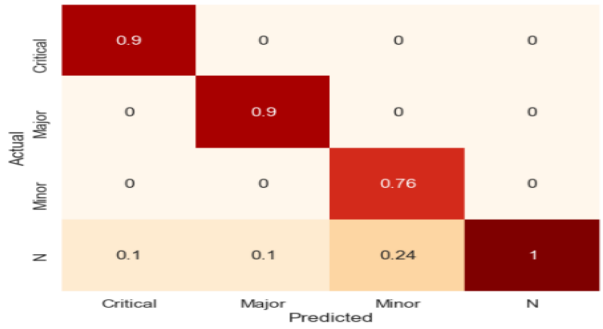
BTS5



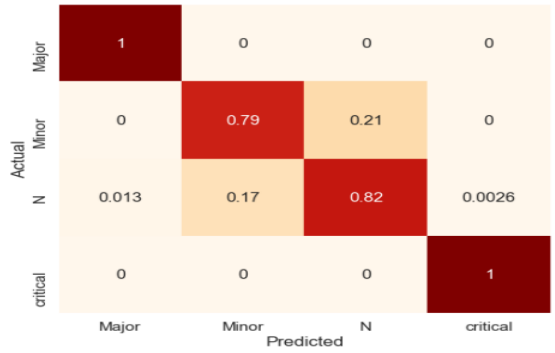
BTS4



BTS3



BTS2



BTS1

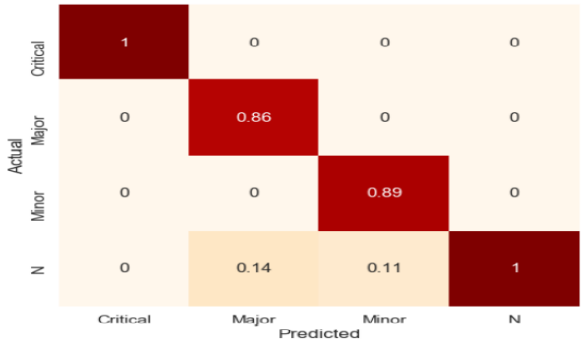


Figure 5.25 Figure 5: Confusion Matrix for Decision Tree Model

The confusion matrix for DT models is shown in Figure 5.24.On BTS5, 100 percent of Critical, 86 percent Major, 80 percent Minor, and 99 percent ‘0’ classes were correctly identified, while 14 percent of Major was misclassified as ‘0’, 20 percent of Minor was misclassified as ‘0’, and 0.82 percent of ‘0’ was misclassified as Minor.BTS4 accurately classifies 100 percent of Critical, 99 percent of Major, 87 percent of Minor, and 95 percent of ‘0’ classes. However, 1.5 percent of Major classes are misclassified as ‘0’, and 13percent of Minor classes are misclassified as ‘0’. On BTS3, 90percent of Critical, 90percent of Major, 76percent of Minor, and 100percent of ‘0’ are correctly classified, but 10percent of Critical, 10percent of Major, and 24percent of Minor are misclassified as‘0’. On BTS2, 100percent of major, 79 percent of Minor, 82 percent of‘0’, and 100percent of Critical classes are correctly classified, however 17percent of Minor classes are misclassified as ‘0’, and 21percent of ‘0’ classes are misclassified as Minor. On BTS1, 100 percent of Critical, 86 percent Major, 89 percent Minor, and 100 percent of ‘0’ classes are correctly classified, however 11 percent of Minor and 14 percent of Minor are misclassified as ‘0’.

For each dataset, the normalized confusion matrix result for the KNN model is presented in the figure below.

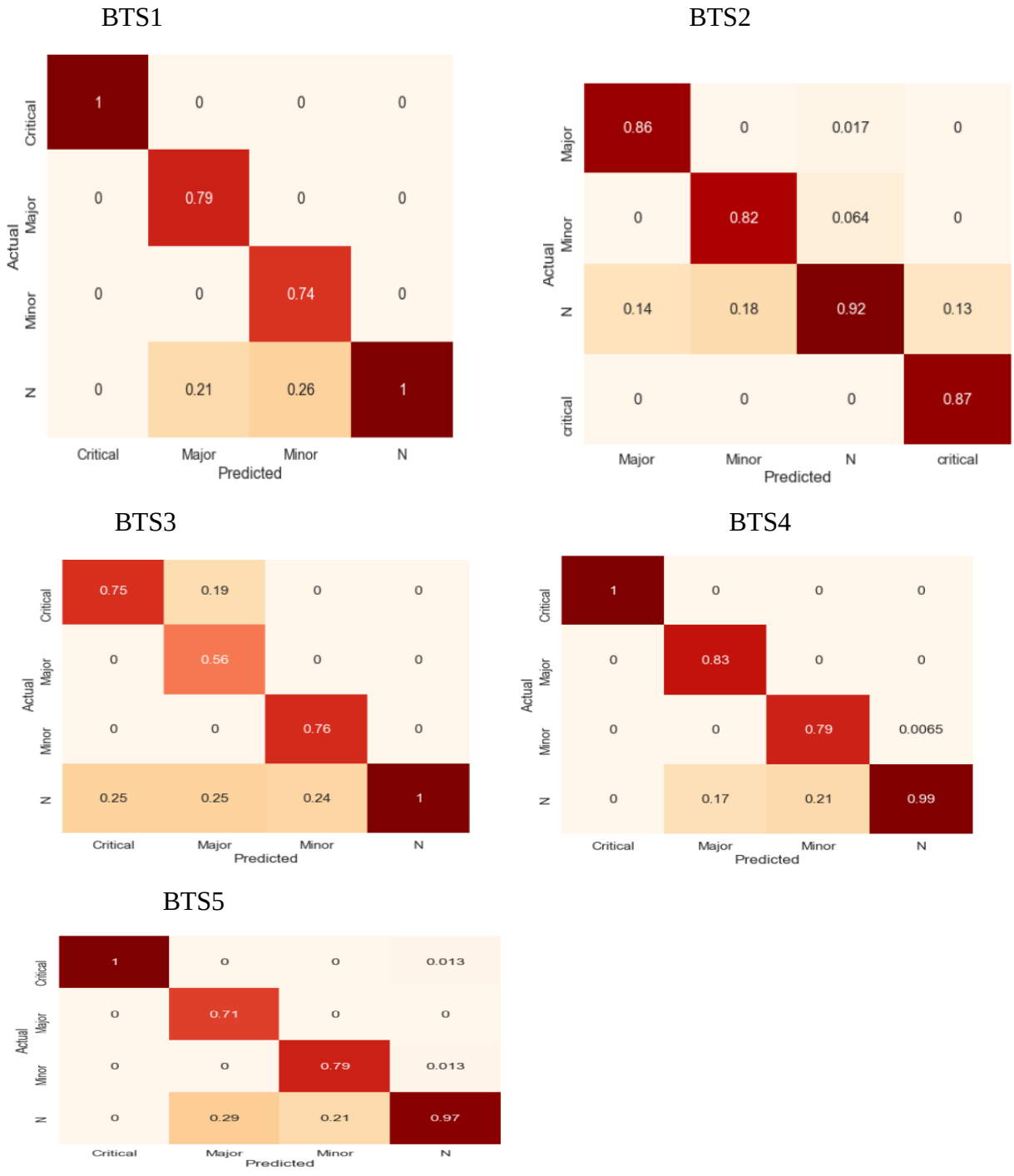


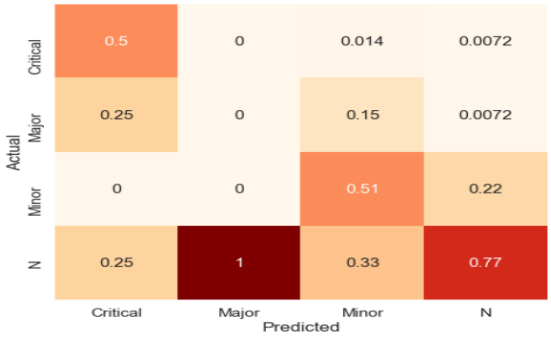
Figure 5.26 For each dataset, a Confusion Matrix for K-nearest neighbor models

The confusion matrix for KNN models is shown in Figure 5.25.On BTS1, 100 percent of Critical, 79 percent of Major, 74 percent of Minor, and 100 percent of ‘0’ classes are correctly classified, however 21percent of Major and 26percent of Minor classes are incorrectly classified as ‘0’. On BTS2, 86 percent of Major, 82 percent of Minor, 92 percent of ‘0’, and 87 percent of Critical classes are correctly classified, but 14 percent of Major classes are misclassified as‘0’, 18 percent

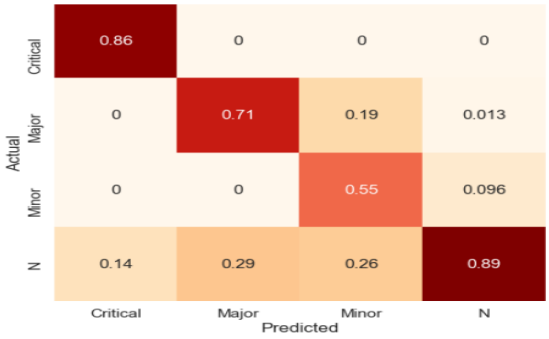
of Minor classes are misclassified as ‘0’, 6.4 percent of ‘0’ classes are misclassified as Minor, 1.7 percent of ‘0’ classes are misclassified as Major, 13 percent of critical classes are misclassified as ‘0’, and 6.1 percent of critical classes are misclassified as Major. On BTS3, 75percent of Critical, 56percent of Major, 76percent of Minor, and 100percent of ‘0’ are correctly classified, however 25percent of Critical classes are misclassified as ‘0’, 19percent of Major classes are misclassified as Critical, 25percent of Major classes are misclassified as ‘0’, and 24percent of Minor classes are misclassified as ‘0’. On BTS4, 100 percent of crucial, 83 percent of Major, 79 percent of Minor, and 99 percent of ‘0’ are correctly classified, however 17percent of Major class and 21percent of Minor class are misclassified as ‘0’. On BTS5100, 100percent of critical, 71 percent of major, 79 percent of Minor, and 97 percent of ‘0’ classes are correctly classified, however 29percent of Major and 21percent of Minor classes are misclassified as‘0’, and 1.3 percent of N classes are misclassified as Critical and Minor.

The result of the SVM Model's normalized confusion matrix for each dataset is presented in the image below.

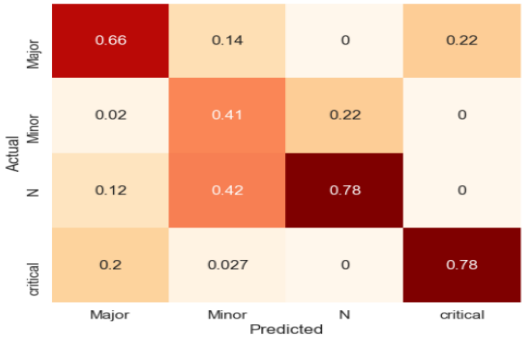
BTS5



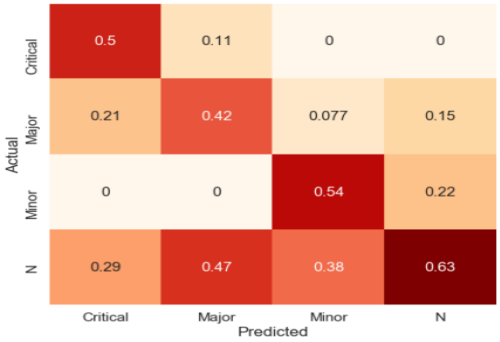
BTS4



BTS2



BTS3



BTS1

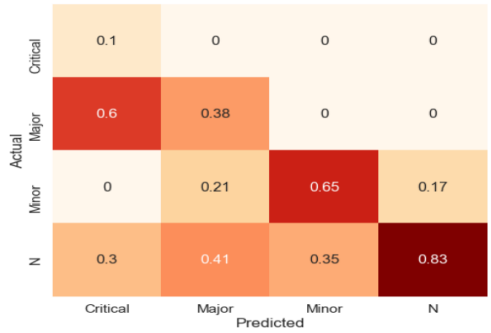


Figure 5.27 Confusion Matrix for SVM models for each selected datasets

The confusion matrix for SVM models is shown in Figure 6.19.On BTS5, 50percent of Critical, of Major, 51percent of Minor, and 77percent of N are correctly classified, but 25percent of Critical class is misclassified as ‘0’ and 25percent of Critical class is misclassified as Major, 100percent of Major classes are misclassified as ‘0’, 33percent of Minor classes are misclassified as ‘0’ class, 15percent of Minor class is misclassified as Major, 0.14 percent of Minor class is misclassified as Critical class, and 22percent of Critical class is misclassified as Critical class. On BTS4, 86percent of Critical, 71 percent Major, 55percent Minor, and 89 percent‘0’ classes are correctly classified, but 14percent of Critical classes are misclassified as‘0’, 29percent of Major classes are misclassified as ‘0’, 19percent of Minor classes are misclassified as Major, 26percent of Minor classes are misclassified as ‘0’, and 9.6percent of ‘0’ classes are misclassified as Minor. On BTS3, 66 percent of Major, 41 percent of Minor, 78 percent of ‘0’, and 78 percent of Critical classes are correctly classified, but 2percent of Major classes are misclassified as Minor, 12percent of Major as ‘0’, 20percent of Major as Critical, 14 percent of Minor classes are misclassified as Major, 42 percent of Minor class misclassified as ‘0’, 22 percent of ‘0’ class misclassified as Minor, and 22 percent of Critical class misclassified as Critical. On BTS2, 50percent of Critical, 42 percent of Major, 54 percent of Minor, and 63 percent of ‘0’ classes are correctly classified, but 29percent of Critical class misclassified as ‘0’, 21percent of Critical class misclassified as Major, 47percent of Major class misclassified as ‘0’, 11percent of Major class misclassified as Critical, 7.7percent of Minor class misclassified as Major, 38percent of Minor class 72 misclassified as N, and 22perecnt of ‘0’class misclassified as Critical. On BTS1, 10percent of Critical class, 38percent of major, 65percent of Minor, and 90percent of ‘0’ class were correctly classified, but 60percent of Critical class, 30percent of Critical class, 30percent of Critical class, 40percent of Major class, 41percent

of Major class, 21percent of Major class, 35percent of Minor class, and 17percent of '0' class were incorrectly classified. The normalized confusion matrix in Figures 6.16, 6.17, 6.18, 6.19 respectively shows the classification result of each model. The actual class and the predicted class are the prediction labels by the models, which are Critical, Major, Minor, and '0'. For each dataset RF, model result shows better performance than DT, SVM, and KNN models.

5.5 Discussion

Fuel severity level prediction for BTS was the subject of this paper. Using supervised learning, we were able to predict the severity degree of the BTS generator fuel tank.BTS fuel level severity attributes with the classes severed and non-severed were included in the dataset for this analysis. Although the goal of the study is to estimate fuel level severity, we found that the number of non-severed classes in the dataset is far more than the severed class.

To address this issue, we used under-sampling and over-sampling methods, with a 60 percent ratio of oversampling and an 80 percent ratio of under-sampling to balance the classes. We reduced the ratio of oversampling, which duplicates the number of minority classes, to 60 percent because we have few examples in our minority class, and duplicating synthetic data can disrupt the class, resulting in over-fitting of the model, so we increased the ratio of under-sampling. As a result, combining both strategies allows the bias in both classes to be balanced. Because there is no severity information in the obtained dataset, the K-means clustering algorithm is utilized to label the severed data.

Based on the values of the fuel level measurements, the computer segmented the data. Finding the number of k cluster numbers is the most challenging issue in k means clustering; in this work, we employed the elbow technique with in cluster sum square. Which evaluates distance variance between data points inside a cluster.

The cluster numbers for each dataset are found with in cluster sum square starts to decline, resulting in three clusters for each dataset. A cluster with a small WCSS shows lesser variability across the points inside a cluster.

The five datasets BTS1, BTS2, BTS3, BTS4, and BTS5 were used to test four machine-learning algorithms: RF, DT, KNN, and SVM.BTS1 has 3568 total data points, BTS2 has 3848 total data points, BTS3 has 3374 total data points, BTS4 has 3743 total data points, and BTS5 has 3161 total data points with seven independent & one dependent feature. Training takes up 80% of the time, while testing takes up 20%.

In all the datasets, the best accuracy obtained by the RF model is 92-97.4 per cent and f-measure 92 -97 per cent scored.And the DT model result is 84.8-94.3 per cent and f-measure 85 -94 per

cent scored, the KNN model result is 77.1.8-89.3 per cent and f-measure 77 -89 per cent scored while the lowest accuracy obtained by the SVM model is 58..1-73.4 per cent and f-measure 59 - 74 per cent scored as shown in Tables 5.6, 5.7, 5.8, and 5.9.

The DT algorithm preformed well as well, however on each dataset, RF outperformed DT.The RF model can handle datasets with many dimensions. Large dimensions have no effect on the algorithm's performance because it works by finding relevant features. It also offers a way for automatically balancing the dataset when one class is favored, which is why the algorithm outperforms in our situation.

SVM has the worst classification performance compared to other classifiers. SVM is not suitable when there are a large number of attributes due to its high training complexity, and because the resulted cluster label is not equal, there is a class imbalance between the clusters labels, which means SVM, cannot perform well when there are some overlapping classes. The method will be biased towards the minority class while separating hyperplane, which will reduce the algorithm's performance.

Conclusion

The goal of this study was to use machine-learning approaches to build a solution for predicting BTS fuel level severity. The goal of this research was to develop, test, and evaluate machine-learning methods for predicting the severity of BTS fuel levels.

To carry out the study successfully, it was essential to recognize and define fuel level severity, fuel level metrics, and classifications, as mentioned in chapter two. Resampling to balance the classes of the dataset, clustering to group the severed classes to the appropriate severity level, model training using RF, DT, SVM, and KNN, and model testing and finally comparing model using evaluation metrics are also methods used to implement and design for BTS fuel level severity level prediction.

We used five collected BTS fuel level datasets in this research: BTS1, BTS2, BTS3, BTS4, and BTS5.To balance the dataset; we resampled it using a combination of oversampling and under sampling algorithms. Each balanced severed dataset class is divided into three severity levels: Critical, Major, and Minor. Fuel level measures are used to identify those clusters.

The model was created using RF, DT, SVM, and KNN based on these labels. The experiment was conducted on five BTS datasets, obtaining four models for each dataset. When compared to DT, SVM, and KNN classifiers, the RF model produced greater test accuracy on each dataset, with a better test accuracy of 92-97.4 percent and f-measure 92 -97 per cent scored. in the five datasets and.

The study's findings were, overall, quite encouraging. In comparison to the other models RF model performed well on each dataset, hence adopting the RF model in BTS fuel level severity prediction is recommended.

Recommendation and Future Work

As prediction, models have shown that it can estimate fuel severity levels to some extent. As a result, we recommend that telecom service providers develop BTS fuel level severity handling methods that can support companies in taking corrective action before BTS fuel runs out and its network is interrupted, causing a loss of customers and revenue.

The study recommended using a clustering and classification machine learning technique to build models. It is crucial to compare the importance of fuel severity prediction for all base transceiver stations using data from several regional sites.

This study is limited only to the severity level of fuel in the BTS. For a more inclusive prediction model, future research can focus on developing a model for some other categories such as priority and probability of a severity of fuel level in BTS. Additionally the effect of temprater and BTS grid power monitoring system.

references

a.p. bradley. (n.d.). a.p. bradley, the use of the area under the roc curve in the evaluation of machine learning. in a.p. bradley, he use of the area under the roc curve in the evaluation of machine learning.

adnane, o. (2020, july 12). kaggle. retrieved from https://www.kaggle.com/ishivinal/machine-learning-model-evaluation-metrics.

al-fuqaha, a. g. (2015). internet of things: a survey on enabling technologies, protocols, and applications. communications surveys & tutorials, . ieee, 17(4), 2347-2376.

almer(2015), h. (2015). machine learning and statistical analysis are used to predict fuel usage in heavy trucks. kth university, stockholm, sweden. master's thesis. . stockholm, sweden.: , kth university, stockholm, sweden.

anna chlingaryan. (2018). machine learning approaches for crop yield prdicion and nitrogen statsus estimation in precision agriculture. computers and electronics in agriculture vloume 151,augest,2018 pages 61-69, 61-69.

ayodele, t. o. (2010, feb 1). types of machine learning algorithms. retrieved from www.intechopen.com: https://www.intechopen.com/chapters/10694

banerjee, p. (2021, augest 1). kaggale. retrieved from https://www.kaggle.com/prashant111/knn-classifier-tutorial.

bhoraskar, a. ((2012)). prediction of fuel consumption of long haul. department of cognitive robotic, department of cognitive robotic. delft university of technology, the netherlands: department of cognitive robotic.

board. (2019, dec 29). retrieved from www.boardinfinity.com: https://www.boardinfinity.com/blog/the-top-5-types-of-clustering-algorithms-data-scientists-should-know/

dabbura, i. (2018, feb 17). tds. retrieved from towardsdatascience.com: https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a

desheng, c. b. (2021, april 9). study on flow field and measurement characteristics of a small-bore ultrasonic gas flow meter. volume: 54 issue: 5-6, page(s): 554-564, page(s): 554-564.

deyu, l. (2014). data mining approach to analyze mobile telecommunications network quality of service: the case of ethio-telecom (doctoral dissertation, aau). information scienece. addis abeba: aau.

ebude antem, y. e. (2018). automated system for monitoring fueling operation on telecommunicationtower sites. institute for mathematical sciences. padova,italy: aims.

edem donald, e. c. (2018, february). iot based monitoring of generator's fuel & battery levels in base station cell sites with sms aler. edem donald , ezeofor chukwunazo '' iot based monitoring of generator's fuel & battery levels in base station cell sites with sms alert''february 2018, 11.

edomdonald, e. c. (2018, february). iot based monitoring of genertor fuel & battery lveles in base station cell sites with sms alert. . global scintific journal, 9.

ethiotelcom. (2016, july 08). http://www.ethiotelecom.et. . retrieved from http://www.ethiotelecom.et.: http://www.ethiotelecom.et.

ethiotelecom. (2021). bulk sms. addis abeba, aa, ethipia: ethiotelecom.

h. yang. (2013,). h. yang, "data preprocessing-chapter 3," citeseerx, 2013, [online]. available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.467.29&rep=rep1&type= pdf. in h. yang, data preprocessing (p. chapter 3).

hiremath, n. k. (2015). intelligent fuel theft detection using microcontroller arm7. international journal of engineering research & technology. hiremath, n., kumbhar, m., pathania, a. s., and patil, v. (2015). intelligent fuel theft detection using microcontroller arm7. international journal of engineering research & technology, 4(09)., 4(09).

http://www.ethiotelecom.et. (n.d.).

http://www.ethiotelecom.et. (n.d.).

https://forms.gle/akdwhn1q1jvtpvn17. (2022, april 10). retrieved from https://docs.google.com/forms/d/e/1faipqlsepkqw4mu4mp2uvyddup1k1latqrxyxfr4ecezslku5cyyhiq/viewform

hurwitz, j., & kirsch, d. (2018). machine learning. 111 river st.: john wiley & sons, inchoboken, nj 07030-5774.

ieeeinternet things j., v. 1.–9. (ed.). (feb. 2014.). j. a. stankovic, "research directions for the internet of things, ". pp. 3–9,.

j. a. stankovic. (2014.). "research directions for the internet of things. j. a. stankovic, "research directions for the internet of things, “ieeeinternet things j., vol. 1, no. 1, pp. 3–9, feb. 2014., pp. 3–9, .

j. a. stankovic, ". d.–9. (feb. 2014.). research directions for the internet of things, “ieeeinternet things . j., vol. 1, no. 1, pp. 3–9, feb. 2014., pp. 3–9.

j. han, m. k. (2012). data preprocessing. pp. 83–124.

j.s.r. jang c.t.sun, a. e. (1997). neuron-fuzzy and soft computing: a computational approach to learning and machine intelligence. .. massachusetts,usa: prentice-hall,inc.

jain, v. (2022, jan 31). analyticsvidhya. retrieved from www.analyticsvidhya.com: https://www.analyticsvidhya.com/blog/2022/01/introduction-to-knn-algorithms/

jaiswal, s. (2021, jan 5). java point. retrieved from supervised machine learning - javatpoint: https://www.javatpoint.com/supervised machine learning - javatpoint

jemal, m. z., & abebe, m. (2021). software defect severity level prediction using machine. adama: astu.

johnson, b. (2021, december 11). www.guru99.com/supervised-machine-learning.html. retrieved from https://www.guru99.com/supervised-machine-learning.html.

johnson, d. (2021, december 11). guru99. retrieved from https://www.guru99.com/supervised-machine-learning.html.

johnson, d. (2022, feb 12). guru99. retrieved from https://www.guru99.com/unsupervised-machine-learning.html.

k. r. darshan and k. r. anandakumar. (2015). a comprehensive review on usage of internet of things (iot) in healthcare systems, . k. r. darshan and k. r. anandakumar, "a comprehensive review on usage of internet of things (iota) in healthcare systems, "international conference on emerging research in electronics, computer science and technology (icerect), mandya, india, 2015, pp. 13, 9.

kaggle. (2017). kaggle.com. retrieved from https://www.kaggle.com/ar2017/basics-of-feature-selection-with-python.

kaggle. (2021, sep 12). https://www.kaggle.com/prashant111/a-reference-guide-to-feature-engineering-methods. retrieved from kaggle.

larson, r. a. (2014., 10 1). mathworld.wolfram.com. retrieved from https://mathworld.wolfram.com/contingencytable.html.

li, j. h. (2019). notice of retraction countermeasure research about developing internet of things econom. li, j., huang, z., & wang, x. (2011, may). notice of retraction countermeasure research about developing internet of things economy: a case of hangzhou city.2011 international conference one-business and e-government (icee)., 11.

m. sokolova, n. j. (2018, february). m. sokolova, n. japkowicz, s. szpakowicz, beyond accuracy, f-score and roc: a family of. m. sokolova, n. japkowicz, s. szpakowicz, beyond accuracy, f-score and roc: a family of, 9.

miller, m. (2015). miller, m. (2015).the internet of things: how smart t.v.s, intelligent cars, smart homes, and smart cities are changing the world. pearson education., 12.

mohammad asif khan, a. w. ((2017)). fuel monitoring system for cellular sites. mohammad asif khan, ahmad waa's, q. u. k. s. k. (2017). context-aware fuel monitoring system for cellular sites. international journal of advanced computer science and applications (ijacsa), 8(8):281–286., 8(8):281–286.

o. arbelaitz, i. g. (2019). applying resampling methods for imbalanced datasets to not so imbalanced datasets. 977–988.

palo alto, c. ((2014-2018)). mobile statistics, technology market research firm. the radiate group, inc. (2014-2018). mobile statistics, technology market research firm. palo alto, ca, usa , 5.

pykes, k. (2020). "oversampling and undersampling. a technique for imbalanced.

r. mohammed, j. r. (2020). "machine learning with oversampling and undersampling techniques : overview study and experimental results. 243–248.

rachel, w. (2020, augest thursady). (monkeylearn.com). retrieved from https://monkeylearn.com/blog/classification-algorithms/ 5 types of classification algorithms in machine learning (monkeylearn.com).

ray, s. (2017, sep 13). analyticsvidhya. retrieved from www.analyticsvidhya.com: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

s. garcia, s. r.-g. (2016). “big data preprocessing: methods and prospects,” big data anal., vol. 1, no. 1, pp. 1–22, 2016, doi: 10.1186/s41044-016-0014-0.

s. garcia, s. r.-g. (2016). big data preprocessing: methods and prospects. big data anal., vol. 1, no. 1, pp. 1–22, 2016, doi: 10.1186/s41044-016-0014-0., pp. 1–22.

savivaresearch. (2013, may). the rise of the tower bussiness. 15.

sayali barde, s. k., pakhal, a., & josh, p. v. (2017). fuel level monitoring system using gsm. ijste - international journal of science technology & engineering | volume 3 | issue 07 | january 2017, 1-3.

self, g. (2018, feb 5). towards data science. retrieved from https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68.

shin, t. (2021, jan 4). kdnuggets. retrieved from www.kdnuggets.com: https://www.kdnuggets.com/2021/01/machine-learning-algorithms-2021.html

sugumar, r. v. (2020). real time fuel monitoring system for diesel generator using internet of things (iot). 3, 168–171. sugumar, r., vignesh, t., surya, d. r., sanjevi, t., & durgadevi, k. (2020). real time fuel monitoring system for diesel generator using internet of things (iot). 3, 168–171., 3, 168–171.

upasana. (2022, jan 05). edureka! retrieved from https://www.edureka.co/blog/machine-learning-algorithms/.

wolff, r. (2020, august 26th,). monkeylearn. retrieved from monkeylearn.com: https://monkeylearn.com/blog/classification-algorithms/

Appendix A:1 Sample Source Code

Appendix A.1: A sample code for data collection.

```
#include<HCSR04.h>
#include <SoftwareSerial.h>
Software Serial my Serial (03, 04); // RX/TX
Volatile bool button Pressed = false;
Cost into echo Pin = 10; // Echo Pin of Ultrasonic Sensor
Cost into trig Pin = 9; // Trigger Pin of Ultrasonic Sensor
//into chk;
//cost into button Pin = 2;
//defines variables
long   duration;
float distanceCm;
float fuelLevel;
float fuelLevel_percent;
float Y;
char celli;
void setup()
{

  Serial.begin(9600);
  Serial.println ("CLEAR DATA");
  Serial.println ("CLEAR SHEET");
  Serial.println("LABEL,Time,Timer,Date,CellId,distanceCm,Fuel                    level(lt),
fuelLevel_percent,Y");
  //lcd.begin(16,2);
  mySerial.begin(9600);   // Setting the baud rate of GSM Module
  //Serial.begin(9600); // Starting Serial Communication
  delay(100);
  PinMode (trig Pin, OUTPUT); // Sets the trig Pin as an Output
  PinMode (echo Pin, INPUT); // Sets the echo Pin as an Input
  delay (1000);
```

```
}

 void loop()
 {
 long duration,cm;
 digitalWrite(trig Pin, LOW);
 delayMicroseconds(2);
 digitalWrite(trig Pin, HIGH);
 delayMicroseconds(10);
 digitalWrite(trig Pin, LOW);
 duration = pulseIn(echo Pin, HIGH); // using pulsin function to determine total time
 //inches = microsecondsToInches(duration); // calling method
 distanceCm = microsecondsToCentimeters(duration); // calling method
  float distanceCm = duration * 0.034 / 2;
  float fuelLevel =(((150 - distanceCm ) * 150 * 150 * 1000)/1000000);
  float fuelLevel_percent =((150 - distanceCm ) * 100) / 150;
  float celli = 1650;
  float Y;
   if(distanceCm<36.5)
   Y = 1;
   else if(distanceCm>=36.5 && distanceCm<75)
   Y = 0.5;
   else if(distanceCm>=75 && distanceCm<=150)
   Y = 0;
   else
   Y = 0;
   // if(y=1)
   //Y = Full;
   // else if(y=0.5)
   // Y = Medium;
    // else
   // Y = Critical;
   //if(distanceCm=='L')
```

```
 // y='0';
 //else if(distanceCm=='M')
 //y='0.5';
 //else (distanceCm=='H');
 // y=1;
 // Serial.print(inches);
// Serial.print("in ");
Serial.print(cm);
Serial.print("cm");
Serial.println();
Serial.print("DATA,TIME,TIMER,DATE");
Serial.print(",");
Serial.print(celli);
Serial.print(",");
Serial.print(distanceCm);
Serial.print(",");
Serial.print(fuelLevel);
Serial.print(",");
Serial.print( fuelLevel_percent);
Serial.print(",");
Serial.print(Y);
Serial.println();
mySerial.println("AT+CMGF=1");    //Sets the GSM Module in Text Mode
delay(1000);  // Delay of 1000 milli seconds or 1 second
mySerial.println("AT+CMGS=\"+251911490874\"\r"); // Replace x with mobile number
delay(1000);
mySerial.println("JAFERA 165O FUEL LEVEL is : ");// The SMS text you want to send
delay(1000);
mySerial.print( fuelLevel_percent);
delay(1000);
//mySerial.println("AT+CMGS=\"+251911490728\"\r"); // Replace x with mobile number
//mySerial.println("JAFERA 165O FUEL LEVEL IN % is : ");// The SMS text you want to send
//mySerial.print(fuelLevel_percent);
```

```
    //delay(1000);
    //mySerial.println("AT+CMGS=\"+251911490269\"\r"); // Replace x with mobile number
    //mySerial.println("JAFERA 165O FUEL LEVEL IN % is : ");// The SMS text you want to send
    //mySerial.print(level_percentage);
   // delay(1000);
    mySerial.println((char)26);// ASCII code of CTRL+Z
    delay(1000);
    delay(2000);
 }
  // long microsecondsToInches(long microseconds) // method to covert microsec to inches
//{
 //return microseconds / 74 / 2;
//}
long microsecondsToCentimeters(long microseconds) // method to covert microsec to centimeters
   {
   return microseconds / 29 / 2;
   }
```

Appendix A.2: A sample code for data visualization.

```
from sklearn.metrics import accuracy_score, balanced_accuracy_score
from sklearn.metrics import classification_report, roc_curve, auc, roc_auc_score
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from scipy.stats import norm
from sklearn.preprocessing import StandardScaler
from scipy import stats
%matplotlib inline
import os
for dirname, _, filenames in os.walk('C:/Users/ETC/Documents/Python Scripts/fueldata.csv'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

for dirname, _, filenames in os.walk('C:/Users/ETC/Documents/Python Scripts/fuelKNNdata.csv'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
        # set seed for reproducibility
        # read in all our data
nfl_dataset= pd.read_csv("C:/Users/ETC/Documents/Python Scripts/fueldata.csv")
sf_dataset = pd.read_csv("C:/Users/ETC/Documents/Python Scripts/fueldata.csv")
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
np.random.seed(0)

import warnings
warnings.filterwarnings('ignore')
#importing the data
```

```
dataset = ('C:/Users/ETC/Documents/Python Scripts/fueldata.csv')
dataset = pd.read_csv("fueldata.csv")
dataset2 = ('C:/Users/ETC/Documents/Python Scripts/fuelKNNdata.csv')
dataset2 = pd.read_csv("fuelKNNdata.csv")
#data visualization
sns.pairplot(dataset.drop(["Start Time","End  Time","Elapsed  hours","Date","Month"], axis=1), hue="Severity of fuel level", size=2)
# view dimensions of dataset
dataset.shape
# preview the dataset

dataset.head()
# preview the dataset Now,
dataset2.head()
dataset.shape
#summary of dataset
dataset2.info()
# look at no missing points in the first ten columns happened
missing_values_count[0:10].sample(5)
# drop redundant columns from the dataset which does not have any predictive power column from dataset
dataset2.drop(labels=['Start Time','End Time','Elapsed hours','Date','Month'],axis=1,inplace=True)
# Frequency distribution of values in variables
for var in dataset2.columns:
 print(dataset[var].value_counts()).sample(5)
# Frequency distribution of values in variables
for var in dataset2.columns:
 print(dataset2[var].value_counts())
dataset2.dtypes
dataset2.isnull().sum()
dataset2['Severity of fuel level'].value_counts()
dataset2['Severity of fuel level'].value_counts()/np.float(len(dataset2))
```

```
print(round(dataset2.describe(),5))

#Univariate plots to Check the distribution of variables
plt.rcParams['figure.figsize']=(12,10)

dataset2.plot(kind='hist', bins=4, subplots=True, layout=(5,2), sharex=False, sharey=False)
plt.show()
correlation = dataset2.corr()
correlation['Severity of fuel level'].sort_values(ascending=False)
#Declare feature vector and target variable
X = dataset2.drop(['Severity of fuel level'], axis=1)
y = dataset2['Severity of fuel level']
# split X and y into training and testing sets
from sklearn.model_selection import train_test_split
X_train, X_test, x_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
# check the shape of X_train and X_test
X_train.shape, X_test.shape
# check data types in X_train
X_train.dtypes
# check missing values in numerical variables in X_train
X_train.isnull().sum()
# check missing values in numerical variables in X_test
X_test.isnull().sum()
X_train.head()
#featuring scale
cols = X_train.columns
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
X_train = pd.DataFrame(X_train, columns=[cols])
X_test = pd.DataFrame(X_test, columns=[cols])
X_train.head()
```

```
# import KNeighbors ClaSSifier from sklearn
from sklearn.neighbors import KNeighborsClassifier
# instantiate the model
knn = KNeighborsClassifier(n_neighbors=3)
# fit the model to the training set
knn.fit(X_train, x_train)
#Predict test-set results
y_pred = knn.predict(X_test)
y_pred
# probability of getting output as 1 - Less critical
knn.predict_proba(X_test)[:,0]
#check accuracy
from sklearn.metrics import accuracy_score
print('Model accuracy score: {0:0.4f}'. format(accuracy_score(y_test, y_pred)))
#Now, I will compare the train-set and test-set accuracy
y_pred_train = knn.predict(X_train)
print('Training-set accuracy score: {0:0.4f}'. format(accuracy_score(x_train, y_pred_train)))
#Check for overfitting and underfitting
print('Training set score: {:.4f}'.format(knn.score(X_train, x_train)))
print('Test set score: {:.4f}'.format(knn.score(X_test, y_test)))
#Compare model accuracy with null accuracy
y_test.value_counts()
# check null accuracy score
null_accuracy = (672/(672+120+108+57+20))
print('Null accuracy score: {0:0.4f}'. format(null_accuracy))
y_test.value_counts()
#Rebuild kNN Classification model using different values of k
# instantiate the model with k=5
knn_5 = KNeighborsClassifier(n_neighbors=5)
# fit the model to the training set
knn_5.fit(X_train, x_train)
# predict on the test-set
y_pred_5 = knn_5.predict(X_test)
```

```
print('Model accuracy score with k=5 : {0:0.4f}'. format(accuracy_score(y_test, y_pred_5)))
# instantiate the model with k=6
knn_6 = KNeighborsClassifier(n_neighbors=6)
# fit the model to the training set
knn_6.fit(X_train, x_train)
# predict on the test-set
y_pred_6 = knn_6.predict(X_test)
print('Model accuracy score with k=6 : {0:0.4f}'. format(accuracy_score(y_test, y_pred_6)))
# instantiate the model with k=7
knn_7 = KNeighborsClassifier(n_neighbors=7)
# fit the model to the training set
knn_7.fit(X_train, x_train)
# predict on the test-set
y_pred_7 = knn_7.predict(X_test)
print('Model accuracy score with k=7 : {0:0.4f}'. format(accuracy_score(y_test, y_pred_7)))
# instantiate the model with k=8
knn_8 = KNeighborsClassifier(n_neighbors=8)
# fit the model to the training set
knn_8.fit(X_train, x_train)
# predict on the test-set
y_pred_8 = knn_8.predict(X_test)
print('Model accuracy score with k=8 : {0:0.4f}'. format(accuracy_score(y_test, y_pred_8)))
# instantiate the model with k=9
knn_9 = KNeighborsClassifier(n_neighbors=9)
# fit the model to the training set
knn_9.fit(X_train, x_train)
# predict on the test-set
y_pred_9 = knn_9.predict(X_test)
print('Model accuracy score with k=9 : {0:0.4f}'. format(accuracy_score(y_test, y_pred_9)))
# instantiate the model with k=9
knn_18 = KNeighborsClassifier(n_neighbors=18)
# fit the model to the training set
knn_18.fit(X_train, y_trai# predict on the test-set
```

```
y_pred_18 = knn_18.predict(X_test)
print('Model accuracy score with k=18 : {0:0.4f}'. format(accuracy_score(y_test, y_pred_18)))
# Print the Confusion Matrix with k =3 and slice it into four pieces
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print('Confusion matrix\n\n', cm)
print('\nTrue Positives(TP) = ', cm[0,0])
print('\nTrue Negatives(TN) = ', cm[1,1])
print('\nFalse Positives(FP) = ', cm[0,1])
print('\nFalse Negatives(FN) = ', cm[1,0])
# visualize confusion matrix with seaborn heatmap
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
import seaborn as sns
f, ax =plt.subplots(figsize = (5,5))
sns.heatmap(cm,annot = True, linewidths= 0.5, linecolor="red", fmt=".0f", ax=ax)
plt.title('KNN K3 Kernel \nAccuracy:{0:.3f}'.format(accuracy_score(y_test, y_pred)))
plt.xlabel("y_pred")
plt.ylabel("y_true")
plt.show()
#I print a classification report as follows:-
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred_7))
#Classification accuracy
TP = cm_7[0,0]
TN = cm_7[1,1]
FP = cm_7[0,1]
FN = cm_7[1,0]
# print classification accuracy
classification_accuracy = (TP + TN) / float(TP + TN + FP + FN)
print('Classification accuracy : {0:0.4f}'.format(classification_accuracy))
# print classification error
```

```
classification_error = (FP + FN) / float(TP + TN + FP + FN)
print('Classification error : {0:0.4f}'.format(classification_error))
#Mathematically, precision can be defined as the ratio of TP to (TP + FP).
# print precision score
precision = TP / float(TP + FP)
print('Precision : {0:0.4f}'.format(precision))
recall = TP / float(TP + FN)
print('Recall or Sensitivity : {0:0.4f}'.format(recall))
#True Positive Rate is synonymous with Recall.
true_positive_rate = TP / float(TP + FN)
print('True Positive Rate : {0:0.4f}'.format(true_positive_rate))
false_positive_rate = FP / float(FP + TN)
print('False Positive Rate : {0:0.4f}'.format(false_positive_rate))
specificity = TN / (TN + FP)
print('Specificity : {0:0.4f}'.format(specificity))
false_positive_rate = FP / float(FP + TN)
print('False Positive Rate : {0:0.4f}'.format(false_positive_rate))
specificity = TN / (TN + FP)
print('Specificity : {0:0.4f}'.format(specificity))
y_pred_prob = knn.predict_proba(X_test)[0:10]
y_pred_prob
# store the probabilities in dataframe
y_pred_prob_df = pd.DataFrame(data=y_pred_prob, columns=['Prob of -Least Critical(0)', 'Prob
of -Less Critical(1)','Prob of - Critical(2)','Prob of -More Critical(3)','Prob of -Most Critical(4)'])
y_pred_prob_df
y_pred_0 = knn.predict_proba(X_test)[:, 1]
# plot histogram of predicted probabilities
# adjust figure size
plt.figure(figsize=(6,4))
# adjust the font size
plt.rcParams['font.size'] = 12
# plot histogram with 10 bins
plt.hist(y_pred_0, bins = 10)
```

```
# set the title of predicted probabilities
plt.title('Histogram of predicted probabilities of Least Crtical(0)')
# set the x-axis limit
plt.xlim(0,4)
# set the title
plt.xlabel('Predicted probabilities of Least Crtical(0)')
plt.ylabel('Frequency'
```

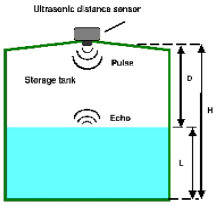
Appendix B: Tools used

Fuel sensor

Ultrasonic fuel level sensor is a fuel-monitoring device, which is designed for the fuel management needs of company or another user. The sensor can be used to monitor level of fuel in tanker.

Figure 19 HC-SR04 fuel sensor

Shrivastava et al. demonstrated that using the ultrasonic sensor for measuring distance, as the distance increases the error of measurement reduces provided it is less than the maximum distance [49, 50].Using the sensor to calculate the volume of fuel in a rectangular tank with height Hmax, length L and width W. The sensor is connected to a timer, which measures the time of flight (ΔT)



Representation of fuel tank with ultrasonic sensor for volume measurement

To calculate the range R (distant between sensor and liquid surface);

$$R = \frac{u\Delta T}{2} \quad \text{----------} \quad (3.2.1)$$

where u is wave propagation velocity. The level of water F_{uel} is calculated:

$F_{uel} = H^{max} - R$.

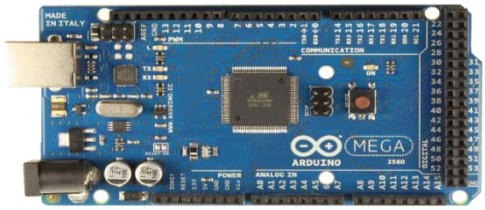


Figure 21 Atmega2560microcontroller:

GSM/GPRS Module

4G/GPRS specifications:

Dual-band 900MHz EGSM, 1800MHz DCS-connect onto any global GSM network with any 4G SIM.

Make and receive voice calls using a headset with a microphone

Send and receive SMS messages

Send and receive GPRS data (TCP / IP, HTTP, etc.)

Appendix C

ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY

School of Electrical Engineering and Computing

Department of Computer Science and Engineering

Questionnaire to be filled by telecom power technician.

Dear Respondents

This Questionnaire is prepared by post graduate program student of Adama Science and Technology University school of Electrical Engineering and Computing department of Computer Science and Engineering to conduct study on " IoT based Generator Fuel Level Severity Prediction System for Cell Towers using Machine Learning" as part of the requirement for MSc degree. The purpose of the the questionnaire is to collect firsthand information on the topic under caption. Since the information required is for academic purposes and be sure that, there is nothing behind the study and all your response will be kept secret (confidential).

Hence, please lend me few minutes of your time to fill out the questionnaire as genuinely and completely as possible and no need to write your name on the questionnaire.

I remain grateful to your kind cooperation

I. Personal Data

Please complete this section by ticking in the box provided

1. Educational Background: Diplor ☐ Degre ☐ MS ☐ other _____
2. Title: Power technici ☐ RAN tec ☐ Supervis ☐ Manag ☐
3. Year of service: 1-3 yea ☐ 4-7 year ☐ 8-10 year ☐ above 10 ☐

II. Main Research Questions

1. How many BTS exist in your branch _____?
2. How many BTS has power generator _____?
3. Would you please fill the following BTS fuel tank information?

Fuel Tank					
Shape	Capacity(lt)	eight	vidth	ength	Number(qty)

4.How is BTS fuel severity determined? Manually ☐ Automatically ☐

4.1 What is the impact if it is manually determind?

1._____

2._____

3._____

4.2 Which factor from the above information used to determine fuel level severity?_____.

5 .When BTS fuel level severityis said to be Normal?_____.

a. fuel height level in tank is ≥ 3/8
b. ¼ < fuel height level in tank ≤ 3/8
c. 1/8 < fuel height level in tank ≤ ¼
d fuel height level in tank is below 1/8

6. When BTS fuel level severityis said to be critical?_____.

a. fuel height level in tank is ≥ 3/8
b. ¼ < fuel height level in tank ≤ 3/8
c. 1/8 < fuel height level in tank ≤ ¼
d fuel height level in tank is below 1/8

7. When BTS fuel level severityis said to be major?_____.

a. fuel height level in tank is ≥ 3/8
b. ¼ < fuel height level in tank ≤ 3/8
c. 1/8 < fuel height level in tank ≤ ¼
d fuel height level in tank is below 1/8

8. When BTS fuel level severityis said to be minor?_____.

a. fuel height level in tank is ≥ 3/8
b. ¼ < fuel height level in tank ≤ 3/8
c. 1/8 < fuel height level in tank ≤ ¼
d fuel height level in tank is below 1/8