The Performance of Classification Method in Telco Customer Trouble Ticket Dataset

Fauzy Che Yayah , Khairil Imran Ghauth*, and Choo-Yee Ting

.

***Abstract*—A customer trouble ticketing system (CTT) is an organization's tool to track the detection, reporting, and resolution of tickets submitted by customers. It also comprises a summary of the issue reported, the status of the ticket, the incident information, and the approach that was previously utilized to resolve the problems. The technician's skill set and experience rely solely on completing the task without the right direction on which area to focus on first. As a result of this manual classification of a trouble ticket, it will be necessary to build methodologies for predicting future resolution codes. The research for this report is mainly focused on one of the telco companies in Malaysia. This study result assists the telco engineer, and the specialists resolve each issue in a very short amount of time. Additionally, the classification of the trouble ticket resolution code method used in this study will indicate the characteristics of each issue that is being investigated. The relationship between events is feasible to discover by exploring the root cause. It is critical to establish a link between recent events and events in the previous. Because of current data mining limitations, the study needs to be more comprehensive. Data processing methods are being implemented within big data platforms to overcome the limitation of data scalability, enhance classification accuracy, and increase computation speed. The research work will continue to progress in the direction of big data centricity. Some of the most effective approaches for big data integration and machine learning will be discussed in this paper. Throughout the experiment, any problems will be explained, as well as the solutions to each situation. A wide range of research subjects will be discussed, including construction classification models for trouble tickets. To achieve reasonable accuracy, a few customized transformations are required. The data set's custom parameter optimization process will further increase the classification trouble ticket's efficiency. However, greater processing capacity is necessitated to use multiple parallel classifiers such as Bayes, Decision-Tree, and Rule-Based with help of bigdata framewrks such as Spark. According to the study, an increase of 8% classification performance substantially influences service recovery time, customer satisfaction, and preventative maintenance expenses in the telco industry.**

***Index Terms*—Trouble Tickets, Sublanguage, Classification, Single Machine, Hadoop , Spark**

I. Introduction

TELECOMMUNICATION companies (telcos) maintain a trouble ticketing system (CTT) for reporting incidents involving the provision of high-quality service. Each time a ticket needs to be fixed, someone has to think about it, making it more challenging to be sure and less accurate. This study uses machine learning to make things faster and more accurate. In addition, research shows that big data is an ideal way to manage a lot of data. So, this study looks at how to classify trouble tickets using a big data approach to make the process more efficient and accurate. This research only focuses on a few areas of Malaysia where telco service is available across the whole country. Figure 1 shows the zone with the most trouble tickets, where the zone selection is being utilized. For instance, in zone PCG (Puchong), the CTT ratio is higher (8.33) than in SAL (Shah Alam), which is (6.78). Telco thinks that the current solution takes a long time to figure out the real problem. Currently, the study is confined to telco-related keyword results and a few simple features, such as term occurrences. The location, hardware, name, events, date, network data, and database model attributes can be used to show how the consolidated dataset is linked and how it looks like there are hidden relationships and patterns in the data.

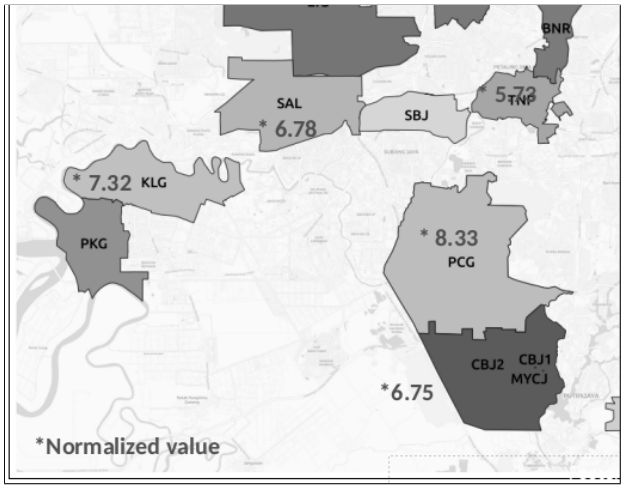


Fig. 1. Selected Telco Serviceable Zone (Normalized CTT Volume)

II. Related Work

Many of the projects are based on improving the classifier's original formula and engineering features so that they can be used as accurately as possible. Some researchers have developed ways to speed up the traditional algorithm's [1] classification process. In some research, a parallel technique improves the optimization process and overcomes the dimensionality issues [2] associated with large data sets.

Using the MapReduce framework [3] , Dai and Ji have developed a novel C4.5 decision tree approach that performs significantly better. They want to find solutions to problems that develop when a decision tree classifier is used to handle a huge dataset, which requires a significant amount of processing time. It is necessary to perform some computations

Fauzy Che Yayah is a PhD candidate at Multimedia University, Cyberjaya, Malaysia. (e-mail: akunyer@gmail.com)

Khairil Imran Ghauth , the corresponding author is a Senior Lecturer at the Faculty of Computing and Informatics, Multimedia University, Cyberjaya, Malaysia. (e-mail: khairil-imran@mmu.edu.my)

Choo-Yee Ting is a Professor at the Faculty of Computing and Informatics, Multimedia University, Cyberjaya, Malaysia. (e-mail: cyting@mmu.edu.my)

on an external device when data cannot be kept in memory, which increases the cost of I/O. As a result, the present C4.5 decision tree method was rewritten as a MapReduce version that featured mapping and reduction operations. Many in-depth tests were conducted using a combination of big datasets [4] and various data transformations format, which can be lead to lower communication and computational costs. When the C4.5 decision tree in the MapReduce classification algorithm is run, the results demonstrate how efficient it is to run as well as how it can be used with any dataset, regardless of its size. This approach will be used in this study because the idea of putting MapReduce together with the traditional algorithm is possible. However, the latency of processing is still an issue for this method.

For optimizing [5] the decision-tree method, Fang Yuan suggested a unique MapReduce-based strategy. A genetic algorithm (GA) based on the decision tree technique was developed to run multiple instances in parallel. Using this novel approach, the optimized rules for the decision tree algorithm will be identified and determined. It will convert the rules of the decision tree into a set of GA chromosomes. Then it will utilize the fitness function to determine the fitness of each chromosome by accomplishing cross-over, selection, and mutation on each chromosome. The last stage is to determine whether or whether the chromosome that has been altered is a viable candidate for the rules of adding, consolidating, or removing information on it. Due to the iterative nature of the GA approach [5], it can be possibly executed concurrently in Hadoop. The MapReduce matrix function allows it to finally reconstruct the decision tree structure in an independent and parallel manner, utilising both the reduction and combination of the data. Despite this, just one method has been successfully changed to enable MapReduce, and the only other benefit is a little increase in computing efficiency.

Shah and Patel developed a novel method for categorising diabetic datasets in a distributed computing environment [6]. When performing classification within Hadoop, it is advised that to use the Spark [7] framework. The fundamental goal of the study is to assess the output data in terms of critical characteristics such as recall and accuracy and among others. The experiment outcome uses several algorithms with and without missing value impacts during the imputation process. Only the built-in Spark algorithms can be used in this method. Naive Bayes [8] was used in this study to reach the goals.

Du and Li [9] accomplish parallelism on the Hadoop network by utilizing the K-nearest neighbours (KNN) [10] technique in MapReduce. This purpose is to accelerate the categorization process and decrease the time required for computing. There are several passages of text in the dataset for this study that are crucial to the classification of public opinion by the public network. Using the experiment, they were able to successfully adapt the classical KNN into a MapReduce application. As a result, categorization became faster and more efficient while dealing with a larger dataset.

Chen developed a new classification approach [11] utilizing the Softmax algorithm in conjunction with MapReduce. With Softmax's method, the most significant difficulty is that it restricts the processing of large data sets, increasing computing time and making the system less efficient. The application of K-fold Cross-Validation throughout the design phase increases the accuracy of the model evaluation and aids in the prevention of overfitting problems in the model. The final model, in conclusion, indicates how improved performance and computing speed may be accomplished in a distributed system like Hadoop.

The traditional classifier is employed in most existing research due to its simplicity and functionality. Making the switch from the current classifier code to MapReduce compliant code is not easy and requires a thorough understanding of the methods. In order to meet the primary criteria, the present traditional classifier approach must be divided into many steps. To convert algorithms to MapReduce, one must realize that the algorithm components must run in parallel. Comparing key-pair and split ratio results determines the requirements. In most past studies, Cross-Validation was not used to resolve overfitting issues and estimate model performance, which is critical. Apache Spark and other data processing frameworks such as it are employed in the proposed solution to deal with the aforementioned difficulty because they are both faster and more adaptive, which makes them more efficient in their operation.

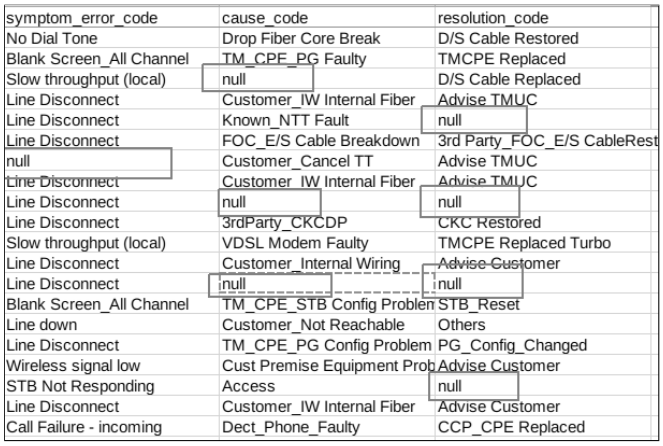
III. Dataset

Every hour, more than a hundred new customer issues are raised in the trouble tickets system. Because of how quickly the data changes, it is hard to process and needs to be moved to a more extensive storage system like Hadoop. Apache Sqoop was developed to ensure the bulk transfer of data from a trouble ticket system to the Hadoop storage. The increased number of trouble tickets is determined by several factors, including shifts in consumer preferences and unbalanced preferred channels for complaints. The technician who has been appointed may require particular skills and expertise to handle the issues accurately. Without the necessary skills, the technician may approach the case very carefully, making manual determinations for problem resolution that may be incorrect, negatively impacting the customer experience service journey. The associated dataset that must be transferred includes several critical data types, including Service Requests (SR), Customer Trouble Tickets (CTT), Network Trouble Tickets (NTT), Customer Internet Bandwidth (CIB), Customer Profiles (CP), and Customer Internet Service Quality (CISQ).

A snapshot of the CTT dataset has both structured and unstructured columns, as shown in the Figure 2 below. The *description* column contains the CTT's unstructured element, which also serves as a free note space for collecting customer observations during the discussion to attempt to resolve the customer's concerns. Because it contains words closely related to the fault event, this crucial knowledge will be useless and static if not explored. According to Figure 2 also, the raw version of the CTT dataset may have missing values or *null*, which must be corrected before the dataset can be used to construct the analytic model. Dataset imputation may be required to increase the quality of the dataset.

A. Customer Trouble Tickets (CTT)

The central Customer Trouble Tickets repository [12] contains a variety of different forms of information concerning



symptom_error_code	cause_code	resolution_code
No Dial Tone	Drop Fiber Core Break	D/S Cable Restored
Blank Screen_All Channel	TM_CPE_PG Faulty	TMCPE Replaced
Slow throughput (local)	null	D/S Cable Replaced
Line Disconnect	Customer_IW Internal Fiber	Advise TMUC
Line Disconnect	Known_NTT Fault	null
Line Disconnect	FOC_E/S Cable Breakdown	3rd Party_FOC_E/S CableRest
null	Customer_Cancel TT	Advise TMUC
Line Disconnect	Customer_IW Internal Fiber	Advise TMUC
Line Disconnect	null	null
Line Disconnect	3rdParty_CKCDP	CKC Restored
Slow throughput (local)	VDSL Modem Faulty	TMCPE Replaced Turbo
Line Disconnect	Customer_Internal Wiring	Advise Customer
Line Disconnect	null	null
Blank Screen_All Channel	TM_CPE_STB Config Problen	STB_Reset
Line down	Customer_Not Reachable	Others
Line Disconnect	TM_CPE_PG Config Problem	PG_Config_Changed
Wireless signal low	Cust Premise Equipment Prob	Advise Customer
STB Not Responding	Access	null
Line Disconnect	Customer_IW Internal Fiber	Advise Customer
Call Failure - incoming	Dect_Phone_Faulty	CCP_CPE Replaced

Fig. 2. Example of Original Trouble Tickets Dataset

any fault that happened. The data dictionary includes a list of the most critical variables, which is shown in Table I.

TABLE I
DATA DICTIONARY: CUSTOMER TROUBLE TICKET

Column Name	Type	Size	Description
Created Date	Date	10	Created Date
Service_ID	Text	20	Service ID
Symp_Error_Code	Text	20	Fault Code
Cause_Code	Text	20	Cause Fault Code
Resolution_Code	Text	20	Resolution Fault Code
Zone	Text	20	Zone Info
Description	Text	50	CTT Freetext

A few critical columns, such as Login ID, store the login information for each subscriber's Internet subscription. The Symp_Error_Code column contains the symptom error code captured during the fault troubleshooting procedure. When an error code is found, the Cause_Code column is used to store information on the cause of the error code encountered. Finally, the Resolution_Code column contains the code that has been executed after the fault has been identified and fixed.

B. Customer Internet Service Quality (CISQ)

Customer Internet Service Quality [13] provides a variety of different forms of information about customer Internet service quality. Additionally, it is referred to as a summary or metric of the overall efficiency of Internet service. The performance metric is calculated twice a day for a random group of subscribers using system probes set up in specific places. Table II shows that the dataset metadata of the consolidated database when the probing operation was completed.

C. Customer Profiles (CP)

The Customer Profiles [14] dataset provides a comprehensive view of a consumer's business based on demographics, experiences, interests, and values, as well as other information. Whether membership-based or product-based, enterprises can rely on customer profile management as their principal source of sales and revenue. This data set is used in conjunction with the CTT dataset to improve the design, accuracy, and discovery of the predictive research model used in this study. The customer profiles data dictionary is shown in Table III below:

TABLE II
DATA DICTIONARY: CUSTOMER INTERNET SERVICE QUALITY

Column Name	Type	Description
Created Date	Date/Time	CTT Created Date
Login_ID	Text	Subscriber Login
Speed	Text	Subs. Network Info
Type	Text	Subs. Network Type
Admin Status	Numerical	Administration Parameter
Onu Power (Up/Down)	Numerical	Onu Power Up / Down
Onu Temp	Numerical	Onu Temperature
Onu Ranging	Numerical	RTD (Round-Trip-Delay)
Onu Ber (Up/Down)	Numerical	Bit Error Rate Up / Down
Onu CRC (Up/Down)	Numerical	Cyclic Redundancy Check
Olt Pwr (Up/Down)	Numerical	Optical Line Termination
Olt Snr (Up/Down)	Numerical	Signal-To-Noise Ratio
Olt Att (Up/Down)	Numerical	Upstream / Downstream Olt
Olt Max (Up/Down)	Numerical	Maximum Power
Olt Cfg (Up/Down)	Numerical	Configuration
Olt Response Time	Numerical	Response Time in ms

TABLE III
DATA DICTIONARY: CUSTOMER PROFILES

Column Name	Type	Description
Login_ID	Text	Subscriber Login
Installation Date	Date	CTT Created Date
Termination Date	Text	Subscriber Login
Premises Details	Text	Subscriber Premises Info
Installation Date	Text	Subscriber Info
Termination Date	Numerical	Subscriber Info
On-Premise Device Type	Text	Network Info
Fiber Copper Dist. Point	Text	Network Info
Reseller Information	Text	Extra Info
Payment Details	Numerical	Billing Info
Building Exchange Info	Text	Network Info

The CP dataset is imported from the database by the customer service software. This dataset contains information on the client, such as current billing information, order signatures, and payment history. This data collection complements the CTT dataset and will aid in the improvement of the model's design and accuracy.

D. Customer Internet Bandwidth (CIB)

The Customer Internet Usages dataset offers information about the various ways to send data over the Internet. By contrast, bandwidth refers to an Internet connection's capacity. When a customer's Internet connection is disrupted, the central probe machine will pause, recalculate the download capacity, and take a snapshot of the data. If a customer's Internet access is permanently disabled for several minutes or hours, the central probe flags the incident as very unreliable. This scenario must be fed with CTT information because the exterior and internal aspects of the CTT resolution code classification process must be combined for a better CTT resolution code classification method. This dataset's variable descriptions are shown in Table IV.

E. Service Request (SR)

The Service Request [15] dataset contains essential information on customer feedback from their initial contact with the telco call centre for specific problems and requests for resolution. Most SR records are caused by service network disruptions and client endpoint device failures. If the call centre determines that the SR can be resolved over the phone, then the SR status is closed. If the issue requires

TABLE IV
DATA DICTIONARY: CUSTOMER INTERNET USAGES

Column Name	Type	Description
Login_ID	Text	Subscriber Login
Network Info	Text	Nasidentifier, Sessionid, Mac Address, Upload Speed, Download Speed, Terminate-cause, Ipv6 Address, Ipv4 Address, Acct Delay
Timestamp Info	Date	Stop Timestamp, Start Timestamp, Time Stamp
Network Tagging	Text	Calling Station, Package Speed, Service Type

additional investigation, it will be classified as a Customer Trouble Ticket (CTT), and other processes will take over. If the root cause is determined to be network-related [16], the incident will be escalated as a Network Trouble Ticket (NTT). Additional information regarding the SR metadata is available in Table V as follows:

TABLE V
DATA DICTIONARY: SERVICE REQUEST (SR)

Column Name	Type	Description
Created Date	Date	SR Created Date
SR Number	Text	SR Information
Login_ID	Text	Subscriber Login
Service_ID	Text	Service ID
Network Info	Text	Network Domain
Timestamp Info	Date	Time Stamp
Description	Text	SR Freetext Description

F. Network Trouble Tickets (NTT)

The Network Trouble Tickets [17] dataset has statistical patterns about important network events. Researchers could potentially utilize the NTT data to evaluate inference methods, such as Internet traffic abnormalities [18]. This feature allows the organization to solve problems in the future by using the knowledge it learned from a similar situation in the past to assist them. There is no standardized approach in writing for the free text area of each network ticket issued. Table VI contains a description of the variables in this dataset.

TABLE VI
DATA DICTIONARY: NETWORK TROUBLE TICKET

Column Name	Type	Description
Created Date	Date	Created Date
Service_ID	Text	Service ID
Symp_Error_Code	Text	Fault Code
Cause_Code	Text	Cause Fault Code
Resolution_Code	Text	Resolution Fault Code
Aging Info	Text	Equipment Aging Info
Equipment Info	Text	Equipment Model, Equipment ID, Equipment Vendor
Description	Text	NTT Fault Freetext

Additionally, the network trouble ticket system maintains a log of all actions taken up until the ticket is closed. Network trouble tickets and trouble ticket systems are vital to the operation of a network every day. In addition, the history of trouble tickets is an essential tool for network management and research into troubleshooting and maintenance methods.

IV. PROPOSED METHOD

A. The Experiment

This study is separated into two parts based on the hardware configuration. Single Machine is used as the client in this experiment, and a Hadoop cluster is used to store the data in the experiment's second component, which is a Hadoop cluster. This configuration has been made to allow the Rapidminer Radoop to work in conjunction with the Single Machine computer. It is essential that the suggested client (Single Machine) setup in the Table VII is followed exactly as indicated in order for the integration to be complete and functional.

In its overall recommendation, this study suggests that the Linux operating system with extensive memory capacity, more CPU core, and large storage be utilized for reliability, speed and security. It is necessary for the program to function properly that all required software components, such as Java and Spark, as well as programming languages such as Python, SparkR, and Scala, be loaded with the most recent version available. The Hadoop cluster consists of one master node and three computing nodes, which work together to process data. For the optimum output and performance, each node must have the exact hardware specs as the others.

TABLE VII
SINGLE MACHINE AND HADOOP EXPERIMENT CONFIGURATION

Items	Single Machine Configuration	Hadoop Configuration	Class
Operating System(OS)	Ubuntu 20.04 Desktop	Ubuntu 19.04 Server	Software
Hadoop Distribution	Hadoop compatible libraries	Cloudera Enterprise 5.16	Software
Data Modeler Software	Rapidminer 9.10 + Radoop Extension	-	Software
SparkR	2.0.1	2.0.1	Software
R	4.0.5	4.0.5	Software
PySpark	2.4.6	2.4.6	Software
Python	2.7.10	2.7.10	Software
Scala	2.10.5	2.10.5	Software
Java	OpenJDK 1.8	OpenJDK 1.8	Software
Processor	Intel/AMD 16 x CPU Core	a)1x Master Node (128-Core) b)3x Computing Nodes (384-Core)	Hardware
Memory	32GB RAM	a)256GB RAM with 3x Computing Nodes	Hardware
Storage Capacity	1.5 TB	a)1x Master Node 4 TB b)3x Computing Nodes 8 TB	Hardware

B. Data Consolidation

The Figure 3 illustrates a UML design for data consolidation. The primary key (*login_id*) connects the data set to another corresponding dataset via a foreign key. The primary key in the first table's column is a value that serves as a unique identifier. The foreign key is typically stated in a second table with the same value as the first table. It is also possible to put it as (*primary (A) table key value = secondary (B) table value of the foreign key*). A SQL statement is used to invoke the *inner join* process.

The *inner join* returns all rows from both tables that contain all columns. This operation is equivalent to consolidating rows from two or more tables and becoming a raw dataset version.

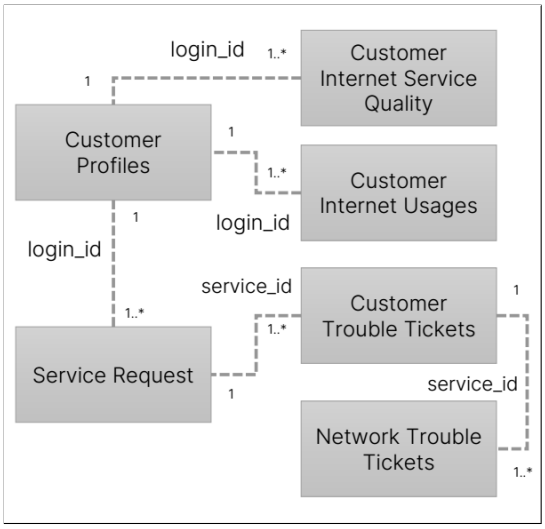


Fig. 3. Data Consolidation and Table Relationship in UML

C. Data Preprocessing

The process of preparing the raw data set for analysis needs considerable work. The method can eliminate duplication, such as that caused by data stitching. To optimize process performance, it is advised that the data preparation process remains within the big data platform, utilizing Hadoop tools such as Apache Hive and Apache Impala. Finally, the dataset may be evaluated using data analysis software such as Rapidminer Studio, and the transformed dataset is ready for the next stage of developing the analytics model. The following summarises the data preparation preprocessing steps as seen in Figure 4:

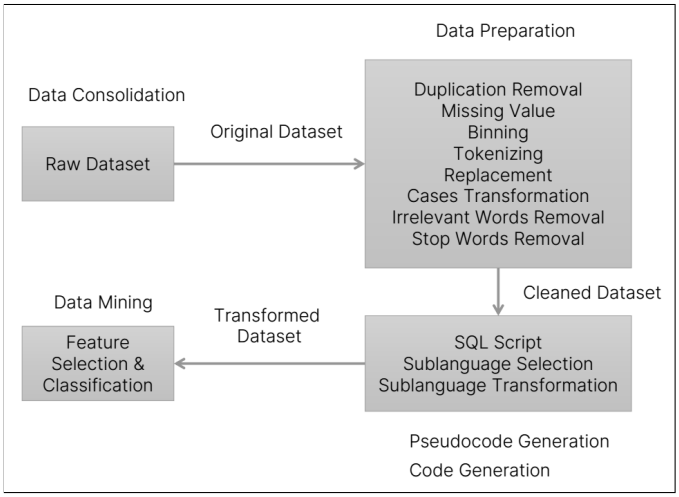


Fig. 4. Data Preprocessing Steps from the Raw Dataset

The following list contains detailed explanations of each process:

1) N-GRAM Keyword Construction - N-GRAM is a continuous sequence of words composed entirely of n elements extracted from the source text. By limiting each definition of the CTT to a single unigram word, a diverse set of concepts is processed. This technique is essential for determining the accuracy of feature selection and classification.
2) Missing Value Replacement - A typical replacement value is used to update any attribute. If any numeric values are missing, substitute the minimum, maximum, or average. Additionally, an imputed value may be used to replace the table's highest frequency value. The benefits of the data completion process are for preventing invalid models coming from the values that are missing or *null*.
3) Discretization by Binning - A data set is stratified when broken down into separate groups called strata. After that, a probability sample is drawn for each category. This method can cut down on samples while still representing the whole dataset. In this study, some datasets with discrete values must be changed to binning format to meet the model classifier's needs.
4) Word Tokenization - The dataset summary contains annotated words about technical details, the resolution method, and the definition of a fault. The text is divided into tokens using this method, dependent on the separator. The effect is a collection of symbols that comprise a single word before it is converted to a word vector for further processing.
5) Token Replacement - The Term Frequency-Inverse Document Frequency (TF-IDF) refers to a group of undefined keywords. Depending on the replacement dictionary [17], this strategy requires the substitution of string handling rules or regular expressions.
6) Transform Cases - This operation standardizes the word case to avoid falsification during the execution of the text processing strategy. In this study, the default setting replaces all words with a lowercase character.
7) Removal of Irrelevant Words - It was found that almost half of the irrelevant terms were unrelated to fault events. The limited amount of protected keywords would speed up data processing. The Document-Term Matrix (DTM) defines word occurrence in the source text.
8) Removal of Stop Words - Not all of the CTT definition's keywords adequately describe the status of the fault. *"line down"* , *"intermittent"*, and *"no backup"* are all examples of frequent technical jargon that must be excluded should be retained in this process.
9) Replacing of Short Abbreviation Words - Specific keywords in the dataset definition utilizes abbreviation terminology that may be unfamiliar to the English default localized dictionary [17]. This needs manually generating a standard dictionary with a list of short abbreviation keywords. Just the relevant abbreviation term is automatically substituted during the text processing stage.
10) Construction of Sublanguage and Dictionary Vector - The description of the CTT data collection required conversion to the textitt vector representation format. The formula describes the primary task: rating the keyword vectors and evaluating each t (k). If any of the k keywords appear in the dataset (t), $t(k)$is set to 1 or 0.

The following steps illustrate the evaluation technique for identifying sublanguages represented in (Figure 5):

1) Term Frequency (tf) - Shows how often a $t(k)$ expression (term, word) occurs in the dataset.
2) Document Frequency (df) - Defines as the dataset

TABLE VIII
THE FINAL TRANSFORMATION TABLE WITH SUBLANGUAGE FEATURES

symptom	zone	alarm	able	agent	assist	router	down
symp1	batu	0.254	0.433	0.323	0.221	0.054	0.04
symp5	kepong	0.323	0.646	0.224	0.214	0.124	0.134
symp6	sa	0.743	0.655	0.563	0.147	0.214	0.123
symp2	subang	0.111	0.232	0.147	0.369	0.248	0.847
symp4	klang	0.543	0.292	0.587	0.784	0.847	0.321
symp5	batu	0.754	0.245	0.369	0.258	0.214	0.444
symp7	bangsar	0.335	0.643	0.235	0.553	0.014	0.632
symp6	kepong	0.532	0.865	0.897	0.196	0.215	0.164
symp2	subang	0.424	0.345	0.223	0.245	0.654	0.036
symp8	sa	0.421	0.212	0.047	0.554	0.747	0.563

numbers containing *t* keywords.

3) Inverse Document Frequency (*idf*) - Determines how the keyword applies to the dataset.
4) *tf x idf* - Define the weighted score for each dataset.

Figure 5 illustrates an example of a CTT description that was recorded when responding to telco subscribers' complaints about specific faults. Based on the custom dictionary produced for each sublanguage group, the collection of highlighted words is recognized as the sublanguage.

As previously explained, this term is used and converted into the matrix format. The preceding phases of the transformation process can be expressed as (Equation 1) below:

$$\text{weighted\_score} = \sum_{i} tf \times idf \tag{1}$$

Stop words are a list of terms that should be avoided when utilizing Natural Language Processing (NLP) to optimize text scanning and processing efficiency. Additionally, it might be characterized as typical English terms. When doing word vector conversion processing, data mining algorithms rely heavily on these records to extract the terms.

The keywords appears in Table VIII have been translated using (Equation 1) and incorporated into the consolidated dataset, becoming the features for the final transformation table. Each keyword now has a significant weighting in the following construction of the analytic model process.

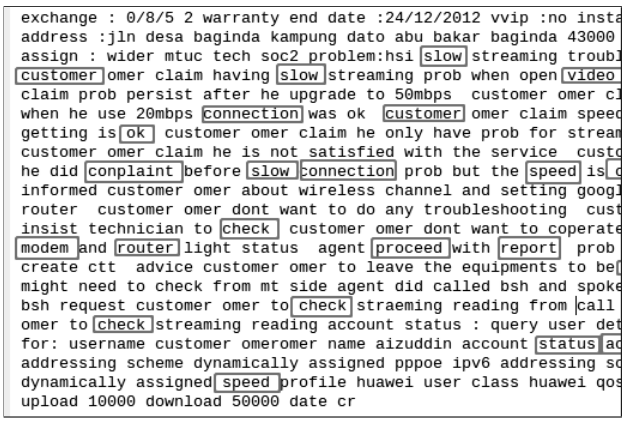


Fig. 5. Sublanguage identification in dataset

D. Model Construction

The percentage of correctly classified samples is usually linked to how well the data is balanced. Data balance is an essential step in predictive modelling. Unbalanced data occurs when one data set class dominates the other. The data discrepancy's fundamental reason could be a common issue. It implies that the difference is caused by factors outside the data space, such as the wrong way the data was collected. If everything is in order and balanced, the following methods contain techniques for improving the model's quality during the model's construction phase.

TABLE IX
TRADITIONAL CLASSIFIER SELECTION METHOD FOR A SINGLE MACHINE

Classification Algorithm	Classifier Type	Classification Time in (seconds)	Classification Accuracy	Classifier Model Explainable	Selected Classifier
Conjuctive Rule	Rule Based	15	10.63 %	✖	✖
Decision Table	Rule Based	125	71.47 %	✔	✖
DTNB	Rule Based	650	87.39 %	✖	✖
JRip	Rule Based	1045	82.63 %	✔	✖
NNGE	Rule Based	75	73.42 %	✔	✖
PART	Rule Based	89	90.67 %	✔	✔
Ridor	Rule Based	3461	81.22 %	✖	✖
ZeroR	Rule Based	10	35.17%	✖	✖
BFTree	Tree	238	81.12 %	✔	✖
FT	Tree	93	93.89 %	✔	✔
***J48**	**Tree**	**11**	**96.20 %**	✔	✔
LMT	Tree	3650	80.72 %	✔	✖
Random Forest	Tree	130	90.12 %	✔	✔
Random Tree	Tree	12	93.94 %	✔	✔
BayesNet	Bayes	34	87.89 %	✔	✔
NaiveBayes	Bayes	10	61.89 %	✔	✖

∗ The single machine normal classification method's base classifier

E. Classifier Selection

The traditional classifier selection criteria are based on classification base performance, explainable model output, and each classifier's highest accuracy. 16 traditional classifiers were tested in (Table IX), and only 6 met the classifier's standards. The remaining classifiers are discarded due to their inferior accuracy, computational complexity, and low explainability. The Single Machine classification approach is applied using this specified traditional classifier. The majority of issues identified are due to an inability to handle polynomial data types (multiclass) and a lack to address specific dataset properties. The number of algorithms supported during this research is restricted to those supported by current versions of computing frameworks such as MapReduce [19], and Spark [20]. As a result, only 4 of Hadoop's 7 available algorithms are employed, as illustrated in Table X. To ensure that the experimental comparison is significant, the commonality mapping of the selected classifiers between the Single Machine technique (traditional algorithm) and the Hadoop method is maintained as shown in Table XI.

F. Feature Selection

The selection of features is a fundamental principle that affects machine learning. Irrelevant parameters can have a negative impact on the performance of the model. The construction of an accurate prediction model needs a combination of feature selection and data cleaning techniques. The target variable for this analysis is *responsetime*, *symptom_error_code*, and *speed*. The chosen minimum threshold

TABLE X
HADOOP CLASSIFIER SELECTION FROM EXPERIMENTAL RESULTS

Hadoop Classification Algorithm	Hadoop Classifier Type	Framework	Limitation	Selected Classifier
Bayesian Network (Radoop)	Bayes	Map Reduce	None	✔
Logistics Regression (Radoop)	Functions Rule Based	Spark	Incapable of dealing with polynomial data types (multiclass)	✖
Linear Regression (Radoop)	Functions Rule Based	Spark	Incapable of dealing with polynomial data types (multiclass)	✖
Decision Tree (Radoop)	Tree	Spark	None	✔
Decision Tree (Mlib)	Tree	Spark	Not capable of handling more than two values (binomial)	✖
Random Forest (Radoop)	Tree	Spark	None	✔
Support Vector Machine (Radoop)	Functions Rule Based	Spark	Does not possess an adequate capability	✖

TABLE XI
HADOOP COMMONALITY MAPPING OF SINGLE MACHINE CLASSIFIERS

Hadoop Classification Algorithm	Hadoop Classifier Type	Single Machine Classification Algorithm	Single Machine Classifier Type	Commonality Mapping
Bayesian Network (Radoop)	Bayes	BayesNet	Bayes	**Bayes→Bayes**
Decision Tree (Radoop)	Tree	FT, J48, RF, RT	Tree	**Tree → Tree**
Random Forest (Radoop)	Tree	FT, J48, RF, RT	Tree	**Tree → Tree**

value for feature selection weighting is 0.05, which corresponds to the 95 % confidence interval shown in Table XII.

In this study, the selected variables serve as the independent variables for the classifier. Following tokenization of the data processing method, the TF-IDF vector [21] is used to extract variables such as *speed*, *down*, *qos*, *maintain*, and *service*. This keyword was chosen based on a sublanguage dictionary [22] that was constructed during this research.

G. Data Evaluation Method

The Cross-Validation method [23] is used to evaluate the data in this study. This method looks at how well a model can generalize and how well it works with new, unknown data. For example, in K-Folds Cross-Validation, the data is separated into k equal parts. The model was created and tested for a total of k iterations using the split dataset. The k components are added to the training data set throughout

TABLE XII
TOP 10 FEATURE SELECTION (WEIGHT BY INFORMATION GAIN)

Variables	Rank	Variable Number	Weightage Threshold (Min 0.05)
responsetime	1	39	0.954
symptom_error_code	2	21	0.798
speed	3	23	0.664
cause_category	4	24	0.305
down	5	1	0.290
btu_type	6	19	0.254
maintain	7	2	0.145
btu_platform	8	5	0.107
qos	9	27	0.088
servis	10	33	0.079
....	[..n]	[..n]	[..n]

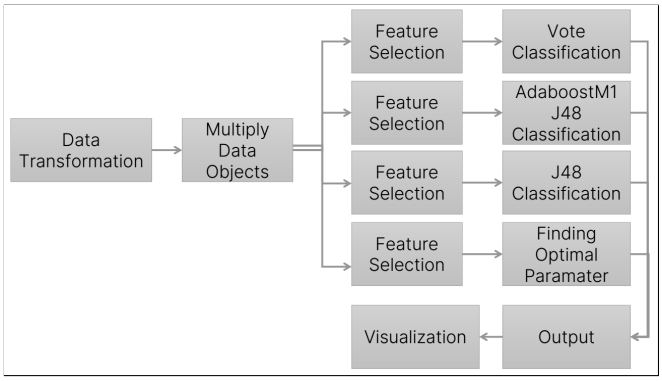


Fig. 6. Data Preprocessing and Classification Method in Single Machine

each repetition of the test dataset and the other *k-1* iterations. The Cross-Validation ratio k chosen for this study is 10. The selection of 10x Cross-Validation decreases the bias [24] since the larger the k, and the less bias is introduced into the analysis. The primary rationale is that only a small amount of data can fit into the memory of a Single Machine, increasing the chances that the Cross-Validation run would succeed and complete the computation. When there is a lot of data, the Cross-Validation runs will take longer.

H. Data Classification Method

This section contains the final transformation table for the classification process. The final process incorporates both independent and dependent factors. The variable *resolution_code* code has been chosen as the dependent variable (target variable). The dependent variable determines the outcome classification accuracy. The CTT classification approach is classified into two categories: a Single Machine and Hadoop classification. The workflows for data processing and categorization are illustrated in Figures 6 and 7.

Before starting the feature selection process, it is necessary to complete the data transformation stage. The associated technique is depicted in Figure 6, which is a basic implementation of feature selection before initiating the modelling process. Following that, the process proceeds onto the three modelling options: the Vote method, the Boosting method, and the normal method, which does not use an ensemble technique. The next stage is the same for each method, and it is the classification procedure determines the method's accuracy. Following that, the output is displayed for comparison and performance evaluation.

Figure 7 shows a similar alternative, but it needs Hadoop for the classification process. This process differs because it will use Radoop [25] components like Radoop Nest and SparkRM [25] before the feature selection process. Several algorithms, such as Naive Bayes, Decision Trees, and Random Forests, have been made to work in the Hadoop environment. The final process, which includes classification and visualization, is comparable to the Single Machine method.

V. RESULTS

A. Single Machine Classification Results

During the data processing step, local classification is performed by stratifying a small number of stratified [26]

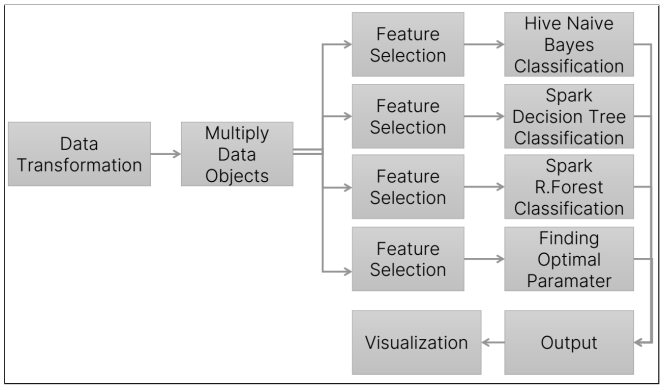


Fig. 7. Data Preprocessing and Classification Method using Hadoop

TABLE XIII
CLASSIFICATION ACCURACY PERFORMANCE BY ZONE (SINGLE MACHINE)

Zone	Stratified Sample Size	Accuracy (Vote) compose of *W-J48,W-BN,W-RF,W-RTree,W-FT,W-PART*+(10x Cross-Validation)	Accuracy (Boosting) compose of *W-J48,W-BN,W-RF,W-RTree,W-FT,W-PART*+(10x Cross-Validation)	Accuracy *W-J48* (Normal)+(10x Cross-Validation)
Bangi	3863	75.86 %	72.49 %	70.51 %
Bangsar	3110	78.04 %	74.13 %	70.15 %
B.Anggerik	3273	74.69 %	75.44 %	62.44 %
Cyberjaya	2291	77.20 %	78.39 %	68.99 %
Gombak	3986	70.02 %	78.44 %	64.21 %
Maluri	4218	77.92 %	79.04 %	**72.49 %**
T.A.Rahman	1216	73.97 %	73.71 %	69.30 %
Pandan	4245	73.55 %	80.11 %	69.24 %
Tampoi	3319	76.46 %	71.43 %	67.26 %
K.Batu	4996	70.12 %	71.12 %	68.19 %
Keramat	4116	71.12 %	**80.83 %**	70.19 %
Klang	2894	76.63 %	80.67 %	69.83 %
Puchong	6567	73.33 %	75.94 %	68.63 %
S.Jaya	6298	76.57 %	72.57 %	62.31 %
T.Petaling	3873	75.57 %	74.17 %	65.37 %
T.Dr.Ismail	8130	78.52 %	72.73 %	69.82 %
S.ABanting	4444	**82.65 %**	75.73 %	67.51 %
All Zone	4167	71.15 %	72.95 %	68.83 %
Average		**75.18** %	**75.35** %	**68.29** %

datasets into multiple zones (i.e., Bangi Bangsar and Bukit Anggerik). Each day, a particular customer's high-speed Internet penetration affects the number of CTT cases in a specific zone. Traditional classifiers (i.e., W-J48, Bayesian Network) are ranked in classification accuracy without considering ensemble approaches, the Vote method, or the Boosting method. W-J48 [27], W-BayesNet [28], W-RandomForest [13], W-PART, W-Random Tree [29], and W-FT [30] are among the classifiers incorporated in Weka's Vote [31] ensemble method. The classification accuracy is more diverse because a different algorithm runs each dataset output in the Vote operator. The final Vote classification results are made by selecting the classifier from the consolidated classifiers with the best accuracy results for the final results. The results are then summarised in Table XIII.

The Boosting approach is another ensemble method that is used. W-J48 was chosen as the algorithm to utilize with this technique. In addition, the Cross-Validation [32] method is used. According to the findings, the average classification accuracy went up from 5 % to 8 %. The last method that has been attempted is to use classification without the ensemble method. W-J48 was also chosen for this method. The best accuracy for (Shah Alam / Banting) zone attained with the Voting approach is 82.65 %. When the W-J48 classifier is utilized, the lowest accuracy at Subang Jaya is 62.31 %. In terms of how well it performs, the Boosting method is in second place, behind the Vote method. A random sample across the country (multiple zones) shows the overall classification performance with a minimum accuracy of 68.83 %.

TABLE XIV
CLASSIFICATION PERFORMANCE BY ZONE RESULTS (HADOOP)

Zone	Stratified Sample Size	Accuracy NB (Spark)	Accuracy DT (Spark)	Accuracy RF (Spark)	Accuracy NB (Spark) + (10x Cross-Validation)	Accuracy DT (Spark) + (10x Cross-Validation)	Accuracy RF (Spark) + (10x Cross-Validation)
Bangi	38634	86.26%	85.68%	87.98%	71.27%	72.36%	71.69%
Bangsar	31106	85.67%	85.22%	84.57%	78.36%	77.96%	76.24%
Bkt. Anggerik	32737	85.64%	86.36%	**91.16%**	72.13%	78.94%	78.13%
Cyber jaya	22911	90.64%	86.36%	85.96%	72.13%	82.65%	75.49%
Gombak	39863	89.83%	**89.32%**	87.74%	78.36%	83.21%	79.31%
Maluri	42188	86.44%	86.97%	86.84%	81.56%	79.65%	81.54%
T.A. Rahman	12166	87.25%	87.14%	79.45%	79.54%	79.96%	78.25%
Pandan	42452	89.23%	85.12%	80.65%	81.45%	83.52%	84.59%
Tampoi	33198	82.69%	80.65%	84.98%	83.54%	81.11%	82.17%
Kep.Batu	49967	83.47%	83.54%	88.74%	74.96%	82.65%	**84.98%**
Keramat	41163	85.63%	83.54%	81.25%	78.96%	79.99%	80.23%
Klang	28965	83.24%	79.54%	83.97%	**84.14%**	80.54%	78.45%
Puchong	65671	89.35%	85.64%	84.56%	77.54%	81.65%	82.59%
Subang Jaya	62983	**91.24%**	88.95%	89.65%	78.45%	83.96%	81.09%
T. Petaling	38737	87.14%	88.98%	82.13%	79.65%	**84.27%**	79.23%
T.Dr. Ismail	81301	85.35%	79.89%	78.45%	77.16%	83.69%	83.56%
S.A. Banting	44449	81.35%	82.17%	80.97%	81.90%	82.45%	81.97%
All Zone	38655	87.10%	82.54%	82.63%	79.65%	81.55%	80.24%
Average		**82.64**%	**84.86**%	**84.53**%	**78.37**%	**80.93**%	**79.98**%

B. Hadoop Classification Results

Only the Vote and Boosting classification methods can provide the highest level of accuracy for a Single Machine classification. The accuracy value varies by zone due to variations in the quality of data gathering and total stratified sampling. To get better accuracy, the classification process must be done in the Hadoop environment, which can handle the large dimension and high volume of the dataset [2]. There are just three categorization algorithms accessible in Hadoop: Decision Tree [33], Random Forest, and Naive Bayes (NB). The Hadoop classification result is better than the Single Machine classification method by about 10 % (Table XIV).

VI. CONCLUSION

The datasets from the telco's trouble tickets are evaluated with a Single Machine and the Hadoop approach. The Hadoop approach solves the problem of record limits while simultaneously improving classification accuracy. This study examines the essential aspects of the dataset variables and their important factors, prior fault resolution patterns,

TABLE XV
CLASSIFICATION ACCURACY (SINGLE MACHINE VS. HADOOP)

Zone	S.M Sampling Size	Hadoop Sampling Size	Single Machine Optimal Accuracy	Hadoop Optimal Accuracy
Bangi	3863	38634	75.86%(Vote)	87.98%(RF)
Bangsar	3110	31106	78.04%(Vote)	85.67%(NB)
B.Anggerik	3273	32737	75.44%(Boosting)	91.16%(RF)
Cyberjaya	2291	22911	78.39%(Boosting)	90.64%(NB)
Gombak	3986	39863	78.44%(Boosting)	89.83%(NB)
Maluri	4218	42188	79.04%(Boosting)	86.44%(DT)
T.A.Rahman	1216	12166	73.97%(Vote)	87.25%(NB)
Pandan	4245	42452	80.11%(Boosting)	89.23%(NB)
Tampoi	3319	33198	76.46%(Vote)	82.69%(RF)
KepongBatu	4996	49967	71.12%(Boosting)	83.54%(DT)
Keramat	4116	41163	80.83%(Boosting)	85.63%(NB)
Klang	2894	28945	80.67%(Boosting)	83.24%(RF)
Puchong	6567	65671	75.94%(Boosting)	89.35%(NB)
SubangJaya	6298	62983	76.57%(Vote)	**91.24%**(NB)
T.Petaling	3837	38377	75.57%(Vote)	87.14%(NB)
T.Dr.Ismail	8130	81301	78.52%(Vote)	85.35%(NB)
S.A.Banting	4444	44449	**82.65%**(Vote)	85.17%(DT)
All Zone	**4167**	**38655**	72.95%(Boosting)	87.10%(NB)
Average			**77.25** %	**87.14**%

* Vote = W-J48,W-BN,W-RF,W-RTree,W-FT,W-PART +(10x Cross-Validation) * Boosting = W-J48 +(10x Cross-Validation) * DT = Decision Tree * NB = Naive Bayes * RF = Random Forest * S.M = Single Machine

the resolution to the symptom error code, and monitoring of specific network components and threshold values that may directly impact the end-user experience when the fault happens. As a result of these findings, having the predicted resolution code for each fault can significantly increase the present fault's resolution performance. The final analytics models developed are capable of predicting future resolution codes with a maximum accuracy of 82.65 % on a Single Machine and 91.24 % using Hadoop. With the assistance of the Spark framework, advancements in data processing approaches are now capable of parallelizing the process, and the dataset size that can fit into memory for computation is also becoming more extensive without issues.

The overall improvement in Hadoop classification accuracy over the Single Machine approach is approximately 8 %. Even an increase of 8 % is considered acceptable, given the research findings will be utilized by the telco company that also supported this research. In total, there are 18 serviceable telecom zones in the study, and each dataset is based on a stratified sampling method. The data transformation in Hadoop changes the original telco dataset into an analytics series that can meet the needs of the chosen classifiers. Each day, customer trouble tickets provide insight into the telco operation's service efficiency and trend in fault trouble tickets. The researchers' objective is to develop the most accurate algorithm for classifying resolution codes. The dataset includes structured and unstructured sections to determine the classification algorithm used. When the CTT data definition is converted to a vector, specific keywords impact the classification model substantially. Additionally, the study's findings also identified a new research topic that will focus on reviewing Standard Operating Procedures (SOP) and service quality in a specific service zone on minimizing trouble tickets and fault restoration times.

REFERENCES

[1] A. Rusli, A. Suryadibrata, S. B. Nusantara, and J. C. Young, "A comparison of traditional machine learning approaches for supervised feedback classification in bahasa indonesia," *IJNMT (International Journal of New Media Technology)*, vol. 7, no. 1, p. 28–32, Jul 2020.

[2] V. Pappu and P. M. Pardalos, "High-dimensional data classification," *Clusters, Orders, and Trees: Methods and Applications*, p. 119–150, 2014.

[3] C. Shekhar Gautam—1 Akhilesh A. Waoo, "Speedup query processing in hadoop using mapreduce framework," *Data Research*, vol. 2, no. 1, p. 43, 2018.

[4] W. Dai and W. Ji, "A mapreduce implementation of c4.5 decision tree algorithm," *International Journal of Database Theory and Application*, vol. 7, no. 1, p. 49–60, Feb 2014. [Online]. Available: https://pdfs.semanticscholar.org/cc13/fde0a91f4d618e6af66b49690702906316ae.pdf

[5] F. Yuan, F. Lian, X. Xu, and Z. Ji, "Decision tree algorithm optimization research based on mapreduce," *2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, Sep 2015.

[6] J. Shah and R. Patel, "Classification techniques for disease detection using big-data," *2019 4th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT)*, Dec 2019.

[7] M. Ali Mohamed, I. M. El-henawy, and A. Salah, "Usages of spark framework with different machine learning algorithms," *Computational Intelligence and Neuroscience*, vol. 2021, p. 1–7, Jul 2021.

[8] S. Bagui, K. Devulapalli, and S. John, "Mapreduce implementation of a multinomial and mixed naive bayes classifier," *International Journal of Intelligent Information Technologies*, vol. 16, no. 2, p. 1–23, Apr 2020.

[9] S. Du and J. Li, "Parallel processing of improved knn text classification algorithm based on hadoop," *2019 7th International Conference on Information, Communication and Networks (ICICN)*, Apr 2019.

[10] S. Huang, M. Huang, and Y. Lyu, "An improved knn-based slope stability prediction model," *Advances in Civil Engineering*, vol. 2020, p. 1–16, Jul 2020.

[11] Z. Chen and J. Cheng, "A parallel softmax classification algorithm based on mapreduce," *2018 13th International Conference on Computer Science and Education (ICCSE)*, Aug 2018.

[12] F. J. M. Velasco, "A bayesian network approach to diagnosing the root cause of failure from trouble tickets," *Artificial Intelligence Research*, vol. 1, no. 2, May 2012.

[13] J. S. Tan, C. K. Ho, A. H. L. Lim, and M. R. B. M. Ramly, "Predicting network faults using random forest and c5.0," *International Journal of Engineering and Technology*, vol. 7, no. 2.14, p. 93, Jun 2019.

[14] D. I. Tholath and F. C. S.J., "Customer journey maps for demographic online customer profiles," *International Journal of Virtual Communities and Social Networking*, vol. 8, no. 1, p. 1–18, Jan 2016.

[15] I. K. Raharjana, I. Ibadillah, P. Purbandini, and E. Hariyanti, "Incident and service request management for academic information system based on cobit," *Proceeding of the Electrical Engineering Computer Science and Informatics*, vol. 5, no. 5, Nov 2018.

[16] S. Velliangiri, P. Karthikeyan, and V. Vinoth Kumar, "Detection of distributed denial of service attack in cloud computing using the optimization-based deep networks," *Journal of Experimental and Theoretical Artificial Intelligence*, p. 1–20, Apr 2020.

[17] A. Medem, M.-I. Akodjenou, and R. Teixeira, "Troubleminer: Mining network trouble tickets," *2009 IFIP/IEEE International Symposium on Integrated Network Management-Workshops*, 2009.

[18] P. V. Vuletić, J. J. Vuleta-Radoičić, and D. Kalogeras, "Federated trouble ticket system for service management support in loosely coupled multi-domain environments," *International Journal of Network Management*, vol. 25, no. 2, p. 95–112, Jan 2015.

[19] S. Jeon, H. Chung, W. Choi, H. Shin, J. Chun, J. T. Kim, and Y. Nah, "Mapreduce tuning to improve distributed machine learning performance," *2018 IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, 2018.

[20] G. Gousios, "Big data software analytics with apache spark," in *Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings*, ser. ICSE '18. New York, NY, USA: ACM, 2018, pp. 542–543. [Online]. Available: http://doi.acm.org/10.1145/3183440.3183458

[21] L. H. Patil and M. Atique, "A novel feature selection based on information gain using wordnet," in *2013 Science and Information Conference*, Oct 2013, pp. 625–629.

[22] S. Symonenko, S. Rowe, and E. D. Liddy, "Illuminating trouble tickets with sublanguage theory," *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers on XX - NAACL 06*, 2006.

[23] E. Allibhai, "Holdout vs. cross-validation in machine learning," Oct 2018. [Online]. Available: https://medium.com/@eijaz/holdout-vs-cross-validation-in-machine-learning-7637112d3f8f

[24] A. M. Molinaro, R. Simon, and R. M. Pfeiffer, "Prediction error estimation: a comparison of resampling methods," *Bioinformatics*, vol. 21, no. 15, p. 3301–3307, May 2005.

[25] A. H. Ali and M. Z. Abdullah, "A parallel grid optimization of svm hyperparameter for big data classification using spark radoop," *Karbala International Journal of Modern Science*, vol. 6, no. 1, Mar 2020.

[26] R. Levin and Y. Kanza, "Stratified-sampling over social networks using mapreduce," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '14. New York, NY, USA: ACM, 2014, pp. 863–874. [Online]. Available: http://doi.acm.org/10.1145/2588555.2588577

[27] N. anaN and V. thri, "Performance and classification evaluation of j48 algorithm and kendall's based j48 algorithm (knj48)," *International Journal of Computer Trends and Technology*, vol. 59, no. 2, p. 73–80, May 2018.

[28] S. Taheri, M. Mammadov, and A. M. Bagirov, "Improving naive bayes classifier using conditional probabilities," in *Proceedings of the Ninth Australasian Data Mining Conference - Volume 121*, ser. AusDM '11. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2011, pp. 63–68. [Online]. Available: http://dl.acm.org/citation.cfm?id=2483628.2483637

[29] A. Joshuva. and V. Sugumaran., "A data driven approach for condition monitoring of wind turbine blade using vibration signals through best-first tree algorithm and functional trees algorithm: A comparative study," *ISA Transactions*, vol. 67, p. 160–172, 2017.

[30] J. Gama, "Functional trees," *Machine Learning*, vol. 55, no. 3, p. 219–250, 2004.

[31] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009. [Online]. Available: http://doi.acm.org/10.1145/1656274.1656278

[32] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 36, no. 2, p. 111–133, 1974.

[33] W. Du and Z. Zhan, "Building decision tree classifier on private data," in *Proceedings of the IEEE International Conference on Privacy, Security and Data Mining - Volume 14*, ser. CRPIT '14. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2002, pp. 1–8. [Online]. Available: http://dl.acm.org/citation.cfm?id=850782.850784