# Automatic Ticket Assignment using Machine Learning and Deep Learning Techniques

MSc Research Project
Data Analytics (MSCDA-B)

Sejal Shah
Student ID: x18196292

School of Computing
National College of Ireland

Supervisor: Dr. Muhammad Iqbal

**National College of Ireland**
**Project Submission Sheet**
**School of Computing**

| | |
|---|---|
| **Student Name:** | Sejal Shah |
| **Student ID:** | x18196292 |
| **Programme:** | Data Analytics (MSCDA-B) |
| **Year:** | 2020 |
| **Module:** | MSc Research Project |
| **Supervisor:** | Dr. Muhammad Iqbal |
| **Submission Due Date:** | 17/08/2020 |
| **Project Title:** | Automatic Ticket Assignment using Machine Learning and Deep Learning Techniques |
| **Word Count:** | 5565 |
| **Page Count:** | 24 |

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

**ALL** internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

| | |
|---|---|
| **Signature:** | |
| **Date:** | 17th August 2020 |

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:**

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies). | ☐ |
| **Attach a Moodle submission receipt of the online project submission**, to each project (including multiple copies). | ☐ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | ☐ |

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# Automatic Ticket Assignment using Machine Learning and Deep Learning Techniques

Sejal Shah
x18196292

**Abstract**

One of the important factors of Information Technology Service Industries is to provide user satisfaction and better customer service while saving cost. To achieve the said objective industries often opt for Ticketing system to handle unforeseen service interrupting events. These unforeseen events are also termed as 'incident'. Incident management processes provide quick fixes or workarounds to solve the incident and ensure little or no business impact. Traditionally, the incidents have been manually assigned to the concern teams which has sometimes resulted in human errors, resource consumption, higher response and resolution times and ultimately poor customer service.

This research project focuses on automated analysis and allocation of 'incidents' to appropriate teams using Natural Language Processing and Machine Learning techniques. The data set used in this research contains information in multiple languages and the data has been translated into English for better understanding of the data indicating that the solution to this problem could be obtained using a combination of various data pre-processing, Language Translation and Classification techniques. High class imbalance has been identified in the data set and it has been solved using combination of under-sampling and over-sampling techniques. To classify the incident tickets multiple algorithm namely K-Nearest Neighbours(KNN), Support Vector Machine(SVM), Random Forest(RF),Artificial Neural Network(ANN),Bidirectional Long Short Term Memory(BLSTM) have been trained and hyper-parameter tuning has been performed using different cross validation techniques. Furthermore, confidence interval and k-fold cross validation technique have been used to validate the scores. The ANN model has obtained 93.6% and 95.1% confidence interval at 95% alpha while accuracy score is 95% on the test data.

***Keywords-Ticketing system, Natural Language Processing, KNN, SVM, RF, ANN, BLSTM***

# 1 Introduction

## 1.1 Automatic Ticket Assignment Domain Overview

Nowadays, organizations heavily rely on IT resources and services for efficient internal and external operations making IT service desks an important factor. Primarily, IT service desks focus on recording user queries and provide first level solution to queries. If the queries cannot be solved by first level solution, the queries get assigned to subject matter expert teams which solve the queries. Traditionally, assignment of tickets has been performed manually by organizations which has sometimes generated human error in form of assignment of ticket to wrong team resulting in high resource consumption, higher query resolution time ultimately deteriorating customer service and satisfaction.

In the support process, incoming incidents are analysed and assessed by organization's support teams to fulfil the request. In many organizations, better allocation and effective usage of the valuable support resources will directly result in substantial cost savings. The ticket assignment system allows automation of daily IT tasks such as assigning tickets to respective teams reducing resource costs. Moreover, the IT service industries can reduce resolving time and increase customer satisfaction by introducing automation to solve incidents which are faced on day-to-day basis. Additionally, it helps in the overall development of an IT industry performance. (Al-Hawari and Barham; 2019) (Grosser; 2015)

Figure 1, Depicts a use case diagram of IT service industry which consists of customer, service agent and admin. With an increase in a large number of tickets in day-to-day life which is mostly unstructured data, text classification plays a vital role in handling data using Natural Language processing to find the ticket issue type and assigning it to respective support team.(Koushik et al.; 2019)

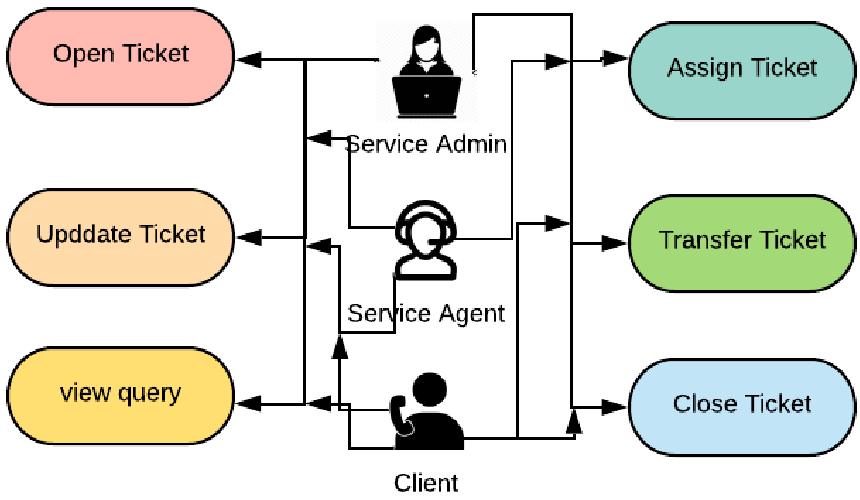



Figure 1: Use case diagram of IT service industry

## 1.2 Background and Motivation

IT management and services are vital entities which are responsible for innovation, strengthening and increasing value, boosting productivity and giving technical services to end-users. [1]. Although, with such profound duty front end IT conversation are considered as lengthy and tedious in terms of solving small queries such as password reset, software

[1]https://towardsdatascience.com/predict-it-support-tickets-with-machine-learning-and-nlp-a87ee1cb

configuration etc. Even though the service requests are genuine and crucial, support institutes often find it difficult to provide efficient services. The manually driven task lead to bottlenecks, particularly at large institutes in form of higher resolution times and resource exhaustion. Generally, the incidents are non structured in nature and complex to be analyzed either manually or by using simple data handling techniques. Thus, Machine Learning and Deep Learning techniques can be proven useful to solve the bottlenecks by combining the traditional system with the new ticketing system.

### 1.3 Research Question and Objectives

*"How efficiently natural language processing and classification techniques from machine learning and deep learning can help in automating the process of ticket assignment?"*

This study focuses on data pre-processing and Natural Language Processing techniques to analyze incident data(ticket) and assign it to the concern teams for resolution of incidents. Automating currently used manual methods of ticket assignment. This research features multiple algorithms from machine learning and deep learning for classifying the tickets.

The objectives of this research have been listed below:

- Collecting and pre-processing the data.
- Exploration of data.
- Pre-process the data based on findings from data exploration.
- Optimize the data using feature engineering and prepare for model training.
- Implement classification algorithms such as Support Vector Machine (SVM), K-Nearest Neighbour(KNN), Random Forest Classifier, Artificial Neural Network (ANN) and Bidirectional Long Short Term Memory(BLSTM).
- Comparison and evaluation of the above-mentioned algorithms using hyper-parameter tuning and K-fold cross validation.

The paper has been organised as Section 2: related work performed in this field, Section 3: describes the methodology used in this paper, Section 4: demonstrates the machine learning and deep learning models implementation, Section 5: explains the evaluation and lastly, Section 6: conclusion.

## 2 Related Work

This section attempts to cover recent studies conducted in the field of Automatic Ticket Assignment and Natural Language Processing (NLP) for classification. This section has been divided based on three major aspects namely:

- The Architecture of models.
- Data Pre-processing techniques.

- Text classification Methods.

All these aspects have played an important role in giving a thorough understanding of previous work.

## 2.1 The architecture used for the models

Jing (2019) has implemented a LSTM network consisting of 16 hidden layers using advanced python library PyTorch with word embedding dimension of 32 and learning rate of 0.01 in Adam optimizer(Sachan et al.; 2019). A new model which is a combination of self-attention model and basic LSTM model has been introduced which has successfully improved long sentences handling for sentiment analysis.

Zhou et al. (2017) have used a single embedding layer network with two composite frameworks on loop. Finally, a fully connected output layer for distributed representation has been added. The architecture consist of a convolution layer and K-max pooling layer with a non-linear activation function. Additionally, authors have also compared the three most famous models namely Random Forest, Gradient Boosting and Logistic Regression on the real-world labelled data set in which Random Forest Classifier outperforms the other two models for the ticket resolution and features importance evaluation.

A study by Seongmoon and Wenlong (2009) includes a two-stage classification process for assigning ticket issue to support unit to detect the similar category ticket and subcategory ticket. In the results, SVM has performed better by achieving an accuracy score of 90% in comparison to Naïve Bayes and KNN. Similarly, a ticket classifier system has been built by authors K. S. Manjunatha and Guru (2019), in which algorithms namely Multinomial Naïve Bayes, KNN(k=5) and SVM with accuracy scores of 69%, 67% and 87% respectively. Moreover, K-cross validation (k =10) has been used for measuring the average performance of each model and SVM has outperformed all the models. In contrast,(Liang and Zhang; 2016) have used BLSTM model and then the concatenation of the last layer with the hidden state of multi-layered BLSTM to achieve the text and then softmax layer has been added for classification. Alternatively, an advanced approach adopted by (Molino et al.; 2018) for the ticket assignment using neural network for 100 epoch in CNN model using dimension 0f 256-word embeddings and 4 convolution layers with size 2, 3 and 4 having filters 512 in each. Categorical features have been embedded using 256 embedding dimensions. Batch normalization layer for reducing overfitting, Adam optimization function with learning rate of 0.000025 and batch size of 256 have been used for training the model.

## 2.2 Data preprocessing Techniques

The study by Seongmoon and Wenlong (2009) has explained pre-processing by purifying the tickets from HTML tags and the numerical value and also, by removing unwanted data. More on, with the help of bag of words approach text data has been converted into a numerical format. Additionally, the data sparsity problem has been resolved using morphological analysis of words by choosing the appropriate root, suffixes and prefixes of words. Also, TF-IDF vectors have been used to improve the accuracy score. In paper (Liang and Zhang; 2016) authors have categorized text using BLSTM approach. Also, to avoid overfitting of data at the time of training batch normalization and dropout layers have been used. Besides, authors have used word2vec vectors of 300 dimensions to obtain bag of words structure, also 10 fold cross-validation has been applied and mean accuracy has been documented for evaluation.

Data pre-processing is the most basic step for the text data. To remove word segmentation result which contains punctuation symbols regular expression has been used. Also, stop words which create redundancy and increase ambiguity while training have been removed. More on, LSTM model has been implemented with 64 LSTM units and learning rate of 0.001, achieving an accuracy score of 97%. Besides, to evaluate the classification scores precision, recall and F1 score have been used as evaluation metrics. The research also concludes that the LSTM model performs better than the parametric model (Liang et al.; 2019). Unlike in study by (Silva et al.; 2018) and (Beneker and Gips; 2017) text tokenization has been used as the first step for pre-processing which separates the variously described incidents in the words, followed by removing stop word and stemming to reduce the words to their root. Secondly, the TF-IDF feature vector has been created for the selection of relevant and irrelevant keywords.

Authors in research paper (Mandal et al.; 2018) have concluded that using TF-IDF representation can result in an increased accuracy of the machine learning models by 3 to 4 per cent. Also, N-grams analysis has been proven useful for better understanding of the model. Furthermore, for evaluating the deep neural network performance 10-fold cross-validation and experiments using word embedding and pre-trained word vectors have been performed. However, researchers suggest that MLP and SVM can be good for practical application but in a large dataset LSTM- glove should be considered as the best choice.

A study by (Khramov; 2019) has obtained better performance by analyzing 200000 tickets with 260 features. In the data pre-processing stage factors such as URL's, emails, punctuation word etc have been removed from the text data due to their negative effects on the model performance. Also, in the training of the algorithm, text data has been transformed into a numerical format for reducing the training time. Additionally, the tri-gram feature has been also applied to extract most frequent combination of three words.

## 2.3 Text classification techniques

A combination of Asymmetric Convolutional Neural Network(ACNN) and Bidirectional Long Short Term Memory(BLSTM) have been implemented by authors(Liang and Zhang; 2016). On the other hand, research by (Silva et al.; 2018) which focused on Machine learning techniques in incident categorization automation features SVM and KNN for ticket categorization in which SVM has achieved an overall accuracy score of 89% and KNN has achieved 82% accuracy score. Also, in research by Pradhan et al. (2017), introduce text classification for category identification where SVM outperformed naïve bayes classifier. However, complexity of naïve bayes has been considered better than SVM classifier. Another research (Semberecki and MacIejewski; 2017) in which comparison between BLSTM and tree-LSTM has been carried out, where authors concluded that one of drawback with BLSTM that performance gets reduced with increasing class count, whereas word2vector approach showed better results.

Natural language processing uses neural network models which have been discriminatively trained, in the research (Yogatama et al.; 2017). This study features a comparison between generative LSTM and discriminative LSTM models. in generative LSTM labeled document has been used for label semantic space to train the model and discriminative

LSTM model where labeled data has been used for learning of the models for text classification in terms of asymptotic error and complexity. On the contrary, Khramov (2019) have proposed research in which comparison between the machine learning model such as SVM, Naïve Bayes and Random Forest have been performed. Also, grid search method has been implemented to obtain the best possible hyper parameters. SVM model has performed well with an accuracy of 84% using Boolean vector representation.

Wang et al. (2019) have combined Convolutional Neural network which has been useful for extraction of features while Recurrent Neural Network has been useful for Chinese and English text categorization. A new algorithm Convolutional Recurrent Neural network (CRNN) which outperforms both models by predicting an accuracy score of 93.38% for English datasets and 98.45%for Chinese dataset has been proposed. Similarly, in a research study conducted by Li et al. (2018) contains combination of Bi-LSTM and CNN model to introduce a model named BLSTM-CNN. Also, authors have concluded that deep learning model outperforms the machine learning models by predicting an accuracy of 96.45%, deep learning models give better results in terms of Loss, accuracy and F1 score than traditional models. A notable finding has been observed that a BLSTM-CNN based model can store features more precisely with less noise. Moreover, this model can capture context and text semantics more correctly to improve the accuracy of text categorization. Lee and Dernoncourt (2016) have attempted a similar approach in which RNN and CNN models have been combined for short text sequential categorization, it has also increased prediction quality and performance of the text classification model.

Table 1: Classification studies in Text classification

| **Author** | **Models used** | **Best Model** | **Accuracy** |
|---|---|---|---|
| (Semberecki and MacIejewski; 2017) | LSTM,CNN | LSTM | 82% |
| (Khramov; 2019) | SVM, NB, RF | SVM | 84.1% |
| (Li et al.; 2018) | SVM, CNN,Bi-LSTM-CNN | Bi-LSTM-CNN | 96.45% |
| (K. S. Manjunatha and Guru; 2019) | MNB,SVM,KNN | SVM | 87% |
| (Zhou et al.; 2016) | BLSTM,BLSTM-2DPooling, BLSTM-2DCNN | BLSTM-2DCNN | 89.5% |
| (Zhou et al.; 2017) | BLSTM,LSTM,C-LSTM | C-LSTM | 87.8% |

From above literature review it has been observed that the use of TF-IDF vectorizers technique in the pre-processing of the text data tremendously helps in traing of the model. Also, most studies have focused on machine learning techniques as compared to NLP and deep learning techniques. The use of ANN has been minimum in previous studies while combination of LSTM networks with networks such as RNN and CNN have been studied. The study of KNN from Machine Learning and ANN,BLSTM from deep learning techniques has not been performed on this type of data set. Table 1 features most notable related work and used techniques in the domain of automatic ticket assignment.

# 3 Methodology

This proposed research study aims at finding the best multi-class classification model for automatic ticket assignment. The study mainly focuses on finding a solution which can analyse incident data and assign it to the concern team for saving resources and improving customer service. This research has a methodology based on Knowledge Discovery in Data mining (KDD) approach for finding the optimal solution. Figure 2 explains the methodology for implementation of this research.

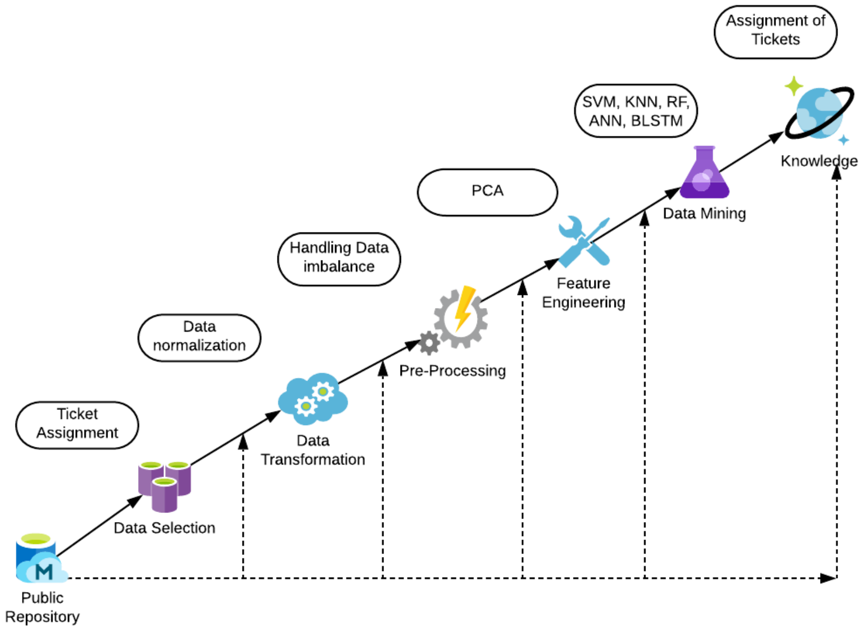



Figure 2: Automatic Ticket Assignment Methodology

In the proposed research, the data has features like Short Description, Description, caller and Assignment Group which have to be analysed thoroughly. To achieve this, Automatic Ticket Assignment methodology has separate stage for each major factor in development of data mining application. Automatic Ticket Assignment methodology has independent layers for every major features.

**The flow of Automatic Ticket Assignment Methodology has been explained below:**

For this proposed research the dataset has been taken from kaggle public repository [2]. Data pre-processing is one of the important aspect which contributed in the final results of model predictions. The gathered data consist of useful information with unwanted noise. Data pre-processing is needed at this stage to remove the unwanted noisy data. Techniques like one-hot encoding and attribute construction are used to change the structure of the data which can be fed into a machine learning algorithm.

Algorithms such as Random Forest, K-Nearest Neighbors(KNN), Support Vector Machine(SVM), Artificial Neural Network(ANN) and Bidirectional Long Short Term

[2] https://www.kaggle.com/aviskumar/automatic-ticket-assignment-using-nlp

Memory Networks(BLSTM) have been applied in Data Mining stage to obtain the knowledge from dataset.

## 3.1 Data Gathering

Dataset has been collected from public dataset repository kaggle, after that the data set has been read and explored using Pandas library from python. Some inconsistencies and missing values have been detected in the data which have been handled in the data pre-processing stage.

## 3.2 Data Cleaning and pre-processing Process Flow

The data set also contained Mojibake(garbled text) and text in other scriptures than English which had to be translated in English for better understanding of the data. Also, the data contained a lot of noise such as blank spaces, numbers, email addresses and void text which could not help in model training which has been removed using pattern matching and regular expressions.

Later, the data has been normalized using Lemmatization technique which assisted in bringing the root words. Furthermore, few columns have been added in the dataset while pre-processing as basis for performing Exploratory Data Analysis(EDA). The data cleaning process flow has been shown in figure 3.

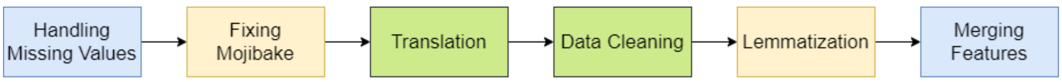



Figure 3: Data Cleaning Process Flow

## 3.3 Exploratory Data Analysis

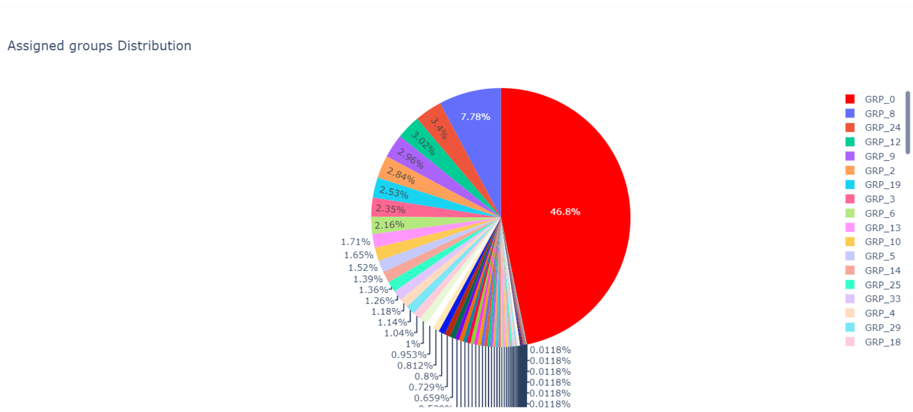



Figure 4: Pie Chart of target column distribution

From Figure 4, it is clear that Group 0 has been assigned maximum number of incidents followed by Group 8 whereas there are some groups which have been allotted only one ticket from the whole dataset.

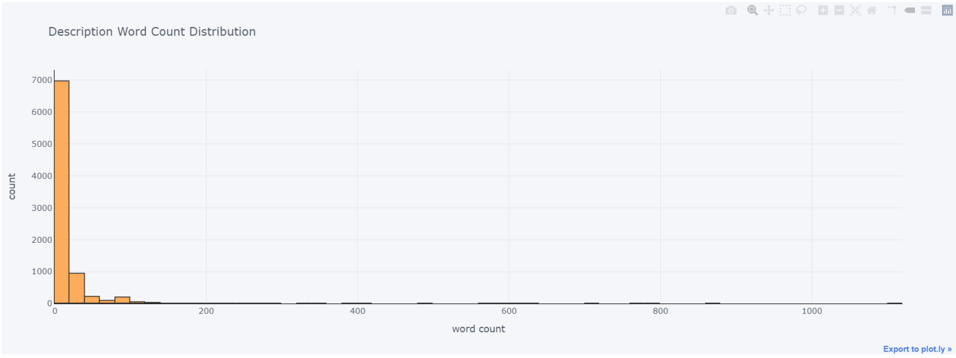



Figure 5: Word count of incident description

Above Figure 5 shows word count of ticket distribution, which is indicating that text length of 0-49 has maximum no. of entries whereas more than 600 has very few entries and data entries with 0-19 words are in a great number followed by 10-30 words entry and with greater than 100 words are very rare.

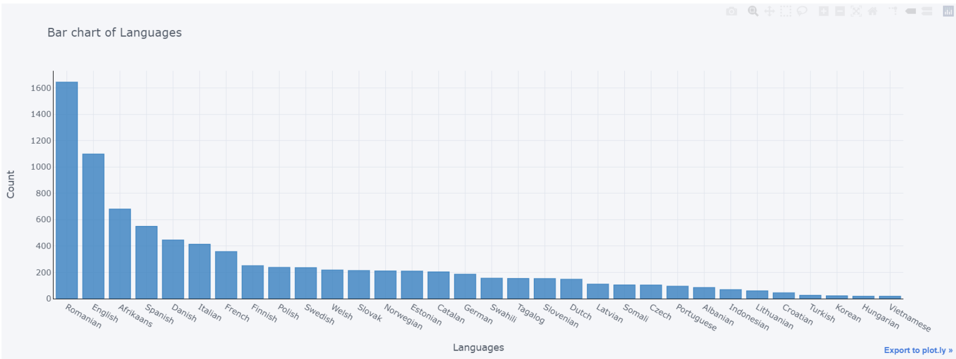



Figure 6: Distribution of original Languages

With reference to the above Figure 6, it can observe that there are 32 languages present in the data set. Romanian and English language have been majorly used.

### 3.3.1 N-Grams and Word Clouds

The essential concepts in text mining is n-grams, which are a group of co-occurring or continuous sequence of n items from a sequence of huge text or sentence. The item here might be words, letters, and syllables. 1-gram is additionally called as unigrams are the unique words present within the sentence. Bigram(2-gram) is that the combination of two words. Trigram(3-gram) is 3 words then on.

**N-Gram analysis**

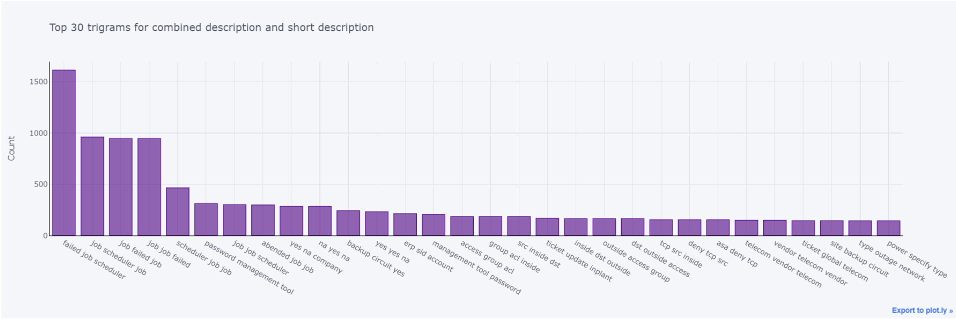



Figure 7: Top 30 Tri-grams from training data

Observations from N-Grams:

- The dataset has most number of issues reported about Failed job scheduler, job failed, password management tool, password reset, abended job etc.

**Word Clouds** .

The collection of words portrayed in different sizes that are used in the model is called word cloud. The more frequent words in the data appears bolder and bigger in the plot which shows its importance in the data. It is also known as text clouds or tag clouds. To transfer the word data from blog posts to databases, it is one of the popular way to extract the important part from a textual data.


Figure 8: Word cloud created using whole training data

Observations from Word Clouds depicted in Figure 8 and Figure 9:

- The dataset has most number of issues reported about Failed job scheduler, job failed, password reset, user.

- The analysis of 'GRP_0' depicts that most of the tickets assigned to 'GRP_0' are related to password issues, outlook, account locked, erp tool etc.

- The need for automating handling of failed job scheduler and password reset can be confirmed.

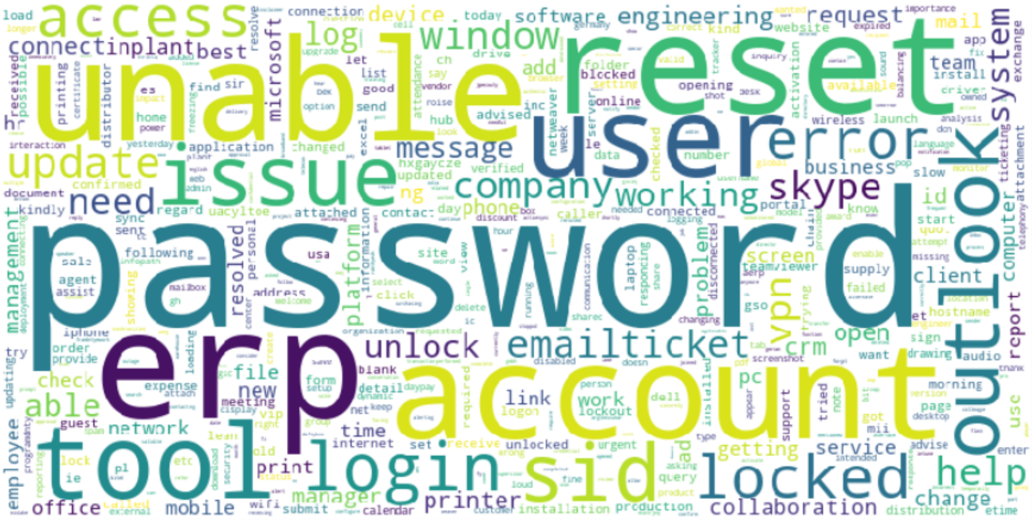
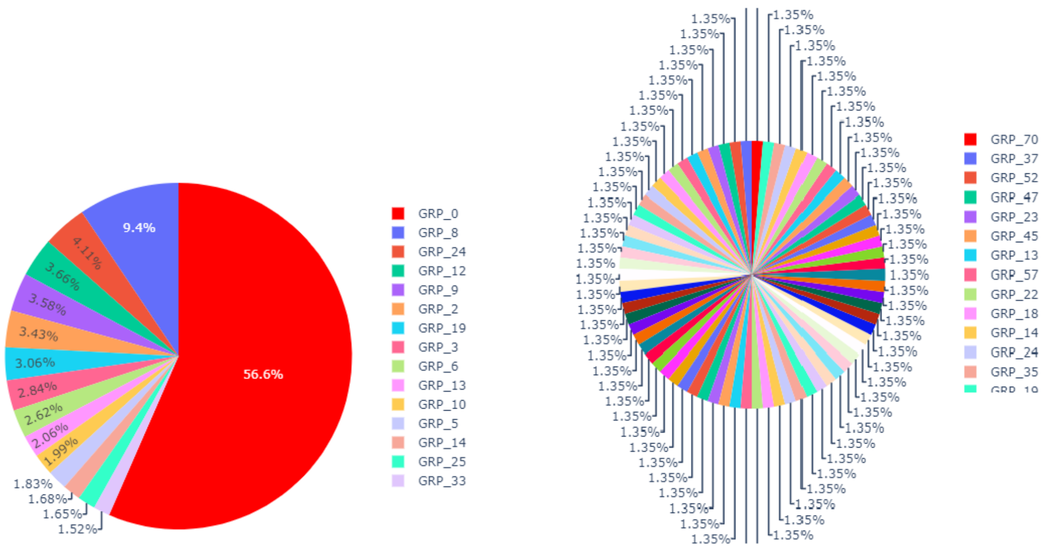



Figure 9: Word cloud for tickets assigned to 'GRP_0'(majority group)

## 3.4 Data preparation and Feature Engineering

In Exploratory Data Analysis (EDA) the data has been explored thoroughly and important insights have been extracted from the data. In the EDA, it was discovered that the data set is severely imbalanced towards 'GRP_0'.Figure 10(a).

### 3.4.1 Dataset imbalance handling




(a) Distribution before fixing class imbalance

(b) Distribution after fixing class imbalance

Figure 10: Comparison of target class

The target class imbalance detected in EDA has been fixed using a combination of Under-sampling and Over-sampling technique which balanced the target class distribution.

After fixing the class imbalance the data set has been observed to have a balanced class distribution which will help in increasing the performance of models. Figure 10 explains the difference between dataset before fixing the class imbalance and after fixing the class imbalance.

Later, the data has been split into train and test sets in train(80):test(20) ratio. Also, the TF-IDF vectorizer has been applied on the data. PCA which covers 95% variance to reduce the dimensions of the dataset has been applied which has reduced the number of dimensions from '10736' to '3468'. Later, the data has been passed to the next stage.

### 3.5 Data Modelling and Evaluation

The vectorized data has been used to train the models. In the proposed research various Machine Learning models such as SVM, KNN, Random Forest and Deep Neural Networks such as ANN, BLSTM have been applied and has been explained in Section 4

The models have been evaluated using various evaluation matrices such as Precision, Recall, F1-score, Confidence Interval and Accuracy detailed in section 5

## 4 Implementation

### 4.1 Support Vector Machine

Support vector machine classifier is a discriminant based method. In which the classifier only concerns for small instance to the border or discriminator and avoids the other examples. Therefore, classifier complexity is based on only the number of support vector but dataset size. A kernel-based method is termed as a problem with convex optimization by finding the best possible solution (Seongmoon and Wenlong; 2009).

Figure 11 shows the set of parameters chosen for SVM model such as 'rbf' kernal for degree 3.

```
Used Classifier: Pipeline(memory=None,
          steps=[('model',
                  SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None,
                      coef0=0.0, decision_function_shape='ovr', degree=3,
                      gamma='scale', kernel='rbf', max_iter=-1,
                      probability=False, random_state=None, shrinking=True,
                      tol=0.001, verbose=False))],
          verbose=False)
```

Figure 11: Support Vector Machine model summary

### 4.2 K-Nearest Neighbour

K- Nearest neighbour is used to find the k nearest similar matches in the training dataset and then predict with the help of closest matches label. The proximity or closeness between the samples of data describes the neighbourhood. There are various methods to calculate the closeness or distance between these data sample based on the problem. Most widely used is Euclidean distance also known as straight line distance. The neighbours in the same group shows similar behaviour and characteristics, also it the main idea behind the simple supervised learning classification algorithm. Now, for K – Nearest Neighbours the value of K in unknown data for classification and assigning it to the group existing mainly in those k neighbours (Silva et al.; 2018).

The default K-Nearest Neighbour function parameters chosen in this research have been shown in Figure 12.

```
Used Classifier: Pipeline(memory=None,
         steps=[('model',
                 KNeighborsClassifier(algorithm='auto', leaf_size=30,
                                      metric='minkowski', metric_params=None,
                                      n_jobs=None, n_neighbors=3, p=2,
                                      weights='uniform'))],
         verbose=False)
```

Figure 12: KNN model summary

## 4.3 Random Forest Classifier

Random Forest Classifier is an ensemble tree-based learning algorithm. Basically, it's a group of decision trees selecting subset of train set randomly. These decision tree classifiers collection are often called the forest.

The individual decision trees are generated using an attribute selection indicator like information gain, gain ratio, and Gini index for every attribute. Every tree is made on an independently random sample. Considering classification problems, It collects the voting from different decision trees to classify the ultimate class of the test set. In this research , Random Forest Classifier has been applied both default and hyper parameters.

The parameters selection has mentioned in Figure 13 with minimum sample split as 2,minimum sample leaf = 1 and n estimators as 100.

```
Used Classifier: Pipeline(memory=None,
         steps=[('model',
                 RandomForestClassifier(bootstrap=True, ccp_alpha=0.0,
                                        class_weight=None, criterion='gini',
                                        max_depth=None, max_features='auto',
                                        max_leaf_nodes=None, max_samples=None,
                                        min_impurity_decrease=0.0,
                                        min_impurity_split=None,
                                        min_samples_leaf=1, min_samples_split=2,
                                        min_weight_fraction_leaf=0.0,
                                        n_estimators=100, n_jobs=None,
                                        oob_score=False, random_state=None,
                                        verbose=0, warm_start=False))],
         verbose=False)
```

Figure 13: Random Forest Classifier model summary

## 4.4 Artificial Neural Network

The nervous system is a combination of trillions of neurons related to each other and bypass information to one another to make decisions. ANN's have been developed based on this theory.

It all starts with assigning some random weights to the hidden layers as an initial guess and output is calculated as compared to preferred output. The degree of correctness of output is decided using loss capabilities like Cross-Entropy for type-related problems.

Assigning weights and calculating outputs is called "Forward propagation" and the changes made to weights are called "Learning". Gradient Descent is one of the most popular ways of changing weights. Gradient descent is used to calculate the advantageous path or negative course of our output and corrected weights are back propagated into network usage of chain rule of derivatives. This manner is repeated until a certain percentage of accuracy is achieved.

Figure 14 shows the description about the model parameters such as dense layers and neurons used.

Model: "sequential"

| Layer (type) | Output Shape | Param # |
|---|---|---|
| dense (Dense) | (None, 1024) | 3552256 |
| dropout (Dropout) | (None, 1024) | 0 |
| dense_1 (Dense) | (None, 1024) | 1049600 |
| dropout_1 (Dropout) | (None, 1024) | 0 |
| batch_normalization (BatchNo | (None, 1024) | 4096 |
| dense_2 (Dense) | (None, 1024) | 1049600 |
| dropout_2 (Dropout) | (None, 1024) | 0 |
| batch_normalization_1 (Batch | (None, 1024) | 4096 |
| dense_3 (Dense) | (None, 1024) | 1049600 |
| dropout_3 (Dropout) | (None, 1024) | 0 |
| batch_normalization_2 (Batch | (None, 1024) | 4096 |
| dense_4 (Dense) | (None, 1024) | 1049600 |
| dropout_4 (Dropout) | (None, 1024) | 0 |
| batch_normalization_3 (Batch | (None, 1024) | 4096 |
| dense_5 (Dense) | (None, 128) | 131200 |
| dropout_5 (Dropout) | (None, 128) | 0 |
| batch_normalization_4 (Batch | (None, 128) | 512 |
| dense_6 (Dense) | (None, 74) | 9546 |

Total params: 7,908,298
Trainable params: 7,899,850
Non-trainable params: 8,448

Figure 14: Artificial Neural Network Summary

## 4.5 Bi-directional Long short term memory(BLSTM)

The bidirectional will compute the inputs in two ways. Unlike LSTM, the bidirectional model will compute the input from past to future and vice versa. The LSTM saves information as it executes backwards.

As shown in Figure 15 3,555,130 number of parameters have been produced by LSTM model for 5 dense layer and (256,128,64,32,16) set of neurons.

| Layer (type) | Output Shape | Param # |
|---|---|---|
| embedding_4 (Embedding) | (None, 850, 200) | 2164000 |
| bidirectional_4 (Bidirection | (None, 850, 256) | 336896 |
| dense_20 (Dense) | (None, 850, 128) | 32896 |
| dense_21 (Dense) | (None, 850, 64) | 8256 |
| dense_22 (Dense) | (None, 850, 32) | 2080 |
| dense_23 (Dense) | (None, 850, 16) | 528 |
| flatten_4 (Flatten) | (None, 13600) | 0 |
| dense_24 (Dense) | (None, 74) | 1006474 |

Total params: 3,551,130
Trainable params: 3,551,130
Non-trainable params: 0

Figure 15: Bidirectional Long Short Term Memory Network Summary

# 5 Evaluation

In this research, the data set has been found to have class imbalance. Though the class imbalance has been fixed, the model evaluation is based on not only accuracy score but also class wise Precision, Recall and F-1 scores. Furthermore, confidence interval and hyper parameter tuning with k-fold cross validation has been performed to validate the performance of classification models. Some classifier models have been trained using GridSearchCV while some classifier models have been trained using RandomizedSearchCV to maintain the trade off between training time and performance. This section describes the various experiments performed for evaluating the reliability and accuracy of the models.

## 5.1 Experiment with K-Nearest Neighbours

K-Nearest Neighbours classifier has been trained and tested on the data set with handled class imbalance. Initially, the model has been trained using default parameters and later, an attempt to obtain the best hyper parameters using GridSearchCV with K-fold Cross validation (10 folds) has been made. GridSearchCV brute forces the given parameter grid to obtain the best set of hyper parameters and guarantees to find best set of hyper parameters which improve the performance of model to a certain extent. In this case, an overall improvement of 3% in testing accuracy has been recorded after hyper parameter tuning and K-Fold cross validation which can be observed in Table 2. With default parameters, the training and testing accuracies came up to be around 96% and 93% respectively. Whereas, after tuning the parameters the training accuracy increased by 2% and testing accuracy increased by 3%.

Table 2: K-Nearest Neighbour Classifier Experiments

| **Experiment** | **Train Accuracy (%)** | **Test Accuracy(%)** |
|---|---|---|
| Default Parameters | 96% | 93% |
| GridSearchCV with K-folds(10 folds) | 98% | 96% |

## 5.2 Experiment with Random Forest classifier

Random Forest Classifier has been trained using default parameters as well as parameters obtained using RandomizedSearchCV with K-fold cross validation. An attempt to obtain the best set of hyper parameters using RandomizedSearchCV with K-Cross fold validation has been made. Though, RandomizedSearchCV does not guarantee the best set of hyper parameters it has been used to maintain the trade off between performance and training time. In some cases, RandomizedSearchCV gives set of hyper parameters which reduces the performance as compared to default parameters. Similarly, the reduction in the performance after tuning the hyper parameters can be observed in the Table 3. It has been observed that the performance of the random forest classifier was higher by 1% with default parameters.

Table 3: Random Forest classifier Experiments

| **Experiment** | **Train Accuracy (%)** | **Test Accuracy(%)** |
|---|---|---|
| Default Parameters | 96% | 95% |
| RandomizedSearchCV with K-folds(10 folds) | 95% | 94% |

## 5.3 Experiment with Support Vector Machine Classifier

Support Vector Machine has been trained using default parameters and Gridseacrhcv with k- fold cross validation(10 folds) to obtain the best hyper parameters for cost(C) parameter. The training and testing performance of the model has been increase by 4% after hyper parameter tuning. While, the degree of over fitting (1%) has remained constant. The results of this experiment on Support Vector Machine(SVM) classifier can be observed in Table 4.

Table 4: Support Vector Machine Classifier Experiments

| **Experiment** | **Train Accuracy (%)** | **Test Accuracy(%)** |
|---|---|---|
| Default Parameters | 92% | 91% |
| GridSearchCV with K-folds(10 folds) | 96% | 95% |

## 5.4 Experiment with Artificial Neural Networks

In the experiments with Artificial Neural Networks two architectures have been built and trained. First architecture has 1 dense input layer, 5 hidden dense layers with 'ReLU' activation, each hidden layer is followed by 'Dropout' and 'BatchNormalization' layers. While, the output layer has 'softmax' as activation function and number of target classes 74 as neurons. The used loss function is 'sparse categorical crossentropy' and optimizer is 'Adam'. The second architecture is almost similar first architecture only the number of hidden layers is 4 and are in decreasing order. The 'Dropout' and 'BatchNormalization' layers have been used to reduce overfitting. The second architecture(4 hidden layers) has higher range of confidence interval as compare to the first architecture(5 hidden layers) which is observable in Table 6 and Table 5 along with parameter details of both architectures. To validate the scores confidence interval has been calculated.

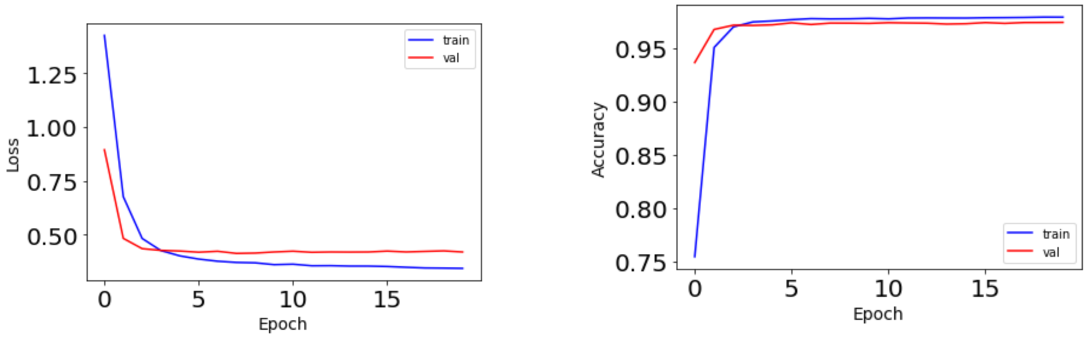
Table 5: Artificial Neural Networks Experiments (5 Hidden layers)

| Parameters | Training Accuracy | Test Accuracy | Confidence Interval |
|---|---|---|---|
| Epochs=20,<br>Batch size=500,<br>Hidden layers=4<br>Neurons=(1024,1024,1024,<br>1024,1024)<br>Activation='ReLU'<br>Optimizer = 'Adam'<br>O/P Activation = 'Softmax' | 96% | 94% | 93.6% - 94.7% |

Table 6: Artificial Neural Networks Experiments (4 Hidden layers)

| Parameters | Training Accuracy | Test Accuracy | Confidence Interval |
|---|---|---|---|
| Epochs=20,<br>Batch size=500,<br>Hidden layers=4<br>Neurons=(1024,512,256,<br>128)<br>Activation='ReLU'<br>Optimizer = 'Adam'<br>O/P Activation = 'Softmax' | 96% | 95% | 93.6% - 95.1% |

Figure 16 depicts the performance of 4 hidden layer Artificial Neural Network (ANN) architecture with respect to 'Loss' and 'Accuracy' by epochs. It can be noted that after around 4 epochs the accuracy of training and testing becomes almost equal while, the loss has kept decreasing for the training data.




(a) Epochs vs Loss (b) Epochs vs accuracy

Figure 16: Performance graphs of Artificial Neural Networks

## 5.5 Experiment with Bidirectional Long Short Term Memory

Experiments with BLSTM have been conducted by using two architectures where one has 128 units in bidirectional layer and 4 dense layers with reducing number of neurons(128,64,32,16) with 'ReLU' as activation function and output layer with 'softmax' activation and 74 neurons(number of target classes) depicted in Table 7. While, other architecture consists of 128 units in bidirectional layer and 3 dense layers with reducing number of neurons(512,256,128) with 'ReLU' as activation function and 'softmax' activation for output layer as per Table 8 . Additionally, both architectures have an embedding layer as input layer which processes the input data and weights from GloVe 6 Billion 200 dimension embedding matrix, 'sparse categorical crossentropy' as loss function and 'Adam' as optimizer function. Also, it is observed that in the second architecture there has been increment in performance in terms of confidence interval by 5% whereas decrease in training accuracy by 1%. To validate the scores confidence interval has been calculated.

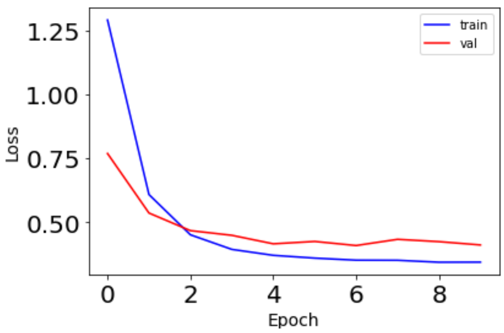
Table 7: Bidirectional Long Short Term Memory (5 Hidden layers)

| Parameters | Training Accuracy | Test Accuracy | Confidence Interval |
|---|---|---|---|
| Epochs=20,<br>Batch size=500,<br>Bi-LSTM layer=1,<br>LSTM units = 128,<br>Dense layers=4,<br>Neurons=(128,64,32,16)<br>Activation='ReLU'<br>Optimizer = 'Adam'<br>O/P Activation = 'Softmax' | 96% | 95% | 85.0% - 96.3% |

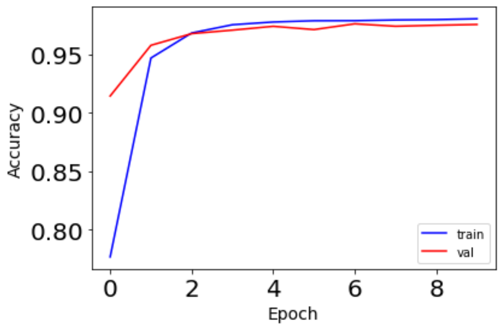
Table 8: Bidirectional Long Short Term Memory (4 Hidden layers)

| Parameters | Training Accuracy | Test Accuracy | Confidence Interval |
|---|---|---|---|
| Epochs=10, Batch size=500, Bi-LSTM layer=1, LSTM units = 128, Dense layers=3, Neurons=(512,256,128) Activation='ReLU' Optimizer = 'Adam' O/P Activation = 'Softmax' | 96% | 95% | 90.8% - 95.9% |

Figure 17 depicts the performance of 4 hidden layer Bidirectional Long Short Term Memory (BLSTM) architecture with respect to 'Loss' and 'Accuracy' by epochs. It can be noted that after around 3 epochs the accuracy of training and testing becomes almost equal while, the loss has kept decreasing for the training data.




(a) Epochs vs Loss




(b) Epochs vs accuracy

Figure 17: Performance graphs of Bidirectional Long Short Term Memory

Classification Report

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.95 | 0.68 | 0.79 | 131 |
| 1 | 0.90 | 1.00 | 0.95 | 136 |
| 2 | 0.98 | 0.87 | 0.92 | 144 |
| 3 | 1.00 | 1.00 | 1.00 | 137 |
| 4 | 0.90 | 0.91 | 0.90 | 141 |
| 5 | 0.99 | 0.96 | 0.98 | 132 |
| 6 | 0.91 | 0.97 | 0.94 | 142 |
| 7 | 0.99 | 1.00 | 1.00 | 143 |
| 8 | 0.99 | 1.00 | 0.99 | 137 |
| 9 | 1.00 | 1.00 | 1.00 | 141 |
| 10 | 0.97 | 0.98 | 0.97 | 128 |
| 11 | 0.94 | 0.95 | 0.95 | 123 |
| 12 | 0.95 | 0.94 | 0.94 | 141 |
| 13 | 0.99 | 1.00 | 1.00 | 139 |
| 14 | 1.00 | 1.00 | 1.00 | 134 |
| 15 | 0.99 | 1.00 | 1.00 | 126 |
| 16 | 0.99 | 1.00 | 1.00 | 124 |
| 17 | 0.96 | 0.97 | 0.97 | 110 |
| 18 | 0.97 | 1.00 | 0.98 | 130 |
| 19 | 0.99 | 1.00 | 1.00 | 148 |
| 20 | 1.00 | 1.00 | 1.00 | 144 |
| 21 | 1.00 | 1.00 | 1.00 | 120 |
| 22 | 0.99 | 0.98 | 0.98 | 132 |
| 23 | 0.99 | 0.95 | 0.97 | 147 |
| 24 | 1.00 | 1.00 | 1.00 | 133 |
| 25 | 0.98 | 0.98 | 0.98 | 143 |
| 26 | 1.00 | 1.00 | 1.00 | 122 |
| 27 | 0.98 | 0.98 | 0.98 | 129 |
| 28 | 0.99 | 0.98 | 0.99 | 127 |
| 29 | 1.00 | 1.00 | 1.00 | 115 |
| 30 | 1.00 | 1.00 | 1.00 | 155 |
| 31 | 0.99 | 1.00 | 1.00 | 125 |
| 32 | 1.00 | 1.00 | 1.00 | 132 |
| 33 | 0.96 | 1.00 | 0.98 | 136 |
| 34 | 0.98 | 1.00 | 0.99 | 120 |
| 35 | 0.99 | 1.00 | 1.00 | 138 |
| 36 | 0.96 | 1.00 | 0.98 | 134 |
| 37 | 1.00 | 1.00 | 1.00 | 94 |
| 38 | 1.00 | 1.00 | 1.00 | 150 |
| 39 | 1.00 | 0.96 | 0.98 | 134 |
| 40 | 0.98 | 0.75 | 0.85 | 133 |
| 41 | 0.99 | 1.00 | 1.00 | 128 |
| 42 | 0.78 | 0.91 | 0.84 | 138 |
| 43 | 0.99 | 1.00 | 0.99 | 136 |
| 44 | 1.00 | 1.00 | 1.00 | 121 |
| 45 | 0.81 | 0.67 | 0.74 | 123 |
| 46 | 0.98 | 1.00 | 0.99 | 125 |
| 47 | 1.00 | 1.00 | 1.00 | 136 |
| 48 | 1.00 | 1.00 | 1.00 | 131 |
| 49 | 0.99 | 1.00 | 1.00 | 119 |
| 50 | 1.00 | 1.00 | 1.00 | 114 |
| 51 | 1.00 | 1.00 | 1.00 | 142 |
| 52 | 1.00 | 1.00 | 1.00 | 148 |
| 53 | 0.99 | 0.56 | 0.72 | 121 |
| 54 | 1.00 | 1.00 | 1.00 | 132 |
| 55 | 1.00 | 1.00 | 1.00 | 145 |
| 56 | 0.93 | 0.61 | 0.74 | 133 |
| 57 | 0.95 | 0.80 | 0.87 | 130 |
| 58 | 1.00 | 1.00 | 1.00 | 128 |
| 59 | 0.98 | 1.00 | 0.99 | 120 |
| 60 | 1.00 | 1.00 | 1.00 | 135 |
| 61 | 1.00 | 1.00 | 1.00 | 120 |
| 62 | 1.00 | 1.00 | 1.00 | 120 |
| 63 | 1.00 | 1.00 | 1.00 | 133 |
| 64 | 1.00 | 1.00 | 1.00 | 133 |
| 65 | 1.00 | 1.00 | 1.00 | 151 |
| 66 | 1.00 | 1.00 | 1.00 | 132 |
| 67 | 0.99 | 0.98 | 0.98 | 127 |
| 68 | 1.00 | 1.00 | 1.00 | 131 |
| 69 | 1.00 | 1.00 | 1.00 | 145 |
| 70 | 0.96 | 1.00 | 0.98 | 130 |
| 71 | 1.00 | 1.00 | 1.00 | 113 |
| 72 | 0.28 | 0.73 | 0.40 | 143 |
| 73 | 0.96 | 0.44 | 0.60 | 150 |
| accuracy | | | 0.95 | 9783 |
| macro avg | 0.97 | 0.95 | 0.96 | 9783 |
| weighted avg | 0.97 | 0.95 | 0.96 | 9783 |

Figure 18: Classification report of BLSTM

Figure 18 depicts the classification report of BLSTM model where classwise precision, recall, f1 score and support can be observed. Classification report of all models have not been included due to space restriction.

| Model | Scores with Default parameters | | | Scores in Experiments | | |
|---|---|---|---|---|---|---|
| | Training Score | Testing Score | Confidence Interval | Training Score | Testing Score | Confidence Interval |
| K-Nearest Neighbours | 96 | 93 | -- | 98 | 96 | -- |
| Support Vector Machine | 92 | 91 | -- | 96 | 95 | -- |
| Random Forest Classifier | 96 | 95 | -- | 95 | 94 | -- |
| Artificial Neural Network | 96 | 94 | 93.6-94.7 | 96 | 95 | 93.6-95.1 |
| Bidirectional LSTM | 96 | 95 | 85-96.3 | 96 | 95 | 90.8-95.9 |

* All values are in percentage

Figure 19: Comparison of model performance

Figure 19 show the comparison of performance of the models.

## 5.6 Discussion

In this research, multiple classification algorithms have been trained and tested to find the best classification algorithm for Automatic Ticket Assignment. In comparison to study by Al-Hawari and Barham (2019) where SVM has been used as a classifier in a machine learning based Ticket classification system which has produced approximately 89% accuracy score, the SVM classifier used in this study has produced 91% with default parameters and 96 % with GridSearchCV with 10 fold cross-validation. Also, this research features deep learning neural models ANN and BLSTM which have not been previously used in Automatic Ticket Assignment context.

Machine Learning models namely KNN and Random Forest classifier have been implemented and evaluated in this research. The original dataset has multiple missing values, mojibake (grabled) text and incidents descriptions in multiple languages all these issues have been solved in initial preprocessing of the data. Additionally, the dataset has been identified to have severe class imbalance towards 'GRP_0' with around 49% of the tickets assigned. The class imbalance has been handled using combination of undersampling (sampling strategy 0:600) and oversampling technique which has reduced the class imbalance. Also, TF-IDf vectorizer technique has been applied on the dataset. After, these processing the number of features in the dataset had increased to 10736 which captured 100% variance of the data. To reduce number of features so that the training time could be minimized PCA(95% variance) has been applied. After training the models with default parameters optimization using RandomizedSearchCV and GridSearchCV with k-folds has been performed. All applied classifiers in this research have produced accuracy score of more than 85%.Comparison of model performance has been shown in Figure 20. The performance achieved by current research is much higher as compared to performance recorded in research by Mandal et al. (2018) where machine learning and deep neural networks namely linear SVM, Naïve Bayes, KNN, Random Forest, CNN and LSTM have been applied on email tickets.

Although, the research has produced remarkable results, the performance could have been more if the dataset in its original form was balanced. Also, the data set contains a huge inconsistency where the same incidents have been assigned to multiple groups. This could have been handled by providing more information such as timestamp, query

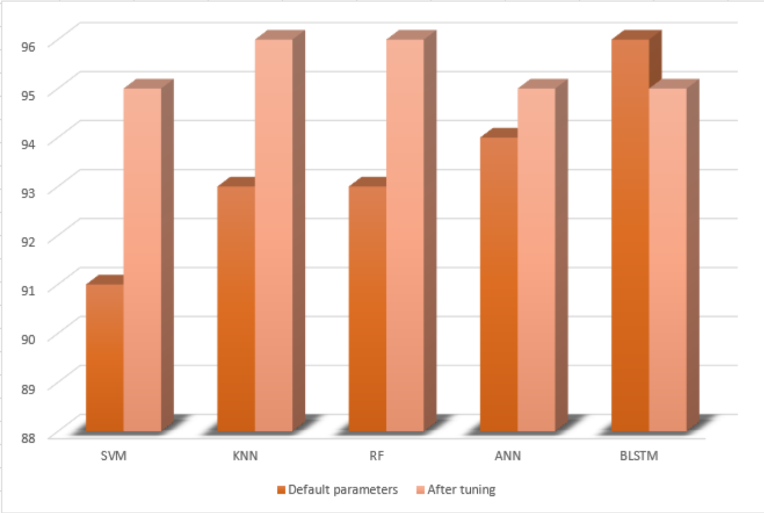



Figure 20: Comparison of model performance

number etc. about the data. Furthermore, performing another couple of iterations of pre-processing to remove more irrelevant words from the documents, training own Glove embeddings instead of using pre-trained glove vectors or embeddings created using Word2Vec could have helped in obtaining more reliable results. Moreover,the average training time of each model is around 30 - 40 minutes even with reduced features which makes maintaining performance verses training time trade off harder.

# 6 Conclusion and Future Work

In this research, an automatic ticket assignment using multi-class classification has been carried out in which various machine learning and deep neural networks have been trained and tested to identify the best text classification algorithm. Resampling techniques have been applied to balance dataset also, a combination of oversampling and undersampling has been performed to reduce the class imbalance problem. However, due to lack of resources implementation Glove embedding has not been carried out but pre-existing GloVe 6B 200 dimension file has been used. This research will be useful for real-world application. Both machine learning and deep learning models have been implemented in this research work however, ANN and BLSTM have not been explored in the previous research which makes a new contribution in this domain. The automatic ticket assignment system in an IT service industry has large number of benefits such as, providing high quality of customer service while cutting costs, reduce human error, increase throughput and reduce work load of customer service teams.

To improve the performance of models in future, techniques such as creation of GloVe embeddings with balanced original data(no target class imbalance), use of other text embedding techniques other than TF-IDF vectorizer, use of GridSearchCV with all classifiers and with all hyper parameters could help to an extent. Though, further research is needed for more methods for improvement or finding alternative solutions.

# Acknowledgement

At first, I would like to thank my research supervisor Dr Muhammad Iqbal for patiently supporting and encouraging throughout the research module. For around 13 weeks of the period, my supervisor has constantly assisted me with my doubts and has always motivated me in the right direction. Secondly, I would like to thank my family member for supporting me in this tough situation unconditionally.


# References


Al-Hawari, F. and Barham, H. (2019). A machine learning based help desk system for IT service management, *Journal of King Saud University - Computer and Information Sciences* (xxxx).
**URL:** *https://doi.org/10.1016/j.jksuci.2019.04.001*

Beneker, D. and Gips, C. (2017). Using clustering for categorization of support tickets, *CEUR Workshop Proceedings* **1917**(September): 51–62.

Grosser, S. T. (2015). ( 12 ) United States Patent, **2**(12).

Jing, R. (2019). A Self-attention Based LSTM Network for Text Classification, *Journal of Physics: Conference Series* **1207**(1).

K. S. Manjunatha, H. A. and Guru, D. S. (2019). *Data Analytics and Learning*, Vol. 43, Springer Singapore.
**URL:** *http://link.springer.com/10.1007/978-981-13-2514-4*

Khramov, D. (2019). Robotic and machine learning: How to help support to process customer tickets more effec- tively, (April).
**URL:** *https://www.theseus.fi/handle/10024/143200*

Koushik, V., Pappu, R., Higgins, L., Cited, R., Rangarajan, K. D., Paramasivam, L. and Data, P. P. (2019). ( 12 ) United States Patent, **2**.

Lee, J. Y. and Dernoncourt, F. (2016). Sequential short-text classification with recurrent and convolutional neural networks, *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference* pp. 515–520.

Li, C., Zhan, G. and Li, Z. (2018). News Text Classification Based on Improved Bi-LSTM-CNN, *Proceedings - 9th International Conference on Information Technology in Medicine and Education, ITME 2018* pp. 890–893.

Liang, D. and Zhang, Y. (2016). AC-BLSTM: Asymmetric Convolutional Bidirectional LSTM Networks for Text Classification.
**URL:** *http://arxiv.org/abs/1611.01884*

Liang, Q., Wu, P. and Huang, C. (2019). An efficient method for text classification task, *ACM International Conference Proceeding Series* pp. 92–97.

Mandal, A., Malhotra, N., Agarwal, S., Ray, A. and Sridhara, G. (2018). Cognitive system to achieve human-level accuracy in automated assignment of helpdesk email tickets, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **11236 LNCS**: 332–341.

Molino, P., Zheng, H. and Wang, Y. C. (2018). COTA: Improving the speed and accuracy of customer support through ranking and deep networks, *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* pp. 586–595.

Pradhan, L., Taneja, N. A., Dixit, C. and Suhag, M. (2017). Comparison of Text Classifiers on News Articles, *International Research Journal of Engineering and Technology(IRJET)* **4**(3): 2513–2517.
**URL:** *https://irjet.net/archives/V4/i3/IRJET-V4I3651.pdf*

Sachan, D. S., Zaheer, M. and Salakhutdinov, R. (2019). Revisiting LSTM Networks for Semi-Supervised Text Classification via Mixed Objective Function, *Proceedings of the AAAI Conference on Artificial Intelligence* **33**: 6940–6948.

Semberecki, P. and MacIejewski, H. (2017). Deep learning methods for subject text classification of articles, *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems, FedCSIS 2017* (November): 357–360.

Seongmoon, W. and Wenlong, W. (2009). Machine learning-based volume diagnosis, *Proceedings -Design, Automation and Test in Europe, DATE* (September): 902–905.

Silva, S., Pereira, R. and Ribeiro, R. (2018). Machine learning in incident categorization automation, *Iberian Conference on Information Systems and Technologies, CISTI* **2018-June**(351): 1–6.

Wang, R., Li, Z., Cao, J., Chen, T. and Wang, L. (2019). Convolutional Recurrent Neural Networks for Text Classification, *Proceedings of the International Joint Conference on Neural Networks* **2019-July**(2018): 1–6.

Yogatama, D., Dyer, C., Ling, W. and Blunsom, P. (2017). Generative and Discriminative Text Classification with Recurrent Neural Networks.
**URL:** *http://arxiv.org/abs/1703.01898*

Zhou, P., Qi, Z., Zheng, S., Xu, J., Bao, H. and Xu, B. (2016). Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling, *COLING 2016 - 26th International Conference on Computational Linguistics, Proceedings of COLING 2016: Technical Papers* **2**(1): 3485–3495.

Zhou, W., Xue, W., Baral, R., Wang, Q., Zeng, C., Li, T., Xu, J., Liu, Z., Shwartz, L. and Grabarnik, G. Y. (2017). STAR: A system for ticket analysis and resolution, *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* **Part F1296**(August): 2181–2190.