S	
E1	
E2	
For Office Use Only	

Masters Project Final Report

(MCS)

2019

Project Title	An Approach to Hate Speech Detection
Student Name	D.H.A. De Silva
Registration No. & Index No.	2017/MCS/020 17440208
Supervisor's Name	Dr. A.R. Weerasinghe

For Office Use ONLY

An Approach to Hate Speech Detection

A dissertation submitted for the Degree of Master of Computer Science

D.H.A. De Silva
University of Colombo School of Computing
2019

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: D. H. A De Silva

Registration Number: 2017/MCS/020

Index Number: 17440208

____

Signature: Date:

This is to certify that this thesis is based on the work of

Mr./Ms.

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Dr. A. R. Weerasinghe

____

Signature: Date:

ABSTRACT

Hate speech on social media becomes a highly considerable issue which is growing rapidly. As a result of the growth of internet users people tend to post on violence contents through social media. Therefore, the influence of sharing violence contents towards individuals and groups becomes a huge impact in today's world and it directs to increase hate crimes in the society. It is essential to have a proper methodology to detect the online hate contents.

Although there are many researches have been carried out based on this area, they are language specific things to detect hate contents. This research has been carried out to develop an efficient and accurate approach to detect the hate speech on social media using Sinhala Language. When comparing with the English language, to develop an approach to detect hate speech on Sinhala is a tedious task because of the large alphabet and its variations.

In order to develop a model to detect hate speech on Sinhala Language, Machine learning and Deep Learning techniques were used as the core approaches. As the solution for this research, four supervised learning approaches including Linear Support Vector Machine, Logistic Regression, Naïve Bayes, Random Forest and Deep Neural Network were used to train the models and predict the accuracy values. Both Count Vectorizer and TF-IDF features used to train the models for Sinhala, Singlish and Mix datasets. Furthermore, to increase the performance level Cross Validation approach have been carried out for Count Vectorizer and saved the model results generated from Cross Validation. Then after that developed an Ensemble model by combining highest accurate models and taking different model combinations of Sinhala, Singlish and Mix data sets to get the highest accurate model. Finally, combination of Sinhala-Singlish data compared with Mix data and it predicted that concatenate with Sinhala-Singlish contents gives a highest accuracy for the hate speech detection model. Since there is no proper mechanism of using Ensemble model for hate speech detection research, this approach will direct as the proper methodology to detect the hate speech on Sinhala Language.

ACKNOWLEDGEMENT

Firstly, I would like to express my sincere gratitude to my supervisor Dr. A.R Weerasinghe, senior lecturer of University of Colombo School of Computing for the continuous support of my research work, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

I would also like to thank our research project coordinator Dr. B H R Pushpananda for his guidance given throughout the year. Also, I want to show my gratitude for the university staff and all the lecturers for the support given to complete this research successfully.

Finally, I would like to thank my family and my friends for their valuable support to make the project success. It’s a great pleasure to acknowledge the assistance and contribution of all people who helped me to complete my research successfully.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1: Introduction

1.1 Problem Domain

Rapid development of increasing the number of internet users in the world, especially for the social media category. Nowadays people tend to use Social Media without any responsibility as adding contents, sharing contents without any consideration. Some of the contents perform the violence attitudes towards the other individuals or groups. Therefore, Hate Speech has become a problem around the world because of misusing the internet. Hate speech can be identified through main three areas as ethnicity, religion and race [1]. This research mainly focuses on develop an approach to detect hate speech on Sinhala Language.

1.2 Problem

Hate speech on social media becomes a problem because people tend to post violence contents in different languages [2]. Therefore, it is difficult to find the contents which are related to hate speech for social media like Twitter, Facebook. For the Sinhala language it would be difficult to identify the contents which go against the community standards. The contents with the racist words, offensive language, and it is a tedious task to categorize a word-set without unambiguous hate keywords in Sinhala as it has more combination of words. Therefore, this research basically creates an approach to analyze and identify the Sinhala contents from social media which can be categorized in to hateful speech.

Advances in communication technology has brought people to one global position. They play the major role with granting the freedom of speech including allowing express their thoughts behaviors and opinions freely. Although this makes a great opportunity racism, trolling, being exposed to large amounts of offensive online contents. Therefore, the rapid growth of hate speech on social media becomes a big impact to the society [3].

Moreover, various social media platforms such as Facebook, Twitter and YouTube have their own diverse procedures to deal with the hateful language contents. As an example, You Tube permits the free speech, but prohibits the hate speech. It has the manually reporting feature to report hate contents. Not only that Facebook, Twitter also have policies to prohibit hateful contents. In Facebook, reporting feature is there in order to prohibit hateful posts and also it has some features for the user protection including un-friend, blocking etc. Finally, the detected hateful contents will be removed from the online.

Sri Lanka has the same kind of problem related to online contents for detecting hate speech in Sinhala Language. Although current researchers have done automated detectors for the hate speech, still there is no generalized mechanism because of the problem of language dependency. Therefore, this research is mainly focused on developing an approach to detect offensive contents for Sinhala Language [4].

1.3 Motivation

Online communication channels including social media dedicated to community-based input, interaction, content sharing etc. While social media helps people with connecting each other, sharing knowledge regardless of location, education background, updating information around the world, it enables the risk of people to being targeted or harassed via offensive language which may severely impact the community in general. In fact, there is no general mechanism to identify hate speech in different languages at once. When it comes to the Sinhala Language, currently people tend to post hateful contents regarding different subjects and situations. Unless people do not report about the offensive contents it remains there. As an example, when one-person comment with hate contents there is a possibility to get the reply with hate contents. Therefore, there should be a proper mechanism to detect the hate contents in online.

Furthermore, as most of the children use the internet nowadays, it’s hard to differentiate hate/ not hate contents. As a result, the violence is increased among people including children and also the children will learn things that they don’t deserve to learn. In fact, detection of hate speech will help to improve the quality of the communication on online as well as it helps to eliminate the violence among people.

The main motivation to do this research is, because there is no proper mechanism to detect the hate speech contents in Sinhala Language. Therefore, an approach to a Sinhala hate speech detection will be carried out by this research.

1.4 Exact Computer Science Problem

Many researches have been carried out throughout the past years under the hate speech detection. The earlier research solutions were carried out on different language hate speech detections, because it’s hard to build a generalized approach. Most of the earlier approaches are based on neural network, machine learning, and deep learning for hate speech detection on various languages including English, Hindi, and Indonesian etc.[3] Classification including binary and ternary, NLP techniques for word analysis, deep learning approaches, lexicon-based approaches, etc. were used for the earlier approaches[6][7]. But developing a generalized approach would be a more complicate and difficult task.

The computer science problem that going to be addressed for this research is, to identify a proper mechanism to detect hate speech on Sinhala language by creating a proper data set by preprocessing it by removal of special characters, stop words and stemming NLP techniques, and proper machine learning/deep learning approaches for create models to detect hate speech. Moreover, it would give more accurate and efficient hate speech detection mechanism for Sinhala language.

1.5 Research Contribution

Objectives:

Following are the main objective and sub objectives which can gain through identifying Hate Speech in Sinhala.

- The main objective of this research is to develop an approach to detect the contents of hate speech in Sinhala Language in Social Media context.

Sub Objectives:

- Collect a representative sample of Sinhala/Singlish hate speech from social media.
- Find the best way to annotate posts as hate or not.
- Explore algorithms that would help in training a suitable model from the training data (intend to apply machine learning algorithms/deep learning algorithms as part of the exploration algorithms).
- Perform model diagnostics in order to validate the model.

1.6 Scope

Hate Speech Detection will become a very important approach with the developing world. This research is mainly focused on developing an approach to detect hate speech in Sinhala context because there is no proper mechanism until now to detect Sinhala Hate Speech in an accurate manner. This research involved to create a training model with the proper data set. Before creating the model, it needs to be done preprocessing the data set by collecting and annotating data, remove unwanted things and create a baseline. Word lexicon and stemming approaches can be used with this research to get the noise free context.

Furthermore, machine learning and deep learning approaches/algorithms will be used to analyze the hate speech from the text and flag them. Basically, it is going to suggest the hate speech from the text. Nowadays for some hate speech detection can be done using moderators. But it is not an efficient way. This research directs to find the hate speech in efficient manner. This research tests the models with Sinhala, Singlish and Sinhala-Singlish Mix data sets. Finally, this will provide an accurate hate speech identification mechanism for Sinhala Language and it will help to overcome hate contents from online communication.

1.7 Structure of the Dissertation

The chapters of the dissertation give the specific details in an order to explain the overview of the project. After going through the domain of the project identified the problem statement, problem, exact computer science problem, and scope details mentioned under Chapter 1. In chapter 2, explains the literature review with the vital description of the study related to the problem statement. Furthermore, the stated studies under this chapter are the current knowledge and new methods with respect to the research.

The chapter 3, explains the selected methodology in order to achieve the target of this research including creating the dataset, preprocessing the data set, Machine learning/ Deep learning approach. The chapter 4 explains the evaluation strategy and results of the evaluation including accuracy.

Chapter 2: Literature Review

2.1 Introduction

Literature Review chapter explains the critical review of the research in the area of Hate Speech Detection. For this purpose, the Chapter has been structured as early developments of hate speech detection, latest achievements of hate speech detection and future trends, an approach for filling the existence research gap by using literature review and novel methodologies. The chapter also defines the research problem based on the literature review in the areas of Natural Language Processing and Machine Learning.

2.2 Early developments in Hate Speech Detection

Following are the numerous varieties of tactics which used for hate speech exposure.

2.2.1 Classification with the Machine Learning Approaches

Machine learning became a mostly used approach in various types of researches. Basically, it used to develop algorithms and mathematical models for the computer systems without using explicit interference. From the early stages machine learning approaches had been used to identify hate speech from various languages. Following are the research approaches which are carried out for hate speech detection using machine learning.

As the first cited work, A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection basically carried out to find the hate speech on Twitter [1] [2]. With the vast development of users of Social Networks and Microblogging web sites, conflicts between people are increased. Therefore, lot of problems occurred due to the hatred speech. According to this paper Hate Speech defines as aggressive, violent and offensive language. This research identifies hate speech expression by using unigram as well as patterns which are collected from automatically trained dataset. Furthermore, machine learning algorithm is used to train a dataset. Moreover, the binary classification techniques are used to identify that a particular tweet is an offensive or not and ternary classification approach detects that the tweet is hateful, offensive or clean. This approach helpful to overcome noise and the non-reliability of data [3]. They have preprocessed the data with decomposing hash tags, tokenization POS tagging lemmatization and generating a negative vector [4]. As the limitations of this research and as the future work it needs a richer dictionary for hate speech which can be used with unigram patterns. Also, it needs to be increase with detect hate speech different genders, age groups and religion.

Furthermore Application of Machine Learning Techniques for Hate Speech Detection in Mobile Applications research was carried out to detect Hate Speech on Mobile Applications [5]. Basically, versatility and omnipresence of data makes very hard to detect trustworthiness intention of the dynamic environments like mobile applications. This research carried out light weight machine learning classification to identify hate speech in Albanian language which used for Mobile Applications. This approach indicates the good classifier accuracy. According to this research it carried out different technologies, platforms, libraries which developed with machine learning backend techniques to find hate speech on Albanian social network users. The neural network is used to provide accurate results through this research. This paper has some limitations as there is no vote or count voting system for users since all the data is open to users. Moreover it addresses future work as train the dataset with NLP techniques for specific word analysis and deep learning approach to identify hidden types hate speech in the text [6] [7] [8].

2.2.2 'Lexicon-Based' tactic with the Machine Learning Procedures

Nowadays in many researches, 'lexicon–based' tactic united with the various kinds of machine learning procedures have been used. Research problems like hate speech detection can also apply this approach. Lexicon approach describes the usage of dictionary of words annotated with their semantic strength and calculate the score for the sentiment of the document.

Some researchers have done an experiment to identify and detect the hate speech by means of social media and it has been mainly carried out to figure out hate speech contexts on social media using local English text dataset [9]. This is used to identify hate contents from social media and flag them automatically. Supervised and unsupervised learning techniques were used to identify the hate contents. Naïve Bayes classifier with 'Tf-idf' features helped to perform the best result also. The main key area in this research is automatically identification on hate speech contents from online, so that five models had been built using supervised and unsupervised techniques and concluded that supervised learning techniques perform the better results. As the limitation it mentioned data annotation is one of the difficult tasks and for the future work more unsupervised techniques will be used to get better results.

'The Automated Hate Speech Detection and the Problem of Offensive Language' research approach was basically targeted to identify 3 basic categories as hate speech, offensive language and those with neither by using the collected tweets [10]. Although social media like Facebook, Twitter have responded to criticism, they are not doing enough to identify hate speech and overcome the attacks on people. This model have some key challenges to provide more accurate classification [11]. This research used a logistic regression with L2 regularization for the final model as it more readily allows us to examine the predicted probabilities of class membership. As the limitation it does not include all instances of offensive language.

2.2.3 Convolutional Neural Networks Approach

Convolutional Neural Network considered as the class of deep neural network and commonly applied for analyzing visual imagery. This approach also applied to the hate speech detection in order to increase the accuracy of the results. Following are some research approaches which used the Convolutional Neural Network technique.

Hate speech detection in Indic Languages also becomes and important aspect in research world. According to the research 'Hate Speech Detection from Code-mixed Hindi-English Tweets Using Deep Learning Models', detected hate speech from English – Hindi code mixed tweets [12] [13]. In India, Hindi is one of the main official languages and many people use this language when using social media. This research proved that domain specific embedding results in an efficient manner to detect hate speech contents. According to the past static classifiers it improves 12% of improvement. The country with the highest internet penetration and rich linguistic diversity hate speech detection is an important aspect in India. The main challenge is to create and use the code-mixed data set. This research used deep learning approach and compare with statistical approach by using the same data set and it provided higher accurate results than statistical method [14]. For that they have trained the work embedding on a large corpus of relevant code-mixed data. As the limitations of the research mentioned that some misclassified series of swear words, possibly incorrect labels which cannot consider as hateful word, and code-switched words in Hindi. Finally, as the novelty part mentioned as assimilating textual cues more accurately.

2.2.4 Recurrent Neural Network Approach

Recurrent Neural Network is considered as class of an artificial neural network where connections between nodes form a directed graph along a temporal sequence and it used for temporal dynamic behavior. This is also used for increasing the accuracy level of the results. In hate speech detection research, following are some earlier approaches which are carried out with Recurrent Neural Network.

The ʻAnalysis Text of Hate Speech Detection Using Recurrent Neural Network' research mainly target to identify the hate speech on Twitter [15] . According to this research the main problem they have identified as social media does not have a method to aggregate information from the existing conversation. The only way to do aggregation is text mining. The Deep Learning and Recurrent Neural Network techniques can identify the text containing hate speech or not. Tokenizing, Cleaning, stemming etc. techniques were used to analyze the text. The limitation of this approach is because of the Epoch, Learning Rate and Batch Size techniques affected to the performance, and accuracy results. Epoch explains the pass entire dataset forward and backward through the neural network. Learning Rate explains calculation of the value of weight correction during the training process. Finally, Batch Size defines the total no of training samples in a batch. Furthermore, by using the well-trained data the output will be more accurate.

2.2.5 Deep Learning Approach

Deep Learning is the part of machine learning methods which is based on artificial neural networks. These days' usage of deep learning approaches become increases for the hate speech detection techniques. Early research projects also intended to use deep learning approach because of the accurate results. Following approach is one of the results of using deep learning for the hate speech detection.

The 'Detecting Offensive Language in Tweets Using Deep Learning' paper identified that simple word based approaches if used for blocking the posting of text or blacklisting users, not only fail to identify subtle offensive content, but they also affect the freedom of speech and expression [6] [16]. Not only that it reported performance for a simple LSTM classifier not better than an ordinary SVM. Unsupervised learning approaches are quite common for detecting offensive messages in text by applying concepts from NLP to exploit the lexical syntactic features of sentences or using AI-solutions and bag-of-words based text-representations [17].

2.3 Latest achievements and future trends in the area of Hate Speech Detections

According to the table 2.1 security can be considered as one of the major concerns about hate speech detection in machine learning/deep learning. Therefore, it is essential to focus on freedom of expression and reducing the illegal discrimination. Further deployment and maintenance of hate speech detection using machine learning/deep learning solution has been researched.

Table 2.1: Achievements and Limitations of Different Researches

Research	Achievements	Limitations
Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection (H. Watanabe et al.)	Identify hate speech expression by using unigram as well as patterns which are collected from automatically trained dataset. Binary Classification and Ternary Classification. Data set with decomposing hash tags, tokenization POS tagging lemmatization and generating a negative vector.	A richer dictionary for hate speech which can be used with unigram patterns. Detect hate speech different genders, age groups and religion.
Application of Machine Learning Techniques for Hate Speech Detection in Mobile Applications (B. Raufi et al.)	Light weight machine learning classification to identify hate speech in Albanian language which used for Mobile Applications. Neural Network Approach.	No vote or count voting system for users since all the data is open to users.
Detecting Offensive Language in Tweets Using Deep Learning (G. K. Pitsilis et al.)	A deep learning architecture for text classification. Usage of pre-trained word embedding, Performance.	Analyzing texts written in different languages.
Identification of Hate Speech in Social Media (N. D. T. Ruwandika, A. R. Weerasinghe)	‘lexicon-based’ tactic united with a machine learning procedure.	Data annotation is one of the difficult tasks.

Automated Hate Speech Detection and the Problem of Offensive Language (T. Davidson et al.)	A logistic regression withL2 regularization for the final model for Accurate classification.	Does not include all instances of offensive language.
Hate me, hate me not: Hate speech detection on Facebook (M. Petrocchi et al.)	First classifier for Italian Language, enlarging the annotation process	Classifier Performance
The Commonwealth Scientific and Industrial Research Organization	Deep learning approach and compare with statistical approach by using the same data set and it provided higher accurate results than statistical method.	Misclassified series of swear words, possibly incorrect labels which cannot consider as hateful word, and code-switched words in Hindi.
Detecting Offensive Tweets in Hindi-English Code-Switched Language (P. Mathur et al.)	Classification of the tweets in HEOT dataset using transfer learning. Convolutional Neural Networks is pre-trained on tweets in English followed by retraining on Hinglish tweets.	As the tweet data in Hinglish language is a small fraction of the large pool of tweets generated.
Analysis Text of Hate Speech Detection Using Recurrent Neural Network (A. S. Saksesi et al.)	The Deep Learning and Recurrent Neural Network techniques can identify the text containing hate speech or not.	Learning Rate and Batch Size techniques affected to the performance, and accuracy results.
Detecting Hate Speech in Tweets Using Different Deep Neural Network Architectures (B. R. Amrutha et al.)	Method to improve hate speech classification (improve F-measure). Confidence.	Lacking reproducibility. Post experimentation.

2.4 Approach for filling the existing research gap by using literature review

Based on the literature review identified that Hate Speech Detection is an important aspect for nowadays. Hate speech on social media becomes a problem because people tend to post violence contents in different languages. Therefore, it is difficult to find the contents which are related to hate speech for social media like Twitter, Facebook. For the Sinhala language it would be difficult to identify the contents which go against the community standards. The contents with the racist words, offensive language, and the words without explicit hate keywords are difficult to classify in Sinhala as it has more combination of words. Therefore, this research basically creates an approach to analyze and identify the Sinhala contents from social media which can be categorized in to hateful speech.

This proposed research approach is going to build a hate speech detection on Sinhala language. Mainly this research focuses to develop the training model and use machine learning/deep learning approaches to identify hate speech from the Sinhala context. As the first step, proper data set needs to be collected. The dataset will be containing Sinhala contents collected from free accessible websites and social media such as Twitter, after extracting Sinhala comments using web scraping methods.

In addition, this research aims to use machine learning and deep learning as the major technologies in order to train the models to identify the best approach that is used to detect the context which contains hate speech in Sinhala language. Moreover, collected data needs to be preprocessed with the noise free context and the training model will be developed. The appraisal of hate speech comment review is a classification problem frequently called sentiment analysis. Stemming approach can be used to collapse distinct words and reduce the dictionary size and it helps to sharping the results with good accuracy level. Also, with trained model we can test the contents which are categorized under hate speech.

By comparing most applicable and recently used deep learning and machine learning algorithms and by observing the accuracy level of each and every approach, it is going to select the most suitable and accurate machine learning approach for this research.

2.5 Summary

This chapter presented a critical review of the research in the areas of machine learning/deep learning and natural language processing as the major output of literature review, we have defined the research problem as an approach to Sinhala hate speech detection using machine learning and natural language processing. It is also identified that suitability of neural network solution composed of Long-Short-Term-Memory (LSTM) and machine learning classification techniques used to identify offensive language contents and making the awareness of the general public.

Chapter 3: Methodology

3.1 Introduction to Methodology

The methodology explains the way of research has been handled with the knowledge of literature review. As mentioned in literature review chapter the scope of this project spread over many modern computer science fields. As mentioned in above chapters the goal of this research is to find an efficient approach to detect Sinhala hate speech. This chapter provides a comprehensive overview of implementation steps which has been carried out to develop this solution. Therefore, this chapter explains the data set, preprocessing data, and machine learning/deep learning approaches which were used to accomplish this task.

3.2 Research Problem

Nowadays hate speech becomes increasing all over the internet but identification of hate contents still didn’t achieve to the satisfied target. Therefore, hate contents posts are increasing day by day. We can’t identify the generalized solution for hate speech detection because it depends on the language. According to this research, it’s going to find the approach for hate speech detection for Sinhala language.

According to the literature survey, most of the social media has manual process to report the offensive posts also they have policies as well. When it comes to the Sinhala language, there is no proper hate speech detection introduced yet. Huge alphabet combination may be one of the key elements for lacking the hate speech detectors for Sinhala language. Therefore, this research is for finding an approach for Sinhala hate speech contents.

Furthermore, literature survey proved that there are many approaches already taken for detecting hate speech on different languages including Hindi, English, Indonesian etc [14]. Some of the main approaches are deep learning, classification with machine learning approaches, neural network approaches etc [16]. But still there is no proper solution for the Sinhala language. Sinhala contents gathered from social media will be used for creating the data set for the proposed model. After preprocessing the dataset machine learning approaches will be used to identify hate speech contents on Sinhala language efficiently. The system will be used python language besides web scarping tools will be used to extract data from social media.

3.3 Proposing Model/Design

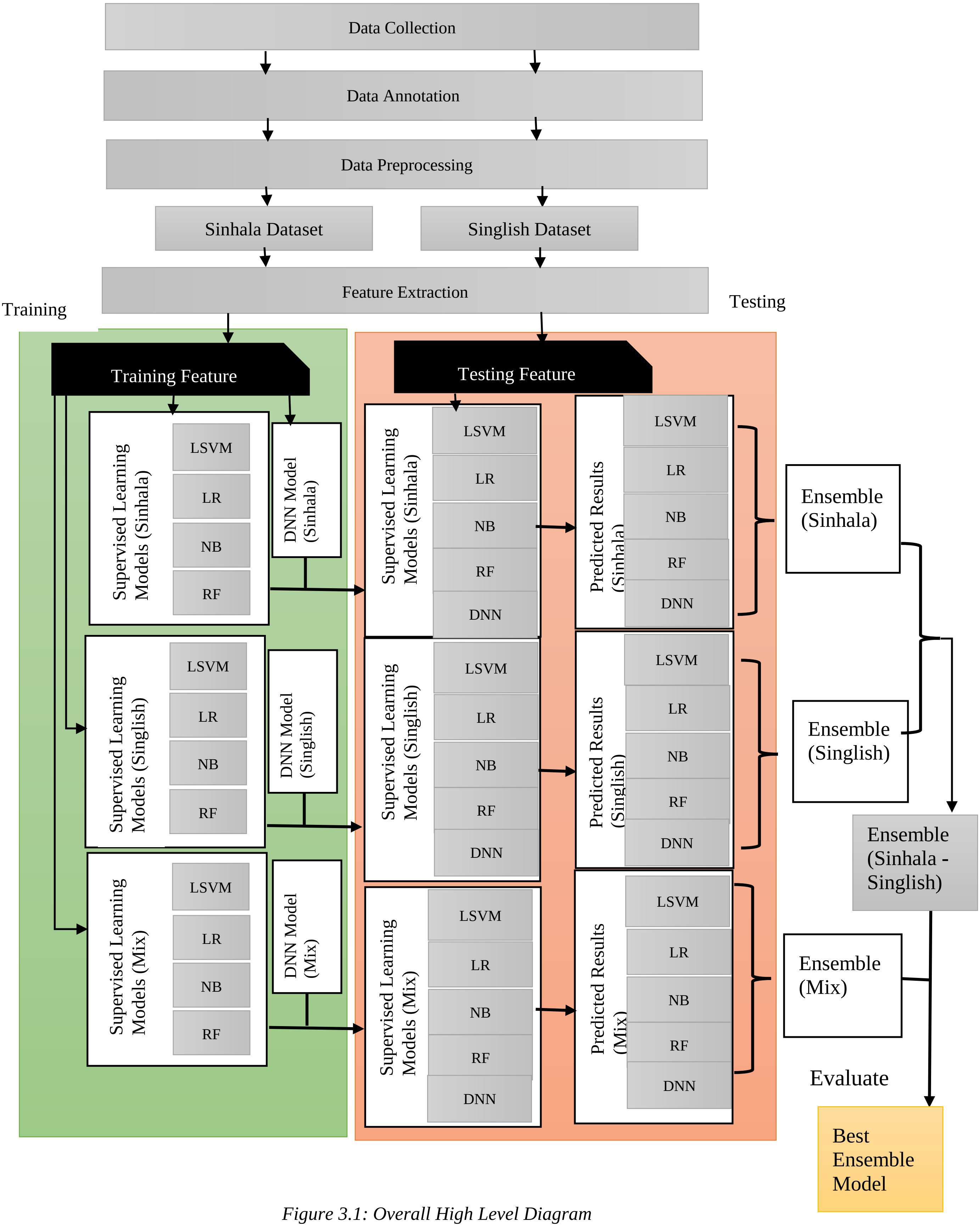


Figure 3.1: Overall High Level Diagram

As shown in figure 3.1, overall high level diagram explains the overall architecture of Hate Speech Detection Research approach. Initially collected data was annotated and preprocessed, in order to create two data sets called Sinhala and Singlish at the end of preprocessing stage. Then feature extraction has been done using Count Vectorizer and TF-IDF Vectorizer. In the training stage, four supervised learning models and deep neural network have been used for Sinhala, Singlish and Mix data sets separately. Furthermore, in the testing stage it combined all four classifiers with deep neural network and predict the result for Sinhala, Singlish and Mix data sets by using Ensemble model and generate results for all three data corpus. Finally, by combining the Sinhala-Singlish ensemble models it can be used to compare the accuracy of the model with Mix ensemble model with the intention of getting the best model.

3.4 Data Set

To perform a successful experiment on hate speech detection availability of a labeled corpus is really important. As the first step, proper data set needs to be collected. The dataset will be containing Sinhala contents collected from free accessible websites and social media such as Twitter.

The data set is created using the social media comments (Twitter). They have given great opportunities to users/readers to express their ideas. Also, they have provided the policies in order to safe the platform. But people posting some offensive contents. Therefore, some posts with offensive comments will be removed manually because it doesn't go with the community standards. The prepared dataset consists around 3500 comments and they were manually labeled by considering the full meaning of the comment and if it comes under hate speech the label is “Yes” otherwise “No”. Table 3.1 represents the overview of annotated data for the training dataset as “hate” or “not hate”.

Table 3.1: Classes of training dataset

No of Hate Comments	**1319**
No of comments without Hate	2276

Furthermore, separate dataset consists around 286 comments used as the testing dataset and it is also manually labeled by considering the full meaning of the comment same as the training set.

Table 3.2: Classes of testing dataset

No of Hate Comments	**157**
No of comments without Hate	129

3.5 Definition of the Hate Speech

As stated by numerous researchers, hate speech can be defining in various ways, but in this research data set was manually labeled in relation to the below hate speech explanation.

"Hate speech is the usage of language to insult or spread hatred towards a particular group or individual based on religion, race, gender or social status."

3.6 Preprocessing

This research aims to use natural language processing, and machine learning as the major technologies in order to train the model to identify and suggest the context which contains hate speech using Sinhala language. Moreover, the training model will be developed and before using the data in the trained model, collected data needs to be preprocessed with the noise free context. Besides it is essential to label the data showing whether it is hate speech or not. The appraisal of hate speech comment review is a classification problem frequently called sentiment analysis. Stemming approach can be used to collapse distinct words and reduce the dictionary size and it helps to sharping the results with good accuracy level. Eventually, this research will need a bag of words where it contains selected labeled words with hate speech or not, besides we need a testing dataset to test the trained model. Also with trained model we can test the contents which are categorized under hate speech.

3.7 Feature Extraction

In Feature Extraction, feature can be defining as a property of the instance that is being classified. In this hate speech detection research, instance is a comment/ sentence and features like words are highly specific. When training the models for hate speech detection the input to the models is the set of features which represent comments in our corpus and all feature extractions were done using scikit learn python library which is used widely for machine learning purposes with python. In this research Count Vectors and TF-IDF features were used for extract features from Sinhala, Singlish and Mix datasets [7].

3.8 Classification Models and Neural Network Model

As the models of the hate speech detection research, four different classifiers including Linear Support Vector Machine, Logistic Regression, Naïve Bayes, and Random Forest along with a Deep Neural Network were tested to find the best model [9] [17]. This was deliberated as a binary classification task, using the implementations and libraries present in scikit-learn toolkit. This task will be done for Sinhala, Singlish and Mixed datasets separately.

3.8.1 Supervised Learning Models

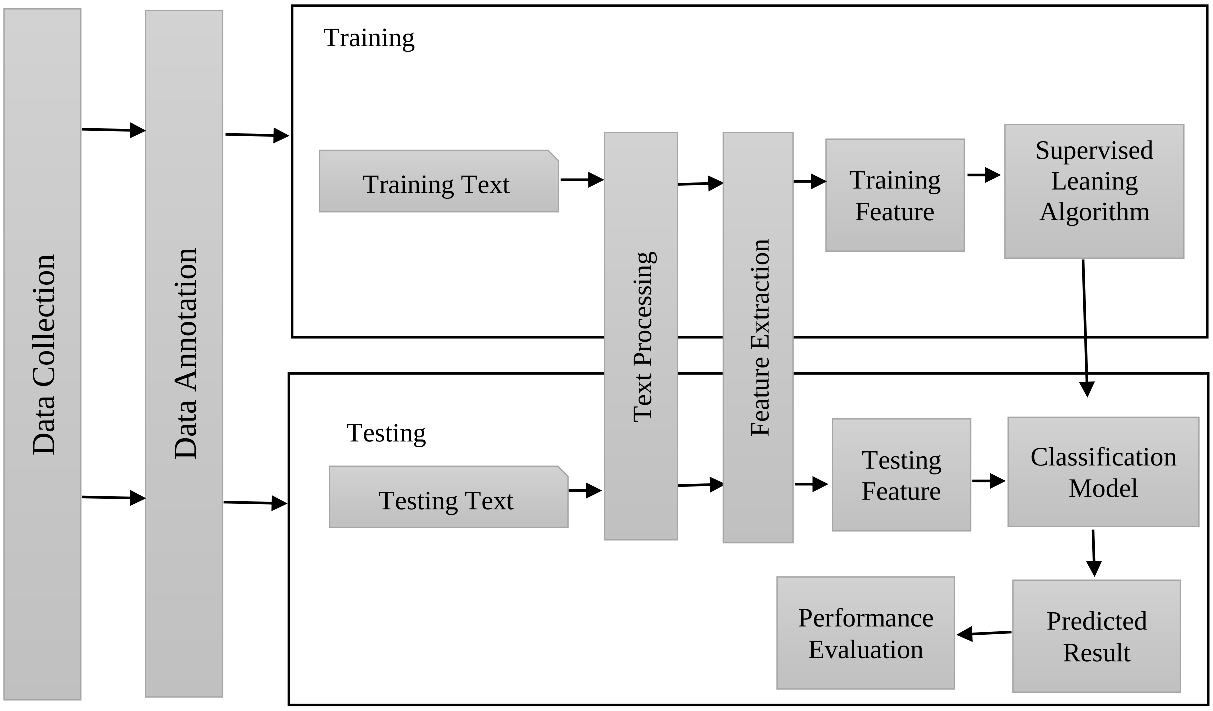


Figure 3.2: High Level Diagram for Supervised Learning models

Figure 3.2 embodies the comprehensive illustration of the system scheme for the model constructed using a supervised learning procedures. Both supervised/unsupervised learning procedures ‘data collection’, ‘data annotation’, ‘text preprocessing’, ‘feature extraction’ and ‘performance evaluation’ are the most common steps. The precise dissimilarity is to train the supervised model features of text data is utilized with the classes of training data but unsupervised model classes are not utilized and there is no precise training and testing stage there. In this investigation subsequent supervised learning models were utilized for hate speech detection.

Linear Support Vector Machine is supervised learning approach which is used for classification. It is used mostly where number of dimensions is greater than the number of samples. The SVM supports both Dense and Sparse vectors as input. In the scikit learn contains classes including SVC, NuSVC and LinearSVC to perform binary and multi classes classification in machine learning for the data set. In our study used Linear SVM and Stochastic Gradient Descent Classifier (SGDClassifier) with its default loss parameter to build the SVM was used.

Logistic Regression is a statistical model is used for analysis where the dependent variable is categorical. In this research our task is binary classification and using this model we can find the best fitting model to describe the independent and dependent variable relationship. In this research Scikit-learn toolkit's Logistic Regression Classifier with its default parameters is used to get the performance results.

Naïve Bayes classifier is also a supervised learning model and it is based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. In our research Scikit-learn toolkit's MultinomialNB classifier used for this experiment.

Random Forest is a supervised learning model and it is used for both classification as well as regression. Random forest algorithm creates decision trees on data samples and then generate the prediction. By using this approach it selects the best solution. In our research Scikit-learn toolkit's RandomForestClassifier used with default parameters to get the accuracy results.

3.8.2 Deep Neural Network Model

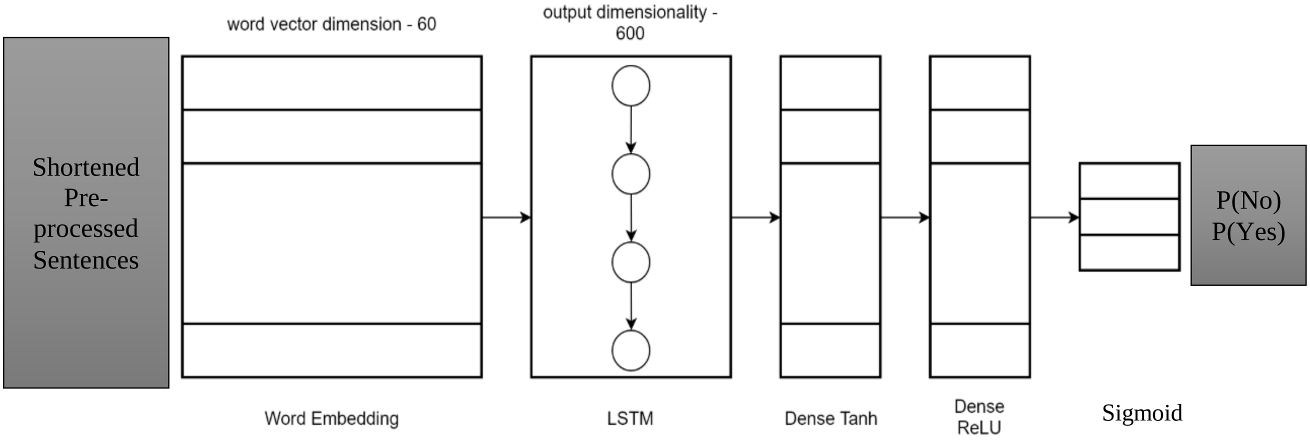

word vector dimension - 60
output dimensionality - 600
Shortened Pre-processed Sentences
P(No)
P(Yes)
Word Embedding
LSTM
Dense Tanh
Dense ReLU
Sigmoid

Figure 3.3: Deep Neural Network Model

Deep Neural Network approach is another way that was used to detect Sinhala hate speech contents in this research. Neural Networks are computational networks which were vaguely inspired by the neural networks in the human brain. It consists of nodes which are connected as nodes. Deep Neural Network have been tested for Sinhala, Singlish and Mix datasets.

In this research we used LSTM approach detect hate speech contents. LSTM networks are a special kind of RNN, which are capable of learning long-term dependencies. LSTM approach used to remember the information for the long periods of time is practically their default behavior [15]. In this research LSTM approach had been used for Sinhala, Singlish and Mix data sets.

Furthermore, this research used sequential model and multi-layer architecture. The multi-layer structure constructed using Embedding, LSTM and Dense layers. Dense layer constructed using three activation functions including ‘tanh’, ‘relu’ and output layer ‘sigmoid’ respectively. We used ‘sigmoid’ function as the output because this research use binary classification, so that it generates one output value. When it comes to multi class classification we can use ‘softmax’ function. Keras embedding layer used for text data and it requires to send to integer encoded data to input so that each word will be represent as unique integer. Also, Dense layer receive values from embedding layer, so that before sending sparse data to the dense layer it is going to be embedded data in the embedding layer. In optimizer we used ‘Adam’ with parameters including learning rate. Learning rate parameter ensures that how much it is going to learn when moving from one epoch to another.

In the beginning of the neural network approach Stratified K-Folds cross-validator was used because it provides train/test indices to split data in train/test sets. This research we have selected the fold count is three because after the 3rd fold there is no change of accuracy increase or decrease of loss. Furthermore, in deep neural network follows the training steps including set epochs’ history, select the best model and save, load saved model and save the prediction results.

3.8.3 Cross Validation

Cross-validation is a technique for evaluating Machine Learning models by training several Machine Learning models on subsets of the available input data and evaluating them on the complementary subset of the data. This can be used to detect and minimize overfitting problems. In hate speech detection research cross validation used in supervised learning models including Linear SVM, Naïve Bayes, Logistic Regression, Random Forest and Deep Neural

Network model for Sinhala, Singlish and Mix datasets and found the highest accurate models and save them.

3.8.4 Ensemble Models

Ensemble modeling is a process of combining multiple models and predict an outcome by using different modelling algorithms or training datasets. In this research we have combined four Supervised Learning models and deep neural network model for Sinhala, Singlish and Mix datasets separately. For an example the Sinhala dataset was trained using five separate individual algorithms and by considering the accuracy values of each model I have come up with the ensemble model for the Sinhala dataset. The mechanism to generate the ensemble model was discussed under 4.3.4 Ensemble Model section. Similarly that has been done for all other datasets (Singlish and Mix) with the intension of generating the ensemble models for each dataset. As the last step both Sinhala and Singlish ensemble models were concatenated as a Sinhala-Singlish ensemble model where it contain the predictions for the whole dataset same as the Mix ensemble model. So that it can be used to compare the results generated by these two ensemble models in order to select the accurate ensemble model which can give us the better prediction results.

3.9 Introduction to Evaluation

In the evaluation of proposed research approach to identify Hate Speech on Sinhala Language, following approaches are to be used.

By using developed trained models with preprocessed data can be evaluate/test to identify the highest accuracy models data contains hateful speech. Initially supervised learning models and deep neural network model evaluate to generate accuracy value for Count Vectors and TF-IDF variables for Sinhala, Singlish and Mix datasets separately. Then apply cross validation for high accuracy values generated from Count Vectors or TF-IDF variables. Then after that saved high accuracy cross validation models apply to the Ensemble Model and predict the results. Standard evaluation can be done using “unseen” test set and identify the detection and suggesting contents of hate speech. Furthermore, it will test the different approaches which are going to use within this research and based on that accuracy level can be measured. Moreover, as the earlier approaches confusion matrix used to construct precision, recall, F-Score of hate speech detection approach. Confusion Matrix is mainly used to describe the performance of the classification model of hate speech detection. Construct Precision, Recall, F-Score denote to find all relevant cases from the hate speech data set. Finally, we can evaluate the machine learning approaches which used to identify Hate Speech contents from the trained model and flag them.

3.10 Model Implementation for Hate Speech Detection

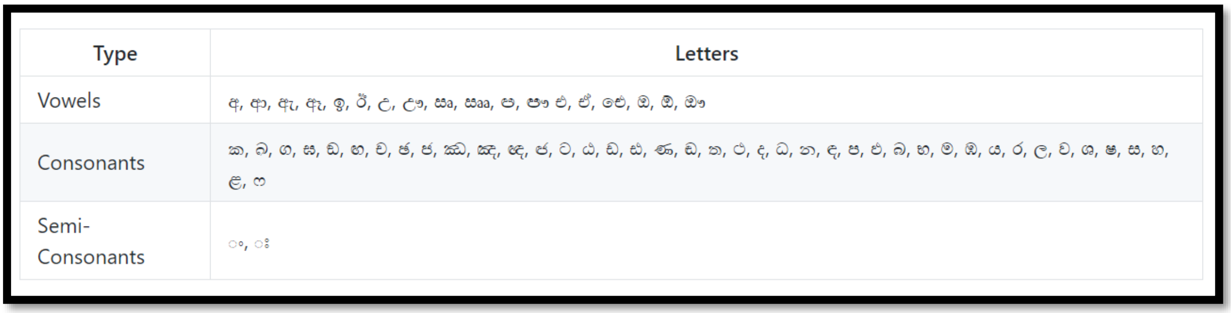
Implementation phase describe the detail process of construction of classification and deep learning models with the codes and technologies. The entire process will be described step by step. The main steps of this experiment includes,

- Data gathering and labeling
- Data preprocessing
- Feature extraction methods
- Build classification, and deep neural network model
- Performance evaluation (Cross Validation and Ensemble method)

Data collection and annotation process described in the beginning of the methodology. Therefore, the detailed process will be explained from the data preprocessing step.

3.10.1 Preprocessing

As the first step all gathered data should be cleaned before they are fed to the classifiers in order to reduce noise. For this task we used Natural Language Processing and developed methods used to carry out preprocess steps, in order to get the cleaned data set. Since the language is Sinhala, we have to have an idea about Sinhala character, vowels etc. Sinhala Language alphabet consists of 61 letters comprising 18 vowels, 41 consonants and 2 semi-consonants.



Type	Letters
Vowels	අ, ආ, ඇ, ඈ, ඉ, ඊ, උ, ඌ, ඍ, ඎ, ඏ, ඐ එ, ඒ, ඓ, ඔ, ඕ, ඖ
Consonants	ක, ඛ, ග, ඝ, ඞ, ඟ, ච, ඡ, ජ, ඣ, ඤ, ඥ, ඦ, ට, ඨ, ඩ, ඪ, ණ, ඬ, ත, ථ, ද, ධ, න, ඳ, ප, ඵ, බ, භ, ම, ඹ, ය, ර, ල, ව, ශ, ෂ, ස, හ, ළ, ෆ
Semi-Consonants	◌ං, ◌ඃ

Figure 3.4: Sinhala alphabet

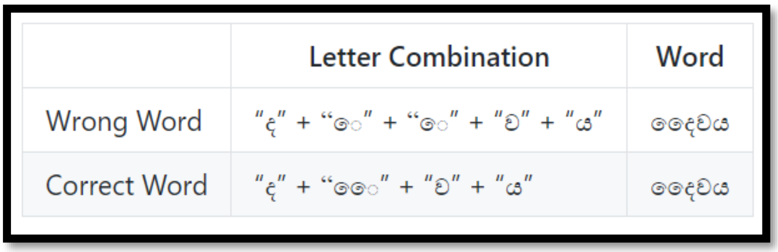
Data preprocessing was carried out commonly for Linear SVM, Logistic Regression, Naïve Bayes, Random Forest supervised learning models and deep neural network model. When cleaning the dataset following steps were followed.

- **Remove symbols**

Under this phase non alpha numeric characters, url, mentions, retweet status were removed and resulting data move to the tokenization. These special characters were removed by checking with the regular expressions using functions.

- **Fixing Vowels**

In order to overcome the mistakes of Sinhala typing, vowel letter fixer can be used. As the following example, although two words are same, computer identified it as two different words.



	Letter Combination	Word
Wrong Word	"ද" + "ෙ" + "ෙ" + "ව" + "ය"	දෙෙවය
Correct Word	"ද" + "ෛ" + "ව" + "ය"	දෛවය

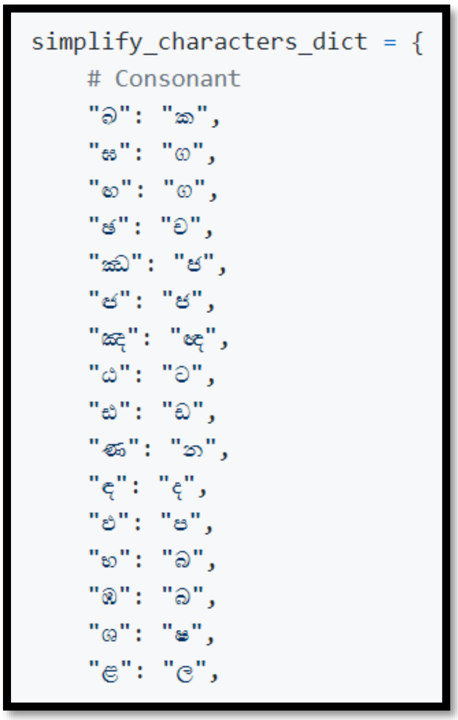
Figure 3.5: Vowel letter mistakes

This model checking the vowels in Sinhala characters and fixed the character issues and return the simplified sentences.

- **Simplifying Sinhalese Characters**

The textual contents in social networks are often informal, unstructured and even misspelled, but by simplifying characters, it is able to identify same word with different misspelled words.

As an example, the word "මුහුදට" can be mistakenly type instead of "මුහුදට". By using simplifying characters helps to identify those words are same.



```
simplify_characters_dict = {
    # Consonant
    "ඛ": "ක",
    "ඝ": "ග",
    "ඟ": "ග",
    "ඡ": "ච",
    "ඣ": "ජ",
    "ඦ": "ජ",
    "ඤ": "ඥ",
    "ඨ": "ට",
    "ඪ": "ඩ",
    "ණ": "න",
    "ඳ": "ද",
    "ඵ": "ප",
    "භ": "බ",
    "ඹ": "බ",
    "ශ": "ෂ",
    "ළ": "ල",
```

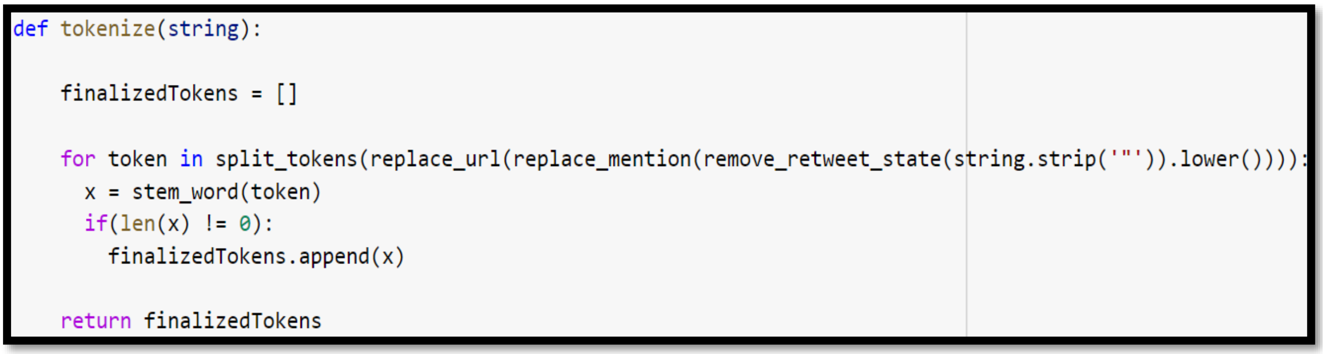
Figure 3.6: Simplified Character Dictionary

- **Stemming**

Stemming defines the process of reducing inflected words to their word stem, base or root form. According to the theory Sinhala data were stemmed by removing the suffixes.

- **Tokenization**

Breaking up strings into words and punctuations is known as tokenization. Words in every comment were tokenized.

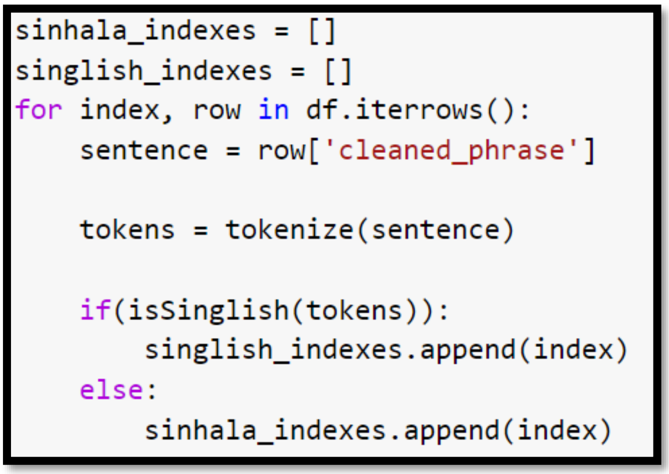


```
def tokenize(string):

    finalizedTokens = []

    for token in split_tokens(replace_url(replace_mention(remove_retweet_state(string.strip('"')).lower()))):
      x = stem_word(token)
      if(len(x) != 0):
        finalizedTokens.append(x)

    return finalizedTokens
```

Figure 3.7: Tokenize the data set

- **Generate unique Singlish token for English tokens**

In this step identify the English and Singlish words and generate unique Singlish token for the discovered English tokens. Then separate Sinhala and Singlish tokens and save the data sets in two separate CSV files by finding the indexes.

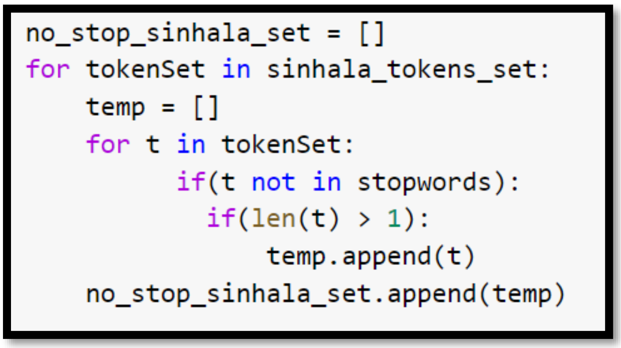


```
sinhala_indexes = []
singlish_indexes = []
for index, row in df.iterrows():
    sentence = row['cleaned_phrase']

    tokens = tokenize(sentence)

    if(isSinglish(tokens)):
        singlish_indexes.append(index)
    else:
        sinhala_indexes.append(index)
```

Figure 3.8: Find Indexes of Sinhala and Singlish data

- **Remove stop words**

In order to remove the stop words from the data set, separate Sinhala stop words list was used and got the cleaned data set in order to reconstruct the sentences and generate count tokens for each sentence.



```
no_stop_sinhala_set = []
for tokenSet in sinhala_tokens_set:
    temp = []
    for t in tokenSet:
          if(t not in stopwords):
            if(len(t) > 1):
                temp.append(t)
    no_stop_sinhala_set.append(temp)
```

Figure 3.9: Remove stop words

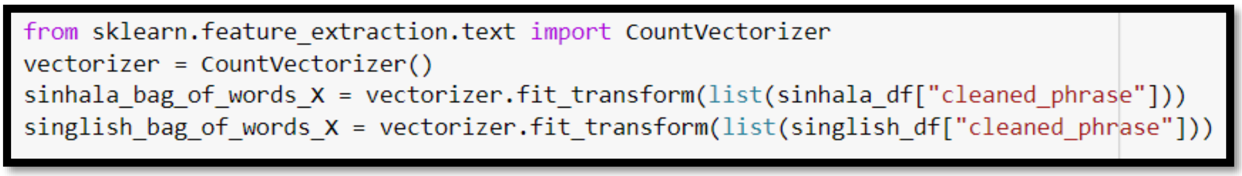
After the above mentioned steps cleaned data was saved on separate CSV files as Sinhala and Singlish data set. This task has been completed to clean the data set before applying to supervised learning model and deep neural network model.

3.10.2 Feature Extraction

In supervised learning approaches, entirely the extraction accomplishments were done with use of python scikit learn toolkit. The feature extraction codes rely on the functions of scikit learn toolkit. In Natural Language Processing feature extraction purposes, CountVectorizer and Tf-idf vectorizers are mainly and widely used as inbuilt vectorizers. Both vectorizer features have been tested for Sinhala, Singlish and Mix data set in each supervised learning model.

- **CountVectorizer – Bag of Word Features (BoW)**

CountVectorizer is going to transform the comment/sentence into an array having the count of appearances of each word in it. Before performing the feature extraction Split the data into training and testing parts. Furthermore, CountVectorizer implements both tokenization and occurrence counting in a single class. A collection of text documents can be converted to a matrix of token counts using CountVectorizer. This vector space model used to count the number of the unique words in all comments and the frequency of each term in vector can be observed. Therefore, bag-of-word features were extracted using CountVectorizer.

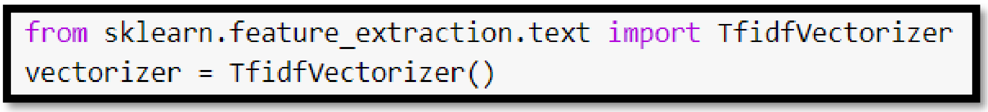


```
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer()
sinhala_bag_of_words_X = vectorizer.fit_transform(list(sinhala_df["cleaned_phrase"]))
singlish_bag_of_words_X = vectorizer.fit_transform(list(singlish_df["cleaned_phrase"]))
```

Figure 3.10: Count Vectorizer

- **Tf-idf Vectorizer – Term Frequency Features (Tf-idf)**

Term frequency-inverse document frequency vector is a way to measure the importance of a word or term and also measure how much rarely a word is present in a document. So, using this vectorizer, the words with highest importance as a feature can be obtained. Furthermore, Tf-idf is frequency of the term is off-set by the frequency of the word in the corpus which clearly says that some words appear more frequently in general.

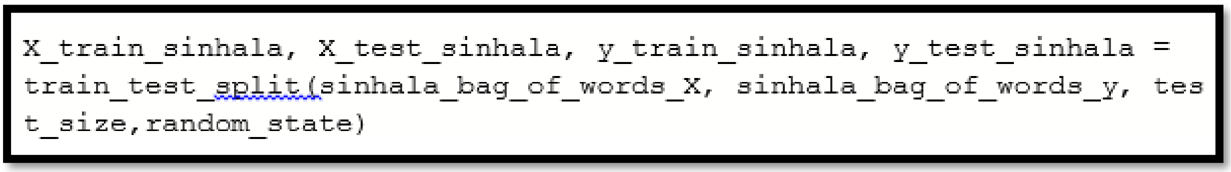


```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer()
```

Figure 3.11: Tf-idf Vectorizer

3.10.3 Build classification, and deep neural network model

As mentioned earlier in methodology, four supervised learning approaches and Deep Neural Network approach were used to detect hate speech on Sinhala language. Supervised learning approaches used CountVectorizer and TF-IDF vectorizer features for Sinhala, Singlish and Mix data sets separately. Then applied machine learning classifier for each approach using training dataset and evaluate using the testing dataset.



```
X_train_sinhala, X_test_sinhala, y_train_sinhala, y_test_sinhala =
train_test_split(sinhala_bag_of_words_X, sinhala_bag_of_words_y, tes
t_size,random_state)
```

Figure 3.12: Train Test Dataset Split

As mentioned in earlier in methodology, deep neural network approach used multiple layers to train the data set including embedding layer, LSTM, dense layers with ‘tanh’,’relu’ and ‘sigmoid’ activation functions. Furthermore, three folds and 10 epochs are used to train the dataset and save the best values on the disk. For all supervised learning models and deep neural network will be evaluated using precision, recall, F-score and confusion Metrix.

3.10.4 Performance evaluation (Cross Validation and Ensemble method)

The results generated from above supervised learning and deep neural network were compared and it was concluded that count vectorizer values generate higher values than TF-IDF values. Therefore, cross validation applied for count vectorizer results in supervised learning and deep neural network for Sinhala, Singlish and Mix datasets. Then after that generated results were compared and applied Ensemble model for highest accurate models from each approach by combining saved models. Ensemble model had been tested for Sinhala, Singlish data separately and Mix data separately. Then concatenate Sinhala and Singlish data again tested with mix data set. By using this method, we can find the highest accurate model to use for the hate speech detection.

Chapter 4: Results and Evaluation

In this chapter the evaluation plan, experiments and results of the experiments will be discussed. Major goal of this investigation is to inspect diverse algorithm behaviors accompanied by diverse features to discover the Sinhala hate speech and find the most accurate model. In this section dissimilar feature collections and dissimilar classifiers will be assessed with regard to 'accuracy', 'precision', 'recall' and 'F-score' measures.

4.1 Data Set involved with the Evaluation

To perform a successful experiment on hate speech detection availability of a labeled corpus is really important. As the first step, proper Sinhala data set was collected from Social Media besides, selected set consist of comments written and posted by the users from numerous posts on Twitter also from Sinhala newspaper article comments which are available publicly. They have given great opportunities to users/readers to express their ideas. Also, they have provided the policies in order to safe the platform. Since data set is prepared by using users' ideas, articles, etc. We got the proper data set to identify hate contents. Also, data set was preprocessed in order to remove special characters, stop words and other irrelevant content. The annotated data set used to feed to the model and results will be generated accordingly. Therefore, credibility of the data set and results from the machine learning approach will be higher.

4.2 Evaluation Approach

For the Hate speech detection evaluation approach, statistical, mathematical techniques can be used to accomplish the task. Since this research based on the quantitative approach, systematic investigation has done for it. Initially, manually collected the Sinhala data and annotated it manually and the result applied to build model in order to get the predictions. Then data set results generate from the model will be analyzed with the metric in order to measure the performance of the research.

Furthermore, the evaluation metric built was used throughout the experiment. This research based on natural language processing, machine learning and deep learning, therefore as the evaluation metrics accuracy, precision, recall and F-score were selected for this research. According to the research dataset, it was mentioned the data either hate or not, therefore binary classification method used for this research. The values checked from the above-mentioned metrics relied on the notation of positives and negatives. According to this research the comment with hate defined as positive and comment without hate defined as negative.

Table 4.1: Structure of the Confusion Matrix

		Predicted Class	
		With Hate	No Hate
True Class	With Hate	True Positive	False Positive
	No Hate	False Negative	True Negative

According to the Table 4.1, True negatives, true positives, false negatives and false positives are defined in the structure of Confusion Matrix. Also, in order to evaluate the results, the confusion matrix was built according to the above table and results will be observed based on that. As the evaluation metrics for supervised learning and Deep Neural Network approaches as well as cross validation followed the below metrics and generate the values accordingly.

- **Accuracy**

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

According to the above-mentioned formula, accuracy can be defined as the fraction of predictions that are correct. In natural language processing researches, accuracy is used for the evaluation, but it has few problems which are very common. Accuracy is not a good measure when the classes of data is unbalanced. Most of the datasets are not 100% balanced. Furthermore, accuracy measure gives more weight to the correctly classified positives and negatives. In data set unbalanced situations the results of the accuracy rate will be higher in one class.

- **Precision**

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

Precision defines fraction of predicted hate comments which were actually hate comments. This method is useful to measure the correctness of the positive predictions. Therefore precision is the best method to check the correctly predicted positives because it does not consider about negatives.

- **Recall**

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

Recall defines the fraction of hate comments that were detected. By using this measure the number of hate comments can be identified and number of hate comments that the classifier missed can be examined. Recall also does not consider about true negatives. Therefore, this is also a better solution for the evaluation of this research.

- **F-Score**

$$F - score = \frac{\text{2 x Precision x Recall}}{\text{Precision} + \text{Recall}}$$

When harmonic mean of precision and recall is calculated it is called as F-score. This measure ensures that there will be no overly rely on either precision or recall, therefore F-score was selected as the main evaluation measure for hate speech detection.

4.3 Results

As mentioned in the Methodology chapter all the comments were stored in one csv file and read into one data frame and then split into two data sets as Sinhala and Singlish data. Then apply the supervised learning models and deep neural network model for the above mentioned data set. Then cross validation has been done for the above mentioned algorithms and the best models used to develop an Ensemble model.

4.3.1 Training and Testing Data Set

This approach has been done for both Sinhala, Singlish and Mix data. Initially data set divided into two sets call Sinhala dataset and Singlish dataset through the preprocessing. Then for both datasets divide into training and testing sets through the model parameters as follows.

Table 4.2: Total Dataset Summary

Total data	3500
No of Hate Comments	1319
No of comments without Hate	2276
Training dataset size	0.9
Testing dataset size	0.1

Table 4.3: Sinhala and Singlish dataset

Total Sinhala data	2340
Total Singlish data	1255
Training dataset size	0.9
Testing dataset size	0.1

Then apply the supervised learning models for both Sinhala and Singlish datasets separately and concatenated dataset and then generate the Bag of Words feature values.

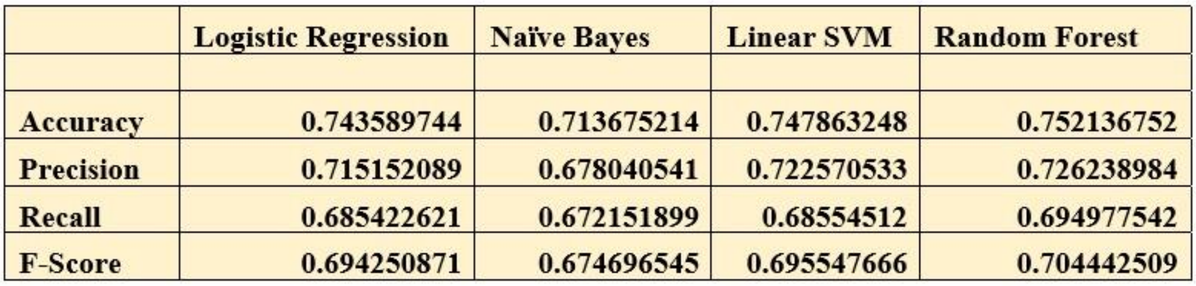
4.3.2 Bag of Words Features and Deep Neural Network

All the bag of words features is extracted using CountVectorizer and Tf-Idf vectorizer in Scikit-learn package. Count Vectorizer applied for both Sinhala and Singlish datasets separately and for the merged datasets as well.

- **Count Vectorizer Results**

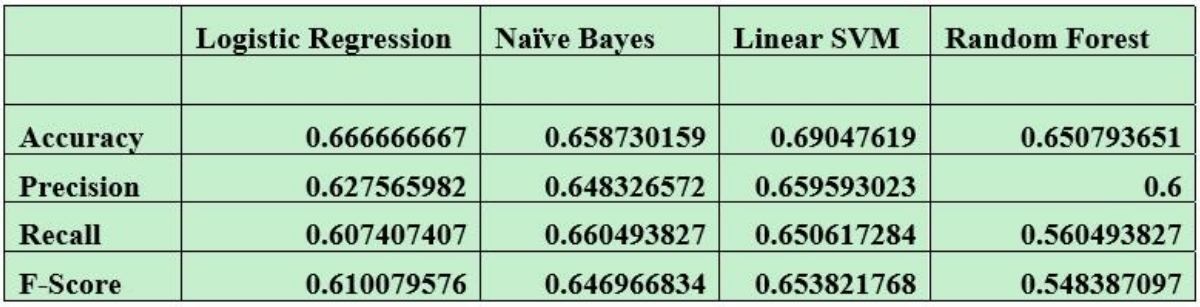
Count Vectorizer results generated from four classifiers will be explained in this section.

Table 4.4: Count Vectorizer - Sinhala data set



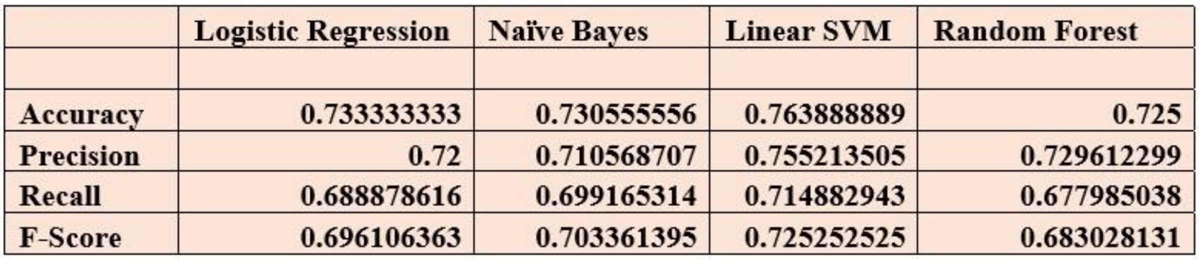
	Logistic Regression	Naïve Bayes	Linear SVM	Random Forest
Accuracy	0.743589744	0.713675214	0.747863248	0.752136752
Precision	0.715152089	0.678040541	0.722570533	0.726238984
Recall	0.685422621	0.672151899	0.68554512	0.694977542
F-Score	0.694250871	0.674696545	0.695547666	0.704442509

Table 4.5: Count Vectorizer - Singlish data set



	Logistic Regression	Naïve Bayes	Linear SVM	Random Forest
Accuracy	0.666666667	0.658730159	0.69047619	0.650793651
Precision	0.627565982	0.648326572	0.659593023	0.6
Recall	0.607407407	0.660493827	0.650617284	0.560493827
F-Score	0.610079576	0.646966834	0.653821768	0.548387097

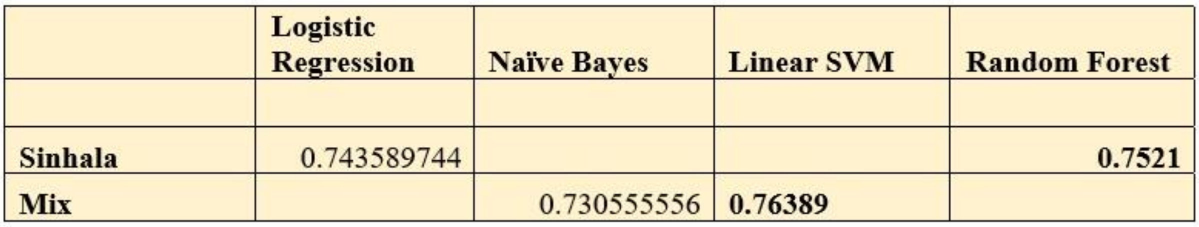
Table 4.6: Count Vectorizer - Mix data set



	Logistic Regression	Naïve Bayes	Linear SVM	Random Forest
Accuracy	0.733333333	0.730555556	0.763888889	0.725
Precision	0.72	0.710568707	0.755213505	0.729612299
Recall	0.688878616	0.699165314	0.714882943	0.677985038
F-Score	0.696106363	0.703361395	0.725252525	0.683028131

According to the Count Vectorizer results, Random Forest classifier contains the highest value of Sinhala data set. Linear SVM contains the highest value in Singlish data set as well as Mix data set. The table 4.7 displayed the summary result of Count Vectorizer in each classifier model.

Table 4.7: Summary of count vectorizer



	Logistic Regression	Naïve Bayes	Linear SVM	Random Forest
Sinhala	0.743589744			**0.7521**
Mix		0.730555556	**0.76389**	

According to the count vectorizer results Logistic Regression contains higher accuracy value for Sinhala data set, Naïve Bayes contains higher accuracy value in mix data set, Linear SVM contains highest value in mix data set and Random Forest contains highest value in Sinhala data set. Furthermore, Linear SVM is having higher accuracy value and F-score in Count Vectorizer.

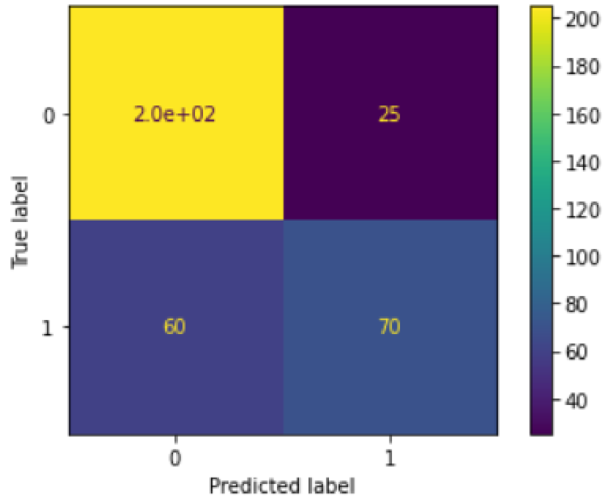
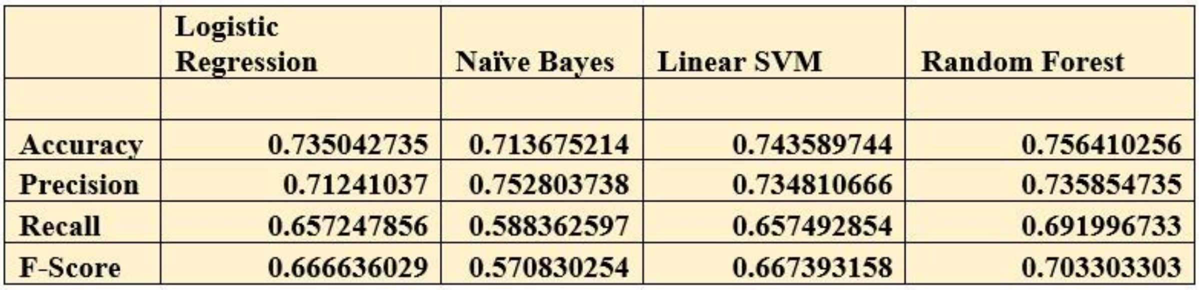


Figure 4.1: Confusion Matrix of Linear SVM

- **TF-IDF Vectorizer Results**

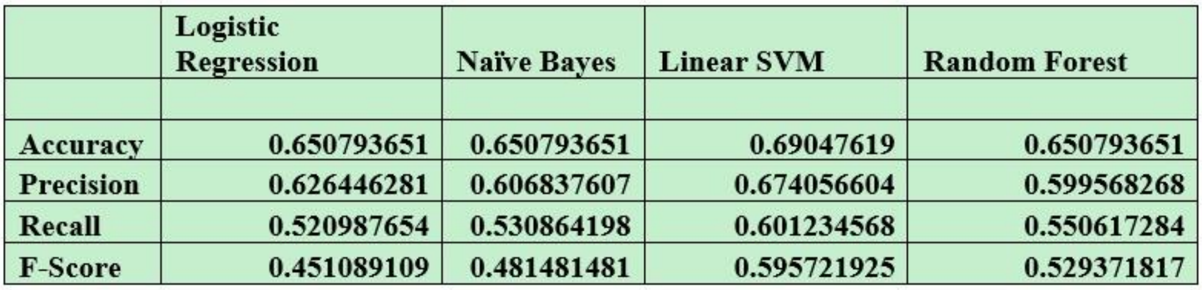
In this section the results generated from TF-IDF vectorizer will be explained.

Table 4.8: TF-IDF Vectorizer - Sinhala data set



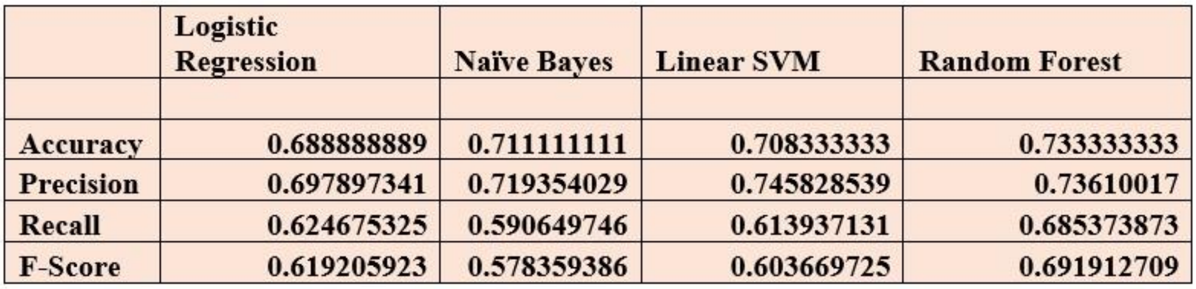
	Logistic Regression	Naïve Bayes	Linear SVM	Random Forest
Accuracy	0.735042735	0.713675214	0.743589744	0.756410256
Precision	0.71241037	0.752803738	0.734810666	0.735854735
Recall	0.657247856	0.588362597	0.657492854	0.691996733
F-Score	0.666636029	0.570830254	0.667393158	0.703303303

Table 4.9: TF-IDF Vectorizer - Singlish data set



	Logistic Regression	Naïve Bayes	Linear SVM	Random Forest
Accuracy	0.650793651	0.650793651	0.69047619	0.650793651
Precision	0.626446281	0.606837607	0.674056604	0.599568268
Recall	0.520987654	0.530864198	0.601234568	0.550617284
F-Score	0.451089109	0.481481481	0.595721925	0.529371817

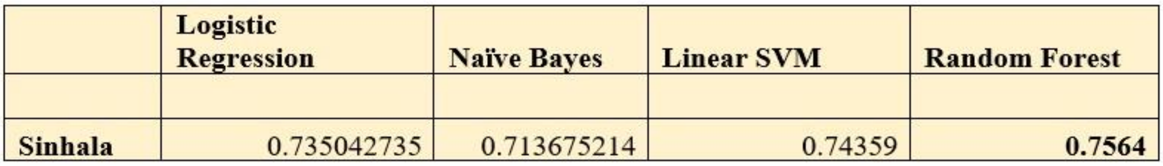
Table 4.10: TF-IDF Vectorizer - Mix data set



	Logistic Regression	Naïve Bayes	Linear SVM	Random Forest
Accuracy	0.688888889	0.711111111	0.708333333	0.733333333
Precision	0.697897341	0.719354029	0.745828539	0.73610017
Recall	0.624675325	0.590649746	0.613937131	0.685373873
F-Score	0.619205923	0.578359386	0.603669725	0.691912709

According to the TF-IDF results as shown in above tables displayed that Random Forest accuracy value is higher in Sinhala dataset. In Singlish data set Linear SVM generated the highest value and Mix data set Random Forest classifier contains the highest value. Table 4.11 shown the summary result of TF-IDF vectorizer.

Table 2.11: Summary of TF-IDF vectorizer



	Logistic Regression	Naïve Bayes	Linear SVM	Random Forest
Sinhala	0.735042735	0.713675214	0.74359	**0.7564**

According to the Table 4.11 TF-IDF vectorizer generated higher value in Sinhala data set in all four supervised learning models. Random Forest is having fairly higher value with the F-score but Linear SVM also have very small difference when compare with Random Forest.

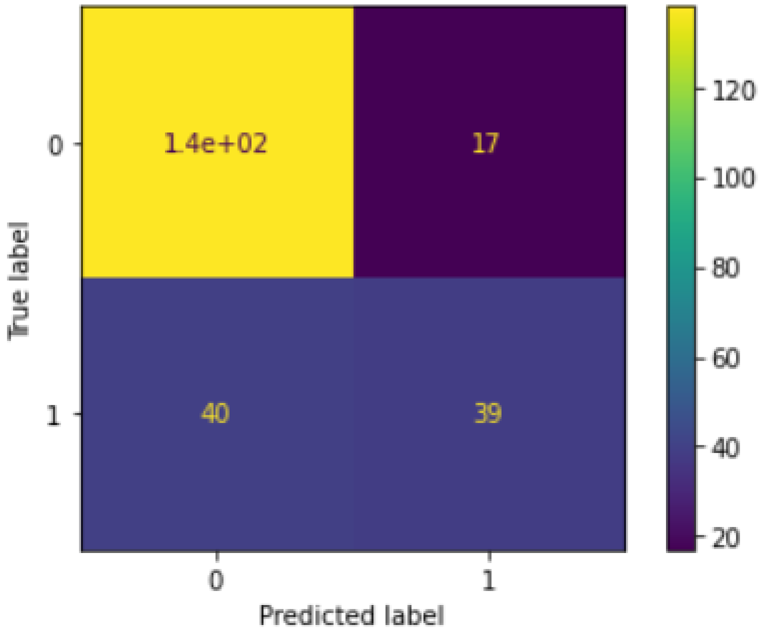


Figure 4.2: Random Forest Confusion Matrix for TF-IDF

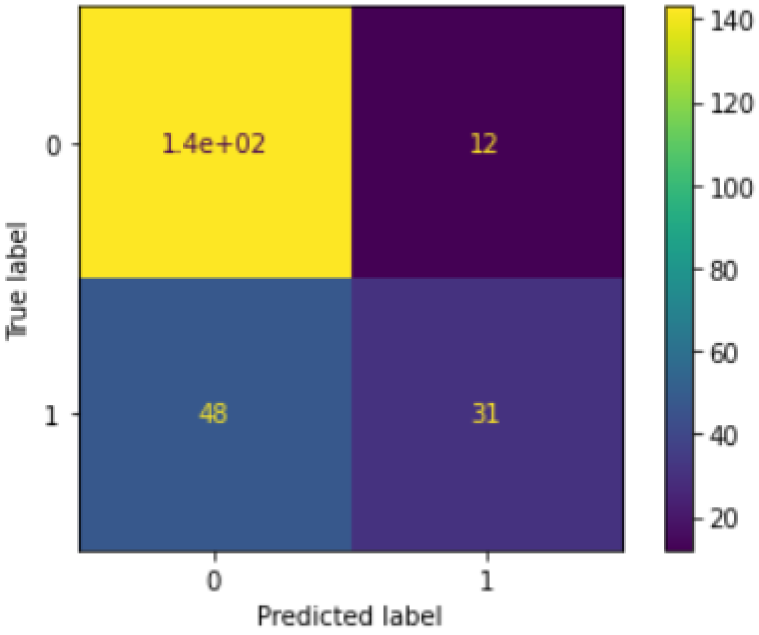


Figure 4.3: Linear SVM confusion matrix for TF-IDF

According to the Count Vectorizer and TF-IDF Vectorizer values, Count Vectorizer generates fairly higher values. Therefore, as for the Cross Validation approach we used the results generated from Count Vectorizer.

- **Deep Neural Network Results**

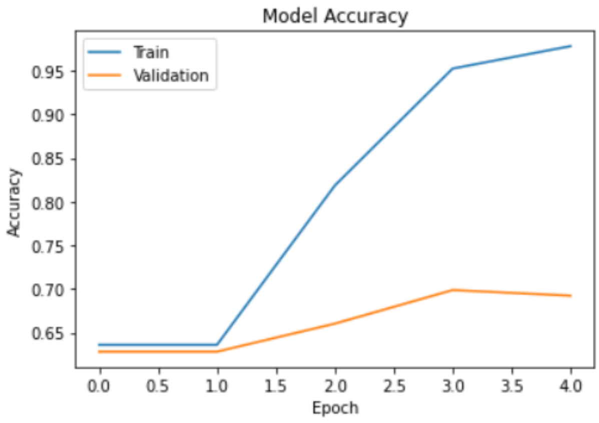
In LSTM model three-fold cross validation has been followed and calculated the Accuracy values for Sinhala, Singlish and Mix datasets.

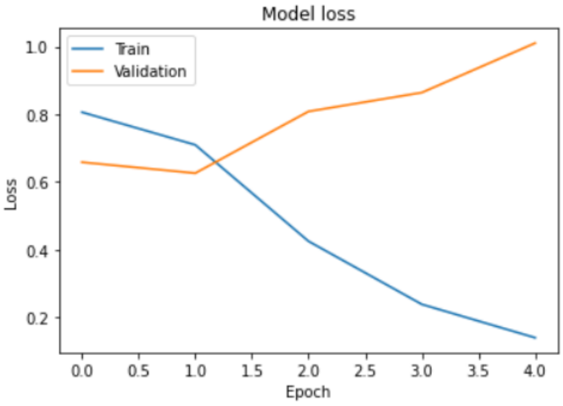
Table 4.12: Accuracy summary in Deep Neural Network

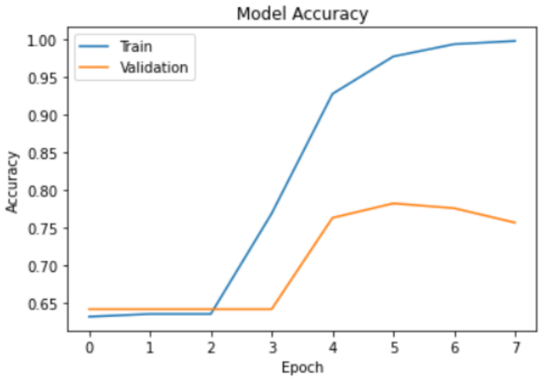
	Sinhala	Singlish	Mix
Accuracy	0.782051	0.595238	0.758333

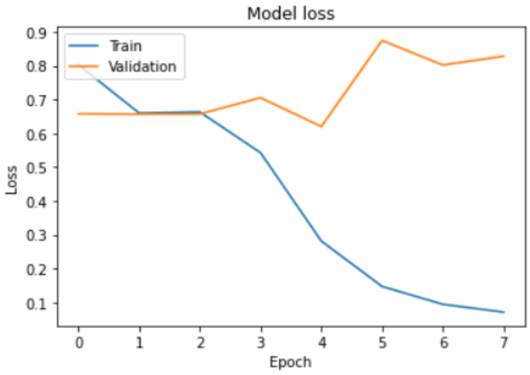
According to the Table 4.12 Sinhala data set contains the highest accuracy value. Following results has been displayed for the Sinhala Dataset.

Table 4.13: Summary of Fold Accuracy and Loss in Deep Neural Network

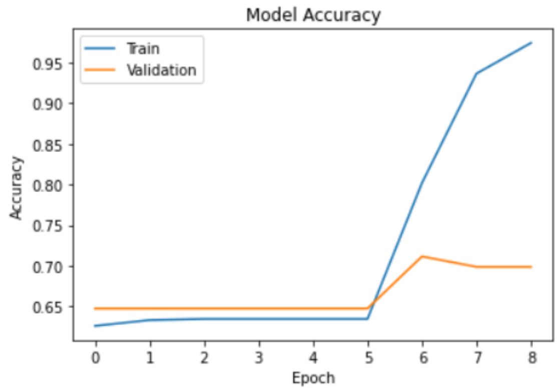
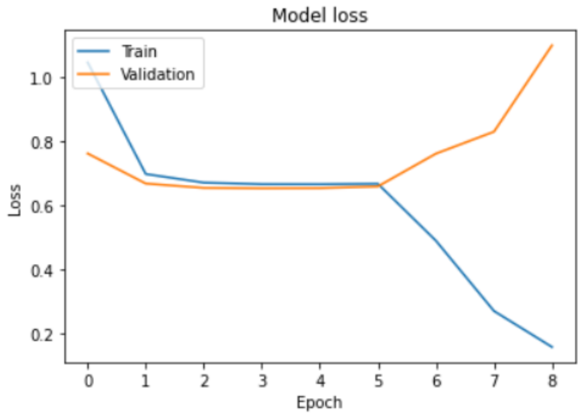
<table>
<tr><th>Fold</th><th>Accuracy</th><th>Loss</th></tr>
<tr><td>0</td><td>


</td><td>


</td></tr>
<tr><td>1</td><td>


</td><td>


</td></tr>
</table>

2		

Following table 4.13 shows the results of Confusion Metrix generate for the three folds.

Table 4.14: Confusion Metrix result in Deep Neural Network

Fold	**Confusion Metrix**			
0			**True Class**	
			No	**Yes**
	Predicted Class	**No**	496	284
		Yes	0	0
1			True Class	
			No	**Yes**
	Predicted Class	**No**	364	86
		Yes	132	198
2			**True Class**	
			No	**Yes**
	Predicted Class	**No**	495	285
		Yes	0	0

In LSTM model calculated the precision, recall and F1 score according to the confusion matrix of each fold of the model in Sinhala Dataset.

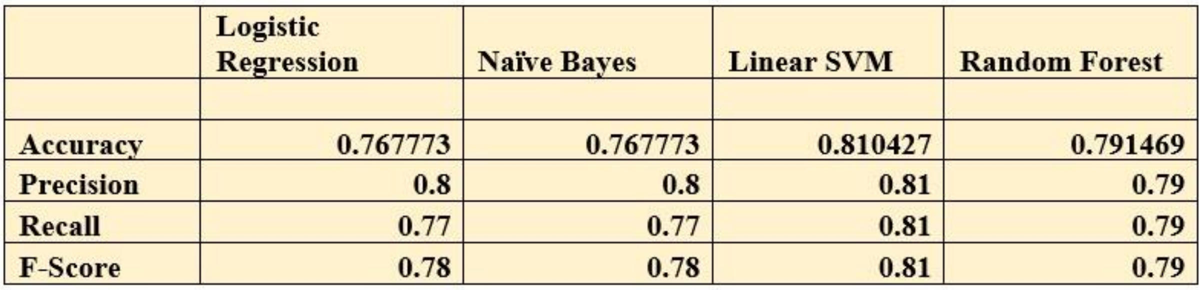
Table 4.15: Precision, Recall, F1-Score values in Deep Neural Network for Sinhala Dataset

Fold	**Class**	**Precision**	**Recall**	**F1 score**
0	No	0.6359	1.0	0.7774
	Yes	INFINITE	0.0	NaN
1	No	0.8089	0.7339	0.7696
	Yes	0.6	0.6972	0.645
2	No	0.6346	1.0	0.7765
	Yes	INFINITE	0.0	NaN

4.3.3 Cross Validation

Following results shows the cross validation of classifiers including Logistic Regression, Naïve Bayes, Linear SVM and Random Forest.

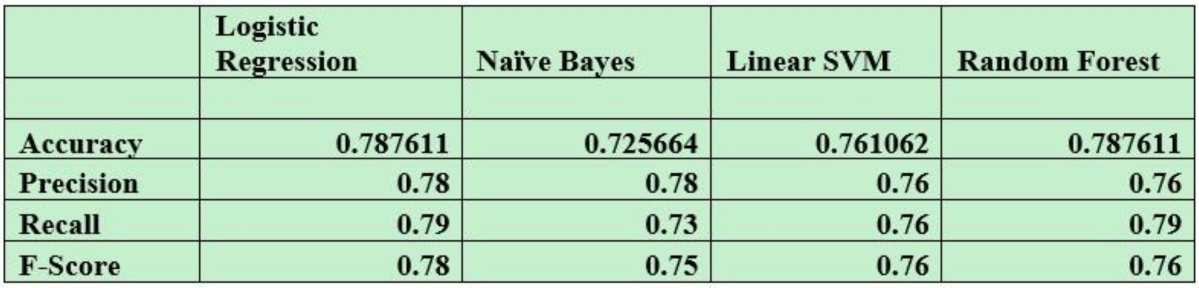
Table 4.16: Cross Validation for Sinhala Dataset



	Logistic Regression	**Naïve Bayes**	**Linear SVM**	**Random Forest**
Accuracy	**0.767773**	**0.767773**	**0.810427**	**0.791469**
Precision	**0.8**	**0.8**	**0.81**	**0.79**
Recall	**0.77**	**0.77**	**0.81**	**0.79**
F-Score	**0.78**	**0.78**	**0.81**	**0.79**

According to the Table 4.16 Linear SVM generated the highest accuracy, F-score values in Sinhala data corpus.

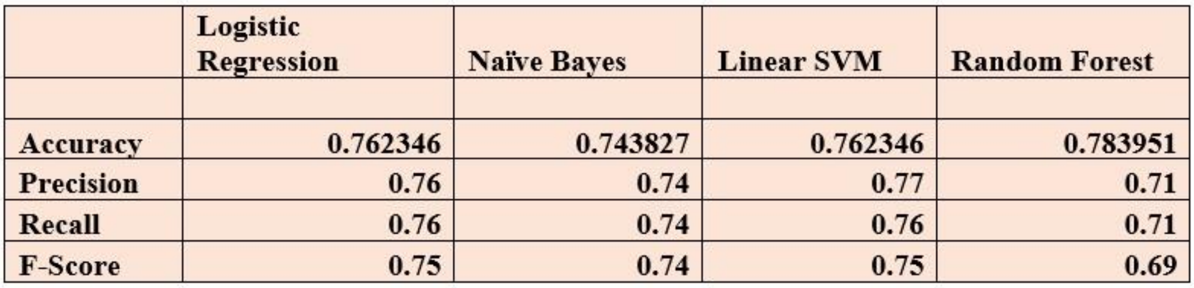
Table 4.17: Cross Validation for Singlish Dataset



	Logistic Regression	**Naïve Bayes**	**Linear SVM**	**Random Forest**
Accuracy	**0.787611**	**0.725664**	**0.761062**	**0.787611**
Precision	**0.78**	**0.78**	**0.76**	**0.76**
Recall	**0.79**	**0.73**	**0.76**	**0.79**
F-Score	**0.78**	**0.75**	**0.76**	**0.76**

According to the Table 4.17 Logistic Regression generated the highest accuracy, F-score values in Singlish data corpus.

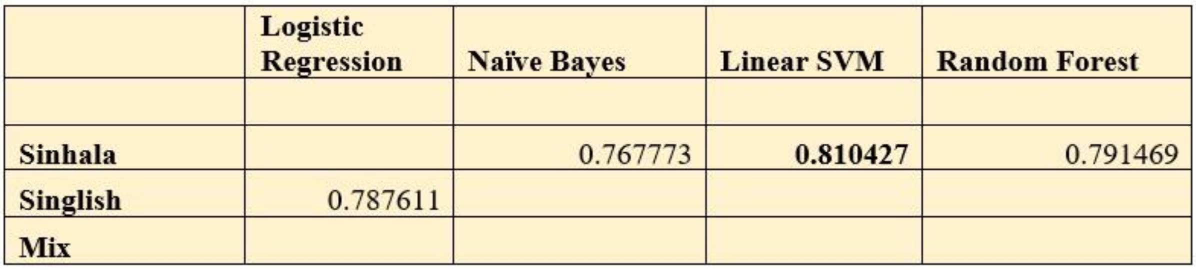
Table 4.18: Cross Validation for Mix Dataset



	Logistic Regression	Naïve Bayes	Linear SVM	Random Forest
Accuracy	**0.762346**	**0.743827**	**0.762346**	**0.783951**
Precision	**0.76**	**0.74**	**0.77**	**0.71**
Recall	**0.76**	**0.74**	**0.76**	**0.71**
F-Score	**0.75**	**0.74**	**0.75**	**0.69**

According to the Table 4.18 Random Forest generated the highest accuracy, F-score values in Mix data corpus. When compare with Logistic Regression and Linear SVM, the accuracy values are fairly similar.

Table 4.19: Cross validation summary



	Logistic Regression	Naïve Bayes	Linear SVM	Random Forest
Sinhala		0.767773	**0.810427**	0.791469
Singlish	0.787611			
Mix				

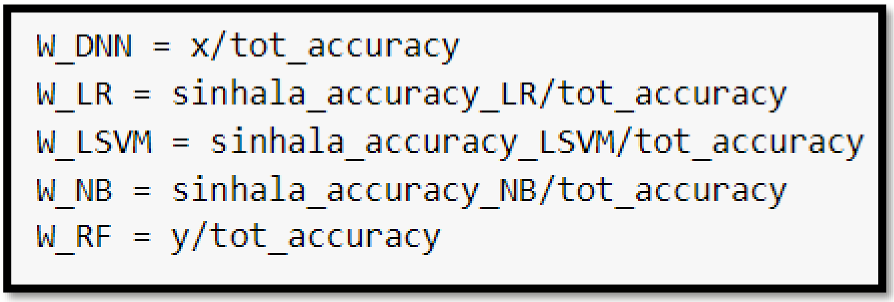
According to the cross validation summary table 4.19, conclude that Linear SVM generates fairly higher value than other supervised learning models.

4.3.4 Ensemble Model

Ensemble model built using the highest accuracy cross validation models generated from each supervised learning model and deep neural network model combination. Therefore, five models used to generate ensemble model results for Sinhala, Singlish and Mix datasets separately. Then after that concatenate with divided Sinhala and Singlish data set and compared with the mix data set results. For the ensemble method used the separate testing data with 286 rows to get the accurate model and it was also preprocessed using the same mechanism.

In this model, first loaded the saved models. This approach also tested for the Sinhala, Singlish and Mix data sets separately. By using preprocessed test data continue the task to generate prediction values. Then generate the accuracy values for classifier models and deep neural network model.

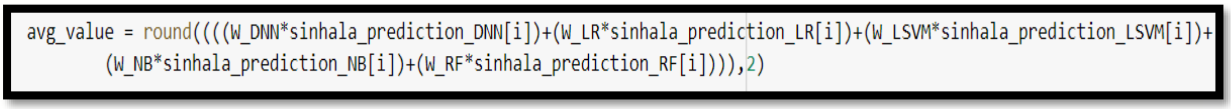
In order to develop the ensemble model, we need to get the summation of the classifier models accuracy and deep neural network accuracy and based on these accuracy values calculate weights for each model.



```
W_DNN = x/tot_accuracy
W_LR = sinhala_accuracy_LR/tot_accuracy
W_LSVM = sinhala_accuracy_LSVM/tot_accuracy
W_NB = sinhala_accuracy_NB/tot_accuracy
W_RF = y/tot_accuracy
```

Figure 4.4: Ensemble model weight calculation

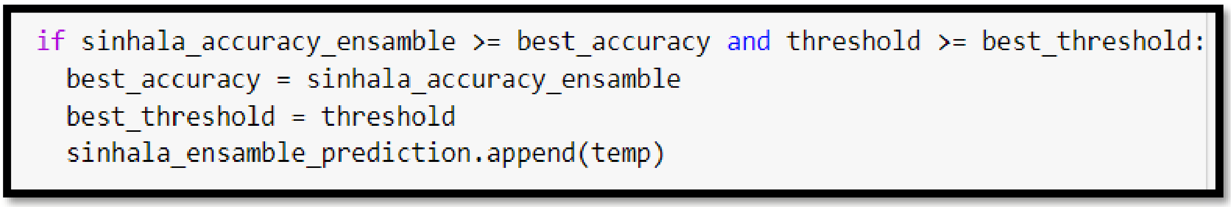
Then using calculated weights and prediction values, calculate the average value and compare it with the threshold value.



```
avg_value = round((((W_DNN*sinhala_prediction_DNN[i])+(W_LR*sinhala_prediction_LR[i])+(W_LSVM*sinhala_prediction_LSVM[i])+
        (W_NB*sinhala_prediction_NB[i])+(W_RF*sinhala_prediction_RF[i]))),2)
```

Figure 4.5: Ensemble model Average calculation

As the next step, in order to find the best accuracy and threshold values, compare the generated ensemble accuracy with the best accuracy and threshold value and calculate best values for the model. This task has been carried out for Sinhala, Singlish, Mix data sets separately and aggregated the Sinhala-Singlish data set.



```
if sinhala_accuracy_ensamble >= best_accuracy and threshold >= best_threshold:
  best_accuracy = sinhala_accuracy_ensamble
  best_threshold = threshold
  sinhala_ensamble_prediction.append(temp)
```

Figure 4.6: Ensemble model best accuracy and threshold calculation

Table 4.20: Individual Model Prediction Results

Sinhala Accuracy Values	**Singlish Accuracy Values**	**Mix Accuracy Values**
sinhala_accuracy_DNN 0.480000	singlish_accuracy_DNN 0.570000	Mix_accuracy_DNN 0.500000
sinhala_accuracy_LR 0.500000	singlish_accuracy_LR 0.560000	Mix_accuracy_LR 0.480000

sinhala_accuracy_LSVM **0.520000 (Best Accuracy)**	singlish_accuracy_LSVM 0.570000	Mix_accuracy_LSVM 0.520000
sinhala_accuracy_NB 0.500000	singlish_accuracy_NB **0.620000 (Best Accuracy)**	Mix_accuracy_NB **0.540000 (Best Accuracy)**
sinhala_accuracy_RF 0.490000	singlish_accuracy_RF 0.570000	Mix_accuracy_RF 0.500000

Table 4.21: Ensemble Model Prediction Results

Sinhala Data set	Ensamble method best accuracy **:0.530000,** best threshold value :0.180000
Singlish Data set	Ensamble method best accuracy **:0.620000,** best threshold value :0.670000
Mix Data Set	Ensamble method best accuracy **:0.530000,** best threshold value :0.590000
Concatenate best sinhala and singlish predictions	**ensamble_sinhala_singlish_accuracy :0.550000**

According to the ensemble prediction data we can conclude that Sinhala-Singlish concatenated data set provides better accuracy than the mix model accuracy.

Chapter 5: Future work and Conclusions

5.1 Conclusions

The main aim of this research is to find an approach to detect hate speech on Sinhala Language. Therefore, four machine learning (supervised learning) algorithms and deep neural network approach were used to generate the results. Then to increase the performance of the above mentioned classifiers cross validations techniques were used to accomplish the highest validation models. As the next step, developed the Ensemble model with combining all four classifiers and deep neural network highest accuracy models as well as checking different combinations for Sinhala, Singlish and Mix datasets. Then after that combined the Sinhala-Singlish data sets again and compared with the Mix data set. As the summary we can conclude that Ensemble Sinhala-Singlish accuracy is better than Mix model accuracy.

Furthermore, it is appropriate to consider an approach with combined models instead of using models separately, because it provides the better result. Instead of using Sinhala, Singlish as separate two models it is better to aggregate divided data sets and compare it with the Mix model which helps to increase the performance.

The main objective of this research was to develop an approach to detect the contents of hate speech in Sinhala Language in Social Media context. In order to accomplish this task few other objectives were followed. One of the sub objective was to collect a representative sample of Sinhala/Singlish hate speech from social media and annotate it. Therefore, we have collected around 3500 data set and annotated it manually. The preprocessing of data set gave the noise free context in order to proceed with the next stage.

The other objective was to explore algorithms that would help in training a suitable model from the training data and perform model diagnostics in order to validate the model. Therefore, we have used four supervised learning models, deep neural network model, cross validation models and developed an ensemble model with all above mentioned classifiers and deep neural network model to get the more accurate results.

Finally, accuracy values and F-scores were compared to detect the performance of discovered algorithms. Initially Count Vectorizer and TF-IDF Vectorizer results compared and Count Vector results generated fairly higher values. Also Linear SVM had the highest accuracy value. Then Count Vectorizer selected for the cross validation and in this phase also Linear SVM contained the fairly higher value. Next we developed the Ensemble model with generated highest value models and we can conclude that Sinhala-Singlish aggregated model gives the higher value than Mix Model results.

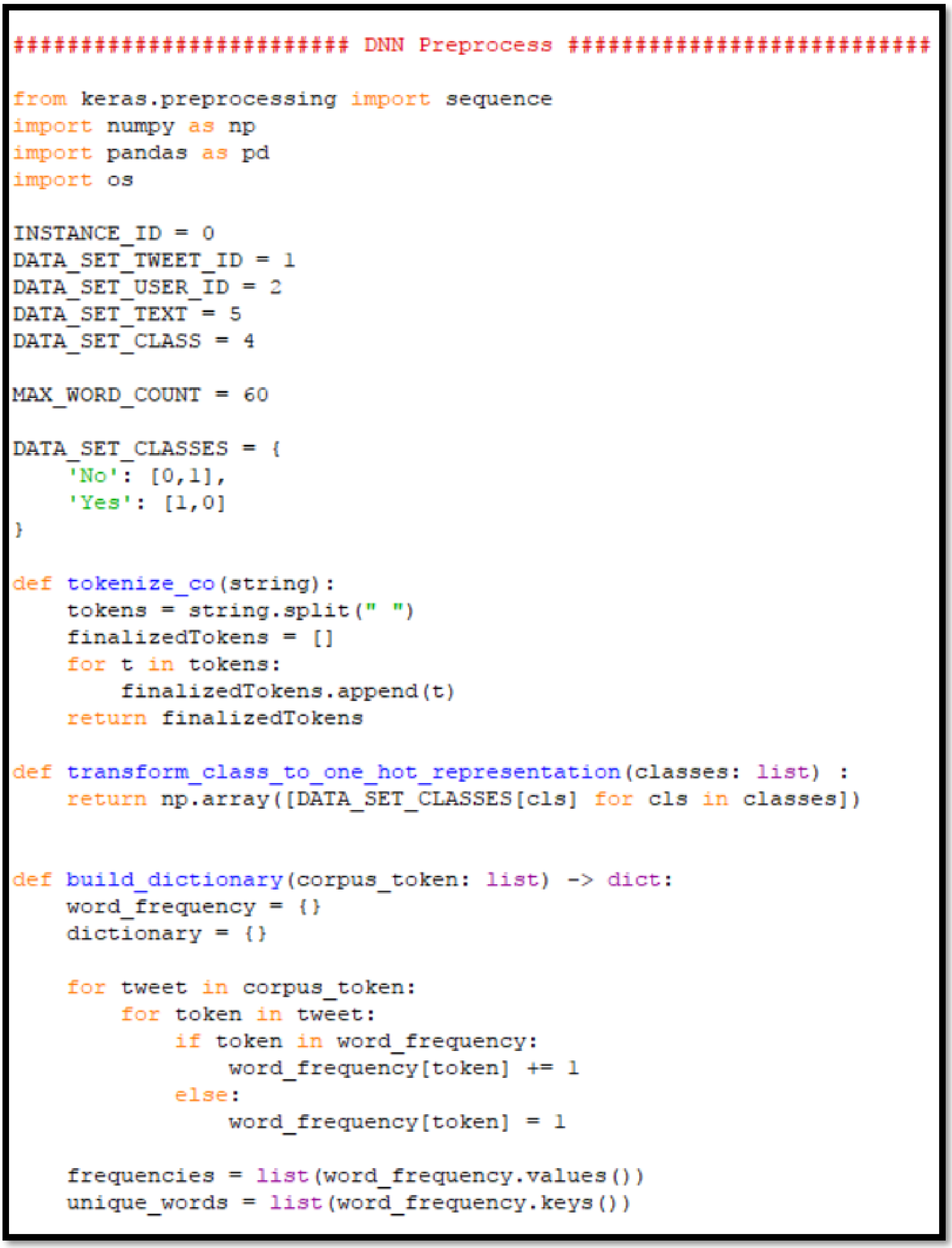
5.2 Future work

This research is mainly focused to develop an approach to detect hate speech on Sinhala Language. In this research we used supervised learning classifiers and deep neural network approach to build the Ensemble model and predict the results. As the future work this research can be enhanced with unsupervised learning models to predict the results.

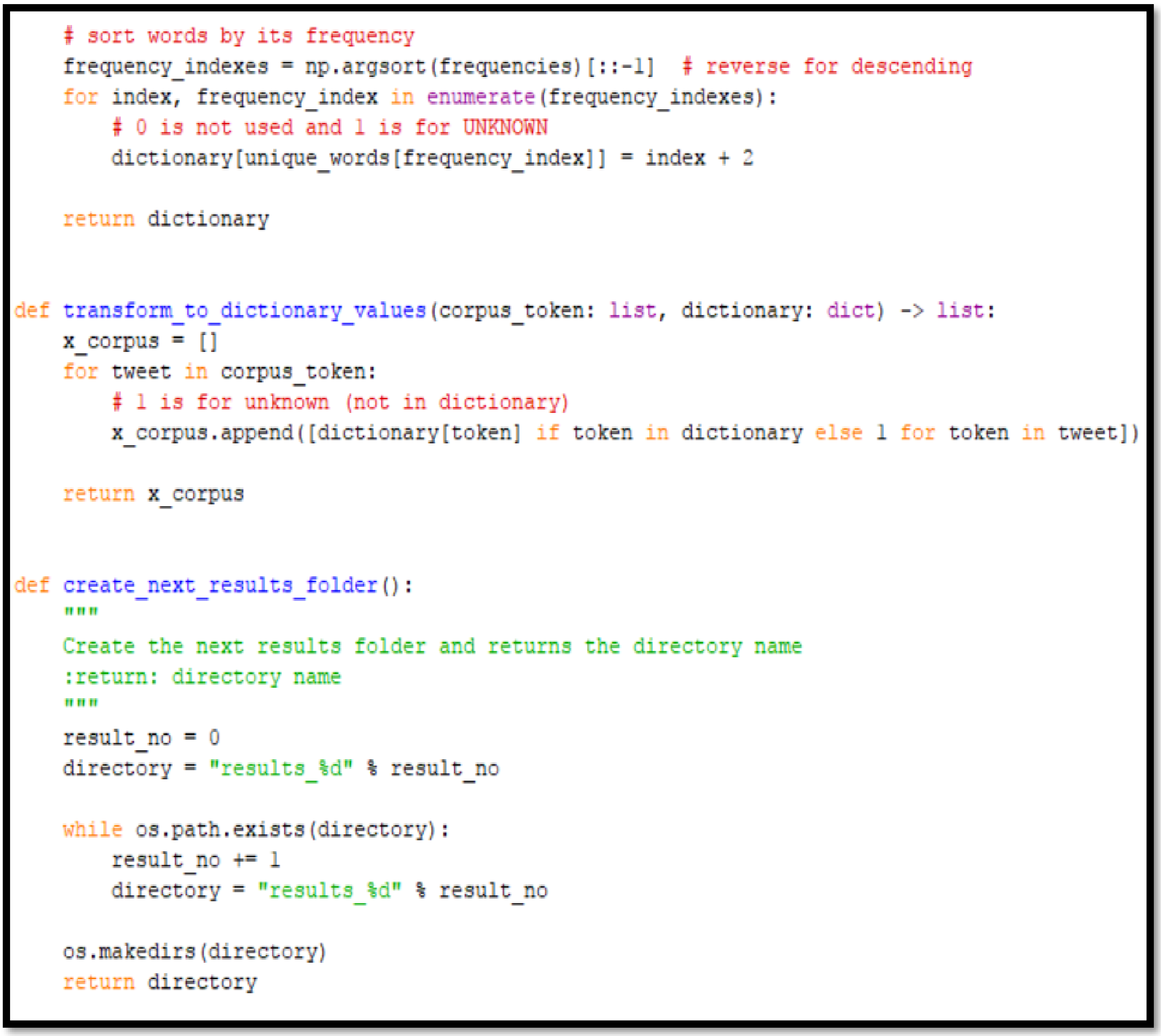
Furthermore, this can be enhanced with increasing the data corpus and annotate the dataset with correct labels. The semi-supervised learning can be used to annotate the dataset as well as training the models. As the manual annotation task is having difficulties, if there is a proper approach available, it would be easier for the preparation of the dataset to predict the accurate results.

APPENDICES

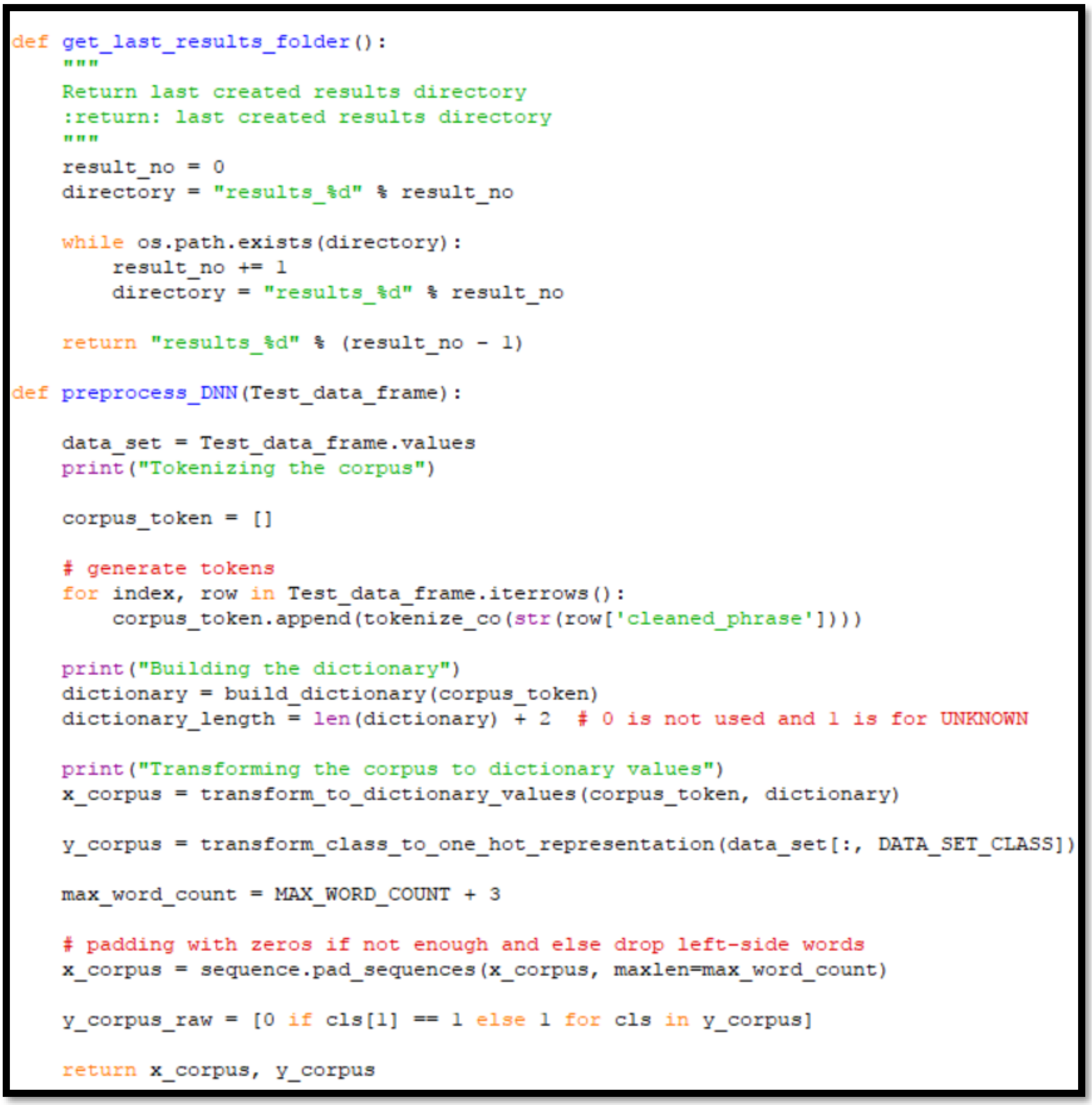
Test Data Preprocessing in Ensemble Model (Testdata_Preprocess.py)



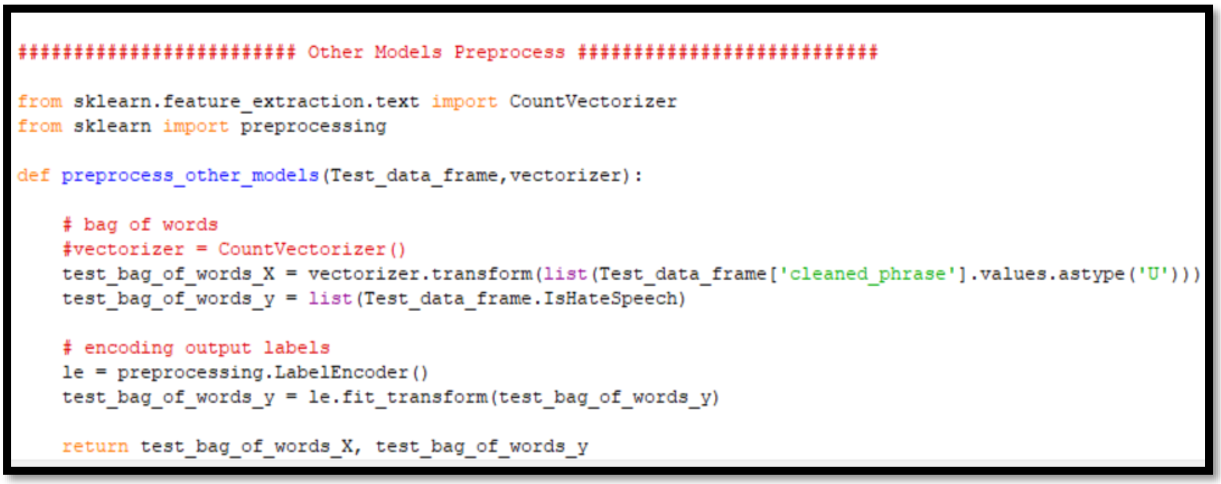
```
######################### DNN Preprocess ###########################

from keras.preprocessing import sequence
import numpy as np
import pandas as pd
import os

INSTANCE_ID = 0
DATA_SET_TWEET_ID = 1
DATA_SET_USER_ID = 2
DATA_SET_TEXT = 5
DATA_SET_CLASS = 4

MAX_WORD_COUNT = 60

DATA_SET_CLASSES = {
    'No': [0,1],
    'Yes': [1,0]
}

def tokenize_co(string):
    tokens = string.split(" ")
    finalizedTokens = []
    for t in tokens:
        finalizedTokens.append(t)
    return finalizedTokens

def transform_class_to_one_hot_representation(classes: list) :
    return np.array([DATA_SET_CLASSES[cls] for cls in classes])


def build_dictionary(corpus_token: list) -> dict:
    word_frequency = {}
    dictionary = {}

    for tweet in corpus_token:
        for token in tweet:
            if token in word_frequency:
                word_frequency[token] += 1
            else:
                word_frequency[token] = 1

    frequencies = list(word_frequency.values())
    unique_words = list(word_frequency.keys())
```



```
    # sort words by its frequency
    frequency_indexes = np.argsort(frequencies)[::-1]  # reverse for descending
    for index, frequency_index in enumerate(frequency_indexes):
        # 0 is not used and 1 is for UNKNOWN
        dictionary[unique_words[frequency_index]] = index + 2

    return dictionary


def transform_to_dictionary_values(corpus_token: list, dictionary: dict) -> list:
    x_corpus = []
    for tweet in corpus_token:
        # 1 is for unknown (not in dictionary)
        x_corpus.append([dictionary[token] if token in dictionary else 1 for token in tweet])

    return x_corpus


def create_next_results_folder():
    """
    Create the next results folder and returns the directory name
    :return: directory name
    """
    result_no = 0
    directory = "results_%d" % result_no

    while os.path.exists(directory):
        result_no += 1
        directory = "results_%d" % result_no

    os.makedirs(directory)
    return directory
```

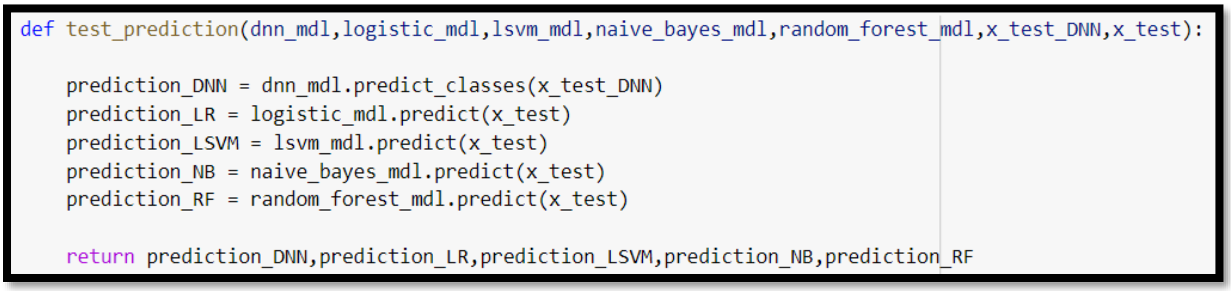


```
def get_last_results_folder():
    """
    Return last created results directory
    :return: last created results directory
    """
    result_no = 0
    directory = "results_%d" % result_no

    while os.path.exists(directory):
        result_no += 1
        directory = "results_%d" % result_no

    return "results_%d" % (result_no - 1)

def preprocess_DNN(Test_data_frame):

    data_set = Test_data_frame.values
    print("Tokenizing the corpus")

    corpus_token = []

    # generate tokens
    for index, row in Test_data_frame.iterrows():
        corpus_token.append(tokenize_co(str(row['cleaned_phrase'])))

    print("Building the dictionary")
    dictionary = build_dictionary(corpus_token)
    dictionary_length = len(dictionary) + 2  # 0 is not used and 1 is for UNKNOWN

    print("Transforming the corpus to dictionary values")
    x_corpus = transform_to_dictionary_values(corpus_token, dictionary)

    y_corpus = transform_class_to_one_hot_representation(data_set[:, DATA_SET_CLASS])

    max_word_count = MAX_WORD_COUNT + 3

    # padding with zeros if not enough and else drop left-side words
    x_corpus = sequence.pad_sequences(x_corpus, maxlen=max_word_count)

    y_corpus_raw = [0 if cls[1] == 1 else 1 for cls in y_corpus]

    return x_corpus, y_corpus
```



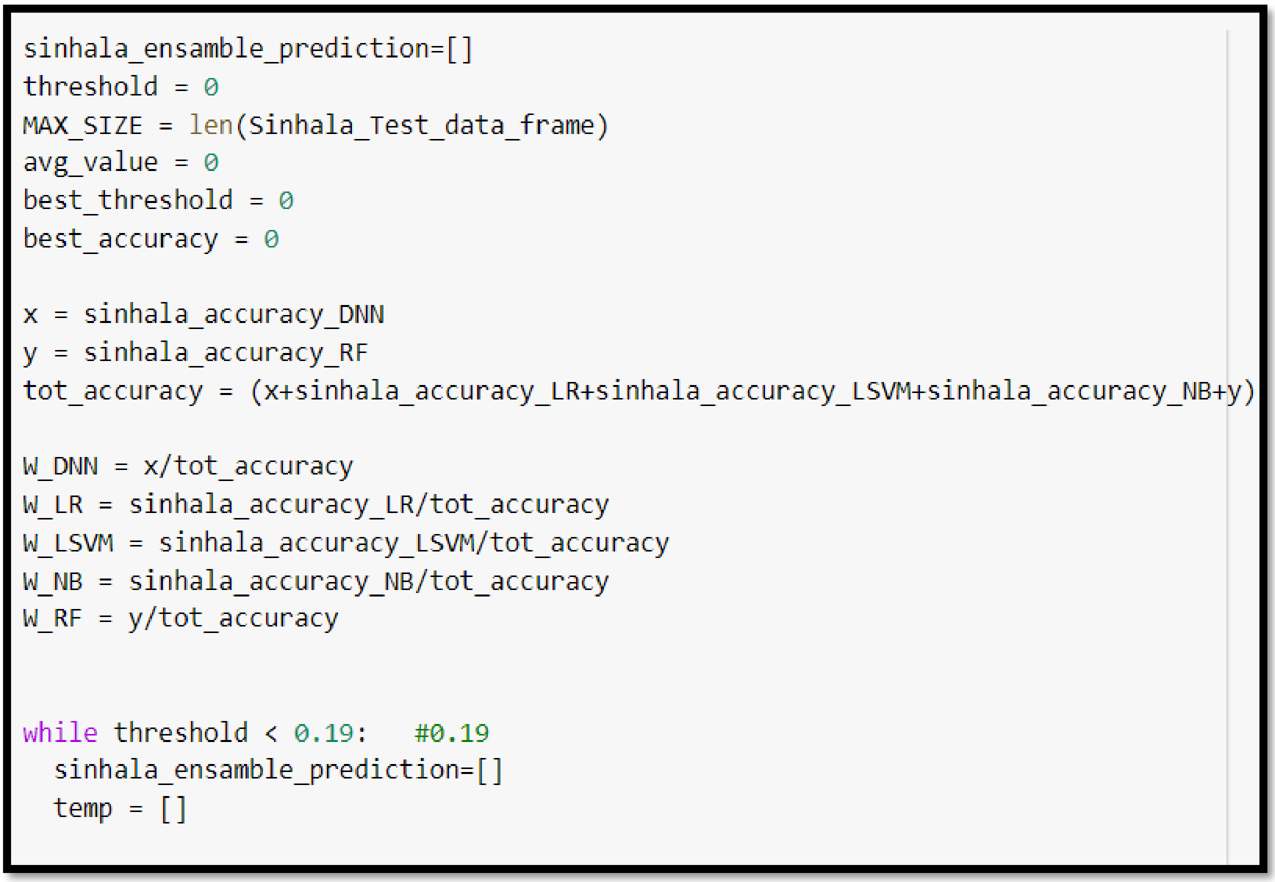
```
######################### Other Models Preprocess ###########################

from sklearn.feature_extraction.text import CountVectorizer
from sklearn import preprocessing

def preprocess_other_models(Test_data_frame,vectorizer):

    # bag of words
    #vectorizer = CountVectorizer()
    test_bag_of_words_X = vectorizer.transform(list(Test_data_frame['cleaned_phrase'].values.astype('U')))
    test_bag_of_words_y = list(Test_data_frame.IsHateSpeech)

    # encoding output labels
    le = preprocessing.LabelEncoder()
    test_bag_of_words_y = le.fit_transform(test_bag_of_words_y)

    return test_bag_of_words_X, test_bag_of_words_y
```

Test prediction method of Ensemble Model

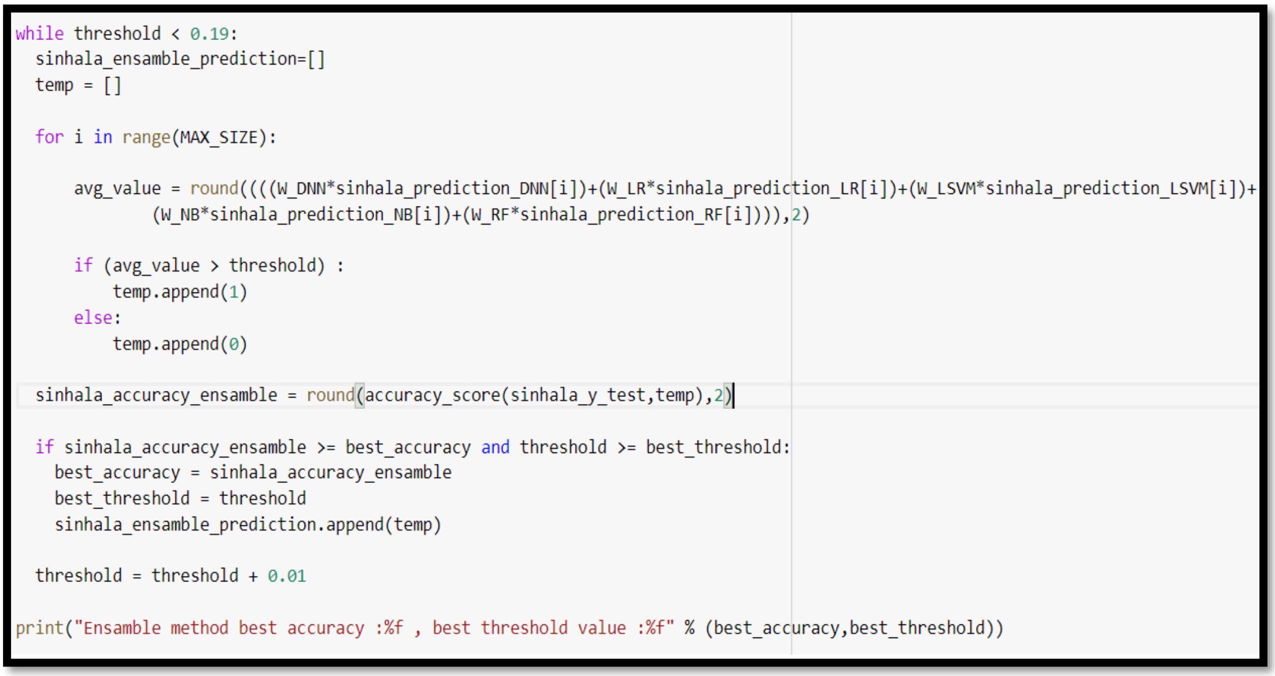


```
def test_prediction(dnn_mdl,logistic_mdl,lsvm_mdl,naive_bayes_mdl,random_forest_mdl,x_test_DNN,x_test):

    prediction_DNN = dnn_mdl.predict_classes(x_test_DNN)
    prediction_LR = logistic_mdl.predict(x_test)
    prediction_LSVM = lsvm_mdl.predict(x_test)
    prediction_NB = naive_bayes_mdl.predict(x_test)
    prediction_RF = random_forest_mdl.predict(x_test)

    return prediction_DNN,prediction_LR,prediction_LSVM,prediction_NB,prediction_RF
```

Ensemble model for Sinhala dataset



```
sinhala_ensamble_prediction=[]
threshold = 0
MAX_SIZE = len(Sinhala_Test_data_frame)
avg_value = 0
best_threshold = 0
best_accuracy = 0

x = sinhala_accuracy_DNN
y = sinhala_accuracy_RF
tot_accuracy = (x+sinhala_accuracy_LR+sinhala_accuracy_LSVM+sinhala_accuracy_NB+y)

W_DNN = x/tot_accuracy
W_LR = sinhala_accuracy_LR/tot_accuracy
W_LSVM = sinhala_accuracy_LSVM/tot_accuracy
W_NB = sinhala_accuracy_NB/tot_accuracy
W_RF = y/tot_accuracy


while threshold < 0.19:   #0.19
  sinhala_ensamble_prediction=[]
  temp = []
```



```
while threshold < 0.19:
  sinhala_ensamble_prediction=[]
  temp = []

  for i in range(MAX_SIZE):

      avg_value = round((((W_DNN*sinhala_prediction_DNN[i])+(W_LR*sinhala_prediction_LR[i])+(W_LSVM*sinhala_prediction_LSVM[i])+
              (W_NB*sinhala_prediction_NB[i])+(W_RF*sinhala_prediction_RF[i]))),2)

      if (avg_value > threshold) :
          temp.append(1)
      else:
          temp.append(0)

  sinhala_accuracy_ensamble = round(accuracy_score(sinhala_y_test,temp),2)

  if sinhala_accuracy_ensamble >= best_accuracy and threshold >= best_threshold:
    best_accuracy = sinhala_accuracy_ensamble
    best_threshold = threshold
    sinhala_ensamble_prediction.append(temp)

  threshold = threshold + 0.01

print("Ensamble method best accuracy :%f , best threshold value :%f" % (best_accuracy,best_threshold))
```

REFERENCES

[1] H. Watanabe, M. Bouazizi, and T. Ohtsuki, “Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection,” *IEEE Access*, vol. 6, pp. 13825–13835, 2018, doi: 10.1109/ACCESS.2018.2806394.

[2] I. Alfina, R. Mulia, M. I. Fanany, and Y. Ekanata, “Hate speech detection in the Indonesian language: A dataset and preliminary study,” in *2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS),* Oct. 2017, pp. 233–238, doi: 10.1109/ICACSIS.2017.8355039.

[3] A. Briliani, B. Irawan, and C. Setianingsih, “Hate Speech Detection in Indonesian Language on Instagram Comment Section Using K-Nearest Neighbor Classification Method,” in *2019 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS),* Nov. 2019, pp. 98–104, doi: 10.1109/IoTaIS47347.2019.8980398.

[4] H. M. S. T. Sandaruwan, S. A. S. Lorensuhewa, and M. A. L. Kalyani, “Sinhala Hate Speech Detection in Social Media using Text Mining and Machine learning,” in *2019 19th International Conference on Advances in ICT for Emerging Regions (ICTer)*, Sep. 2019, vol. 250, pp. 1–8, doi: 10.1109/ICTer48817.2019.9023655.

[5] B. Raufi and I. Xhaferri, “Application of Machine Learning Techniques for Hate Speech Detection in Mobile Applications,” in *2018 International Conference on Information Technologies (InfoTech),* Varna, Sep. 2018, pp. 1–4, doi: 10.1109/InfoTech.2018.8510738.

[6] G. K. Pitsilis, H. Ramampiaro, and H. Langseth, “Detecting Offensive Language in Tweets Using Deep Learning,” *Appl. Intell.*, vol. 48, no. 12, pp. 4730–4742, Dec. 2018, doi: 10.1007/s10489-018-1242-y.

[7] P. S. Br Ginting, B. Irawan, and C. Setianingsih, “Hate Speech Detection on Twitter Using Multinomial Logistic Regression Classification Method,” in *2019 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS),* Nov. 2019, pp. 105–111, doi: 10.1109/IoTaIS47347.2019.8980379.

[8] K. Nugroho *et al.*, “Improving Random Forest Method to Detect Hatespeech and Offensive Word,” in *2019 International Conference on Information and Communications Technology (ICOIACT),* Jul. 2019, pp. 514–518, doi: 10.1109/ICOIACT46704.2019.8938451.

[9] N. D. T. Ruwandika and A. R. Weerasinghe, “Identification of Hate Speech in Social Media,” in *2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer),* Colombo, Sri Lanka, Sep. 2018, pp. 273–278, doi: 10.1109/ICTER.2018.8615517.

[10] T. Davidson, D. Warmsley, M. Macy, and I. Weber, “Automated Hate Speech Detection and the Problem of Offensive Language,” p. 5.

[11] M. Petrocchi and M. Tesconi, “Hate me, hate me not: Hate speech detection on Facebook,” p. 10.

[12] “The Commonwealth Scientific and Industrial Research Organisation,” *Curr. Biol.*, vol. 7, no. 3, p. R126, Mar. 1997, doi: 10.1016/S0960-9822(97)70976-X.

[13] P. Mathur, R. Shah, R. Sawhney, and D. Mahata, “Detecting Offensive Tweets in Hindi-English Code-Switched Language,” in *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, Melbourne, Australia, 2018, pp. 18–26, doi: 10.18653/v1/W18-3504.

[14] T. L. Sutejo and D. P. Lestari, “Indonesia Hate Speech Detection Using Deep Learning,” in *2018 International Conference on Asian Language Processing (IALP),* Bandung, Indonesia, Nov. 2018, pp. 39–43, doi: 10.1109/IALP.2018.8629154.

[15] A. S. Saksesi, M. Nasrun, and C. Setianingsih, “Analysis Text of Hate Speech Detection Using Recurrent Neural Network,” in *2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC),* Bandung, Indonesia, Dec. 2018, pp. 242–248, doi: 10.1109/ICCEREC.2018.8712104.

[16] S. Zimmerman, C. Fox, and U. Kruschwitz, “Improving Hate Speech Detection with Deep Learning Ensembles,” p. 8.

[17] B. R. Amrutha and K. R. Bindu, “Detecting Hate Speech in Tweets Using Different Deep Neural Network Architectures,” in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, May 2019, pp. 923–926, doi: 10.1109/ICCS45141.2019.9065763.