

# Sentiment Analysis of Afaan Oromoo Facebook Media Using Deep Learning Approach

Megersa Oljira Rase  
Institute of Technology, Ambo University, PO box 19, Ambo, Ethiopia

## Abstract

The rapid development and popularity of social media and social networks provide people with-unprecedented opportunities to express and share their thoughts, views, opinions and feelings about almost anything through their personal webpages and blogs or using social network sites like Facebook, Twitter, and Blogger. This study focuses on sentiment analysis of social media content because automatically identifying and classifying opinions from social media posts can provide significant economic values and social benefits. The major problem with sentiment analysis of social media posts is that it is extremely vast, fragmented, unorganized and unstructured. Nevertheless, many organizations and individuals are highly interested to know what other peoples are thinking or feeling about their services and products. Therefore, sentiment analysis has increasingly become a major area of research interest in the field of Natural Language Processing and Text Mining. In general, sentiment analysis is the process of automatically identifying and categorizing opinions in order to determine whether the writer's attitude towards a particular entity is positive or negative. To the best of the researcher's knowledge, there is no Deep learning approach done for Afaan Oromoo Sentiment analysis to identify the opinion of the people on social media content. Therefore, in this study, we focused on investigating Convolutional Neural Network and Long Short Term Memory deep learning approaches for the development of sentiment analysis of Afaan Oromoo social media content such as Facebook posts comments. To this end, a total of 1452 comments collected from the official site of the Facebook page of Oromo Democratic Party/ODP for the study. After collecting the data, manual annotation is undertaken. Preprocessing, normalization, tokenization, stop word removal of the sentence are performed. We used the Keras deep learning python library to implement both deep learning algorithms. Long Short Term Memory and Convolutional Neural Network, we used word embedding as a feature. We conducted our experiment on the selected classifiers. For classifiers, we used 80% training and 20% testing rule. According to the experiment, the result shows that Convolutional Neural Network achieves the accuracy of 89%. The Long Short Memory achieves accuracy of 87.6%. Even though the result is promising there are still challenges.

**Keywords:** Sentiment Analysis; Opinionated Afaan Oromoo facebook comments; Oromo Democratic Party Facebook page

**DOI:** 10.7176/NMMC/90-02

**Publication date:** May 31<sup>st</sup> 2020

## 1. Introduction

The revolution of web2.0 and the increasing numbers of blogs, social media networks, web reviews, and many others have fundamentally changed the way people express their opinions and share information on the Internet. Due to the rapid development and popularity of social media networks, a huge amount of user-generated content, have been made available online. Identifying and determining whether the opinion of user generated content as positive or negative has become essential for different businesses and social entities as it is important for service providers and vendors to create successful marketing strategies and for the understanding of areas of improvement in products and services (Liu, 2012). Sentiment analysis is also important for tracking political opinions and politicians to understanding their social image, etc. (Bakliwal, et al., 2013).

People are able to express their opinions in form of posts, comments, tweets (Twitter), emoticons etc. with regard to many issues that affect their day to day lives (Vinodhini & Chandrasekaran, 2012). These online comments or opinions can be about several topics like government, organizations, products, politics, and many others. Since sentiment analysis can influence the interest of different parties such as customers, companies, and governments, organizations are highly interested in analyzing and exploring online opinions. While several commercial companies are interested to know the opinion of the public with respects to their products and services, many government organizations are interested to know the public feedback with respect to the new policy, rules and regulations set out as well as public services delivered.

Before the expansion of the internet and web2.0 technology, manual surveys had been used as the main method for answering the question of what do people think about some of the major economic and social events. Careful sampling of the surveyed population and a standardized questionnaire has been the standard way of learning about large groups of people. Now a day, the era of wide-spread internet access and social media has brought a new way of learning about large populations.

Therefore, collection and analysis of opinions have become easier because individuals share their views about different topics through social networks such as Facebook, Twitter, or they leave comments and reviews regarding

topics on a particular website such as web reviews (Akinyi, 2014).

Analysis of opinions plays an important role in all science areas (politics, economics, and social life) (Liu, 2012). For example, in marketing, if the sellers know about the customer's satisfaction with a particular product they may predict demand on the service or product. The same for politicians, they will know whether people support them or not, the government also understands the will of the society on their policy.

Sentiment analysis is mainly concerned with the analysis of emotions and opinions from the text. We can also refer to sentiment analysis as opinion mining to understand the outlook of a speaker or a writer with respect to some subject. The outlook may be their judgment or evaluation, their affective state or the envisioned emotional communication (George and Joseph, 2014). Most popular classification of sentiment: positive or negative (binary classification) (Chiavetta, et al., 2016).

The evolution of the Internet and of social media texts, such as Twitter, YouTube, and Facebook messages, has created many new opportunities for people to express the attitudes towards any individual, product, services, organizations, political parties, etc. on their own language. So that the amount of Afaan Oromoo documents on the web is also increasing. However, there are very few works proposed on Afaan Oromoo sentiment analysis. Sentiment analysis is not a new field and it has been researched since 2000. The previous works are focus on the English language. But in Afaan Oromoo and other under-resourced languages, this field is new and it is at the initial state.

Sentiment analysis can be conducted at different levels. The most famous level of sentiment analysis is document level, aspect level, and sentence level. *The document level* deals with determining the overall opinion of the document expressed by the opinion holder (Liu, 2012). This level of analysis assumes that each document expresses opinions on a single entity (e.g., a single product egg. review) (Pang, et al., 2002). The aspect level sentiment analysis is the task of extracting the relevant aspects of the reviewed product or entity and determining the sentiment of the corresponding opinion about them. The assumption of the aspect level sentiment analysis is that all opinions are generally directed at a specific topic/object (Popescu and Etzioni, 2005). Sentence level sentiment analysis is the more fine-grained analysis of the document. In this, polarity is calculated for each sentence as each sentence is considered a separate unit and each sentence can have different opinions. The core task in sentence-level sentiment analysis is subjectivity classification (Liu, 2012). The subjective sentence is the one that has polarity information that can be classified as positive or negative. Whereas objective sentence is the sentence that has factual information and it is classified as neutral or no opinion (Liu, 2012). For example: "The Nokia phone is very good!" The sentence is the subjective sentence as the sentences consist of opinionated words. Whereas the objective sentence has no opinionated words. *For example*, "Nokia is the product of Nokia Company" this sentence holds factual information and it has no sentiment orientation.

In this study, we considered the document-level sentiment analysis assumes that a document expresses a single sentiment. This approach suitable for some areas like reviews, where a last statement about the entity is assumed to be required which is a weighted conclusion arising from different sides even if the review carries different opinions. The social networking site Facebook is the targeted website for this paper. This is because Facebook has many members and vast user-generated data is available. In this paper, an attempt has been made to apply deep learning approach for sentiment analysis on Afaan Oromoo facebook post comments. We also addressed the challenges Afaan Oromoo Sentiment Analysis.

## 2. Related works

### 2.1. Sentiment Analysis in Afaan Oromoo

In Afaan Oromoo the sentiment analysis is new and only a few works were studied. We encountered only two researchers on the Afaan Oromoo language. (Tariku, 2017), conducted aspect based summarization of Afaan Oromoo news text on the news domain. This work is the first attempt at Afaan Oromoo opinion mining. The researchers used manually crafted rules and a lexicon-based approach. The dataset obtained from the ORTO news service. As reported by the researcher, even though the system shows good results, the lack of resources such as lexical database and linguistic resources such as POS made the work challenging. There are also gaps that are needed to be elaborated more. For example, people express their feeling on social media indirectly and their system cannot handle this problem. The other works by (Abate, 2019). The researcher developed an unsupervised approach for Afaan Oromoo on a Facebook domain. Data is obtained from the official facebook page of the Oromoo democratic page and other Activists pages on current political situations. N-gram and POS used as features. As the researcher claims the proposed work shows a promising result. For more illustration the previous studies on AO sentiment analysis is summarized in the table 3 below.

The general work proposed by the two researchers needs the lexical database and it involves the manual collection of lexicons. Moreover, the machine learning method performs better with less human intervention (Vinodhini & Chandrasekaran, 2012). In addition, regarding social media texts where nature the texts are informal, indirect (Tariku, 2017) , slang and idiomatic it is difficult to deal with the previous techniques. Despite these researchers, we proposed a state of the art machine learning and deep learning approaches such as Convolutional

neural network, long-short memory deep learning approaches. According to the literature (Hailong, et al., 2014), the lexicon-based models were not very accurate and a good rule-based model was very hard to elaborate, we implemented state-of-the-art methods for Afaan Oromoo sentiment analysis.

## 2.2. Sentiment analysis of Amharic Language

Unlike Afaan Oromoo Sentiment analysis is not new in Amharic language and many works have been proposed by researchers (Alemu, 2018), (Philemon & Mulugeta, 2014), (Gebremeskel, 2010), (Mengistu, 2013), (Tilahun, 2014) and (Abreham, 2014). The researchers (Gebremeskel, 2010) and (Tilahun, 2014), proposed by using the combination of a rule-based and lexicon-based approach (Gebremeskel, 2010) for movie reviews and news domains, and (Tilahun, 2014) for Hotel, University, and Hospital. The first work, (Gebremeskel, 2010) proposed by using the lexicon and context valence shifter feature selection method. The dataset he used is 303 which is too small. The author (Mengistu, 2013) proposed a supervised machine learning approach NB and decision tree algorithm, again on movie reviews and news but with some modification on the size of the dataset. The researcher used a bag of words and information gain feature selection methods. Another work has been done by the author (Abreham, 2014) with some improvement on the dataset as well as he used three different machine learning algorithm namely NB, MNB, and SVM. N-gram presence, n-gram frequency and n-gram TF-IDF used for the feature extraction method.

(Philemon & Mulugeta, 2014) Also proposed a multi-scale sentiment analysis ranging from -2 to +2. The author used a set of n-gram (unigram, bigram, and hybrid) for the feature selection method and Naïve Bayes classifier for the classification algorithm. As the researcher's claim, the bigram approach performs better, 43.6%, 44.3%, and 39.5% for unigram, bigram, and trigram. But according to the researcher's report, the morphological richness, data cleanness and the absence of large corpora in Amharic make a sentiment analysis of Amharic challenging. Dataset collected from social media, marketing, and news. As (Philemon & Mulugeta, 2014) the machine learning approach requires less effort. (Abreham, 2014) , conducted another solution by using three different machine learning algorithms (NB, DT and ME). The author conducted binary classification which assigns a given document to negative and positive. Bag of words and information gain is used as features and dataset from the news. The researcher (Alemu, 2018) conducted empirical research by using deep learning techniques to improve previous works. (Alemu, 2018), proposed a new solution by applying state of the art study. The dataset obtained from the official facebook page of *Fana Broadcasting Corporation* regarding the socio-political domain. The proposed solution includes emotion icons such as emoji. This is the first work deep learning approach toward Amharic language and as the researcher claims an accuracy of 90.1 %, 82.4 % and 70.1% obtained based the three experiments. According to The researcher, the size of training data and test data has an impact on the performance of the classifier. For example, with 90% training data and 10% test data, accuracy 90.1% obtained, and with 70% training data and 30% test data an accuracy 70.1 obtained. The literature review of Amharic Sentiment Analysis and Opinion mining using different approaches and techniques summarized in the following table 4.

## 2.3. Deep learning for sentiment Analysis

Deep learning technology is one of the most states of the art machine learning approaches, has been recently successfully used in sentiment analysis tasks. (Dong, et al., 2014) Proposed a new model, called the Adaptive Recursive Neural Network (AdaRNN) and aims to classify Twitter texts into three sentiment classes: positive, neutral, and negative. As reported by the author the AdaRNN achieved 66.3% accuracy. (Huang, et al., 2016) Designed Hierarchical Long Short-Term Memory (HLSTM) and gotten 64.1% accuracy on tweet texts. (Tang et al., 2015) presented a new variant of the RNN model, called Gated Recurrent Neural Network (GRNN), and achieved an accuracy of 66.6 % and 45.3% on two different datasets ( Yelp2013–2015data) and (IMDB data) respectively. On the other hand (Qian, Huang, Lei, & Zhu) applied Long Short-Term Memory (LSTM) for binary classification of sentiment and gotten 82.1% accuracy on the movie review data.

Authors (Liu, et al., 2016) designed RNN for text classification with multi-task learning. The following tasks are selected: multi-class classification (somewhat negative, negative, neutral, somewhat positive, positive), binary classification, subjectivity classification which involves subjective or objective (sentence level) and binary classification on document-level. In the article (Liu, et al., 2016) authors presented 3 model architectures of sharing information to model text sequence. The first model architecture utilizes one shared layer for all tasks. The second architecture utilizes different layers for different tasks. The last model assumes the assignment a certain task to a certain level, but also has a shared layer for all the tasks. After experiments were conducted, authors compared obtained results and concluded that on some tasks they achieved better results opposed to the state-of-the-art baselines. Even though the RNN achieved better results there is a disadvantage in RNN. The limitations of RNN is that it is not very good in holding long term dependencies and the problem of vanishing gradient resurface in RNN.

As (Tsunghan et.al., 1996) it is stated that Recurrent Neural Networks (RNN) are capable of dealing with short-term dependencies in a sequence of data. Nevertheless, RNNs have suffered when dealing with long-term

dependencies. These long-term dependencies have a great influence on the meaning and overall polarity of a document. So having methods of capturing long term dependency is very important. Long Short-term memory networks (LSTM) overcomes this long-term dependency problem by introducing a memory into the network.

(Kim, 2014) Designed multichannel CNN and obtained a maximum of 89.6% accuracy with seven different types of data through their CNN model with one convolutional layer. (Moschitti, Severyn and Alessandro, 2015) Employed a pre-trained Word2Vec for their CNN model and achieved 84.79% (phrase-level) and 64.59% (message-level) accuracies on SemEval-2015 data. The CNN model used in (Severyn, et al., 2015) was essentially the same as the model of (Kim, 2014). (Deriu, et al., 2017) Implemented the CNN model which has a combination of two convolutional layers and two pooling layers for four different languages which classify twitter data and obtained a 67.79% F1 score. Another study (Ouyang, et al., 2015), designed the CNN model with convolution pooling layer pairs, and the authors claimed that the model outperformed other previous models.

As we can understand from the above literature, for the sentiment classification, there are two leading types of deep learning techniques: LSTM and CNN. In this work, we proposed a CNN and LSTM model, for effective sentiment classification.

### **2.3.1. LSTM for sentiment analysis**

LSTM is one of the recent successful algorithms in sentiment analysis and other natural language processing tasks. (Wang, et al., 2015) Described that identifying the sentiment of these social media blogs is a challenging task that has attracted increased research interest in recent years and requires state of the art technology to handle these problems. As (Wang, et al., 2015) states that the traditional RNNs are not powerful enough to deal with complex sentiment terminologies, hence an LSTM network is instigated for classifying the sentiment of social media texts. (Liu et al., 2018) Investigated the effectiveness of long short-term memory (LSTM) for sentiment classification of short texts with distributed representation in social media. The researchers addressed that, since social media posts are usually very short, there's a lack of features for effective classification. Thus, word embedding models can be used to learn different word usages in various contexts. To detect the sentiment polarity from short texts and longer dependency, we need to explore deeper semantics of words using deep learning methods.

LSTM (Long Short Term Memory) is the kind of RNN that is used to learn long-range dependencies for text sequences. LSTM contains memory blocks which also known as gates to control the text flow. The memory blocks contain three gates named as input gate; forget gate and output gate to control the flow of information (Miedema, 2018). The author (Miedema, 2018) also described that, the Shortcoming of the Recurrent Neural network and implemented the LSTM for sentiment analysis. Based on many kinds of literature we explored that LSTM is the more advantageous state of the art neural network algorithm for sentiment analysis. So in this work, we focused on LSTM. We proposed the LSTM for Afaan Oromoo based on (Miedema, 2018) but we extended with two hidden layers of LSTM with different memory units. In LSTM or RNN Sentiment analysis will be the two dimensional. Performed in sequence to vector model, that means the input is the sequence of words and the output is a two-dimensional vector indicating the positive or negative class of the sentiment.

### **2.3.2. Convolutional Neural Network for Sentiment Analysis**

CNN is one of the states of the art deep learning classification algorithm. The convolutional filters that automatically learn important features for any task make it more famous. In sentiment analysis also very important, since the convolutional filters can capture the semantic and syntactic of sentiment expressions (Rios & Kavuluru, 2015). In another case, CNN does not need linguistic experts to understand the linguistic structure of the language (Zhang, et al., 2015). As (Kim, 2014), a single convolutional layer, a combination of convolutional filters can achieve comparable performance even without any special hyperparameter adjustment. Because of these CNN is successfully applied to various natural language processing tasks search query (Shen, et al., n.d.), semantic parsing for question answering (Yih et al., 2014), sentence modeling (Nal et al., 2014).

We proposed the CNN model for Afaan Oromoo sentiment analysis based on the architecture developed by (Kim, 2014). Our approach composed of multiple parallel kernel sizes or filters. We focused on a multichannel CNN model with one hidden layer.

## **3. The proposed Afaan oromoo Sentiment Analysis Model**

In this section, we introduce the methodology or the steps we followed in order to conduct Afaan Oromoo Sentiment Analysis. The proposed Afaan Oromoo Sentiment analysis system architecture is depicted in the following system architecture.

### **3.1.1. Data collection**

For this study, the primary data source from *Oromo Democratic Party /ODP* official Facebook page is extracted by using face graph API. The reason for choosing this page is that there is a huge user generated opinions. This page is the government organization page and the government policy related post is released every day on this page. So that the genuine and reliable user-generated data is available on this page. Moreover, this page is a public page and people express their idea about government freely on this page. We focused on sociopolitical related issues, government policy, and other related issues. The total amount of reviews collected is 1452, 726 positive

and 726 negatives. The extracted data is saved in comma delimiter (CSV) format in excel.

### 3.1.2. Data preprocessing

As stated previously, for this thesis we used a supervised machine learning method. Since the supervised method requires the labeled dataset for training purposes the dataset collected was labeled manually by experts. After that, the data is split into training and testing data using scikit-learn `train_test_split`. The training data used to train the classifiers, and the test data used for testing the accuracy of the classifiers.

We split our dataset according to the 80/20 rule (Philemon & Mulugeta, 2014) i.e. eighty percent of the dataset goes to the training set and twenty percent goes to the test set. We used the `train_test_split` method of the sklearn library to perform this task in python. `train_test_split` is faster, simpler so it would be easier to analyze the testing errors.

Another step is preprocessing in order to exclude irrelevant data from the dataset. Preprocessing is very important as it reduces the computational time and increases the classifier performance because noisy data can slow the learning process and decrease the efficiency of the system. Accordingly, our preprocessing includes the following: **Cleaning:** Removal of user names, Removal of links, lower casing, Removal of none Afaan Oromoo texts, unnecessary characters, etc.

**Stopword removal:** some Afaan Oromoo Stopwords are significant for the sentiment classification and need to remain in the text. For instance, the “hin” is used to indicate the negativity of the word: for example, “dhufeera”, “hin dhufne”. In another case, some stop words constitute a phrase: “walii hin gallu”, “isiin waliin jirra” etc. These stop words portray important information. So we filtered removed stop words through a manual process that is not relevant for the classification process.

**Normalization:** homophones like “baay’ee” and “baayee” has the same meaning with different writing. The only difference is that the apostrophe “’” is replaced by “y”.

- Normalization of elongated texts, for example, sin jaallannaaaaaa is normalized to sin jaallanna
- Normalization of numbers into equivalent texts. Example: “sin jaallanna 100% “normalized to” sin jaallanna persentii dhibba tokko”.

**Spelling correction:** we encounter many wrongly spelled texts. So they need to be corrected to the right spelling.

### 3.1.1. Convolutional neural network (CNN)

In this work, we implemented a multi-channel convolutional neural network that performs by using different kernel sizes. As the researcher (Kim, 2014) the multichannel convolutional neural network with multichannel architecture has more effective, especially on small datasets. Despite the researcher implemented on top of word2vec, we used the randomly initialize word embedding i.e. the word embedding learned during training. The researcher also describes experimented with static and dynamic (updated) embedding layers, instead, we focused only on the use of different kernel sizes. A multi-channel convolutional neural network for text classification involves using multiple versions of the standard model with different sized kernels. This allows the document to be processed at different resolutions or different n-grams (groups of words) at a time, whilst the model learns how to best integrate these interpretations. The figure below depicts the proposed architecture of CNN.

In order to build CNN model for sentiment classification, each comment is broken into sentences and, sentences are first tokenized into words, and represented as a matrix where each row corresponds to words. That is each row is a vector that represents a word vectors that index the word into a vocabulary. Let say  $S$  denote the length of the sentence and  $d$  be the dimension of the word vector, therefore we now have the matrix with shape  $S \times d$ . That means the length of the sentence is  $S$  means the count of words in a sentence. Let say the sentence has a total of 9 words, and let say the dimension of the word vector is 5, so we have the matrix of shape  $9 \times 5$ . Now we replaced all words in the sentence replaced by a fixed dimension of 5. Now we have a 5-dimensional word vector. The transformation input is completed and represented as a high dimensional vector, the next step is to apply convolutional filters. We have an Embedding matrix (i.e., input embedding layer) of  $d$  dimension. The filter matches the word vector and varies the region size  $h$ . The region size refers to the number of rows representing words in the sentence matrix that would be filtered at a time. Then convolutional filters slide over full rows of input embedding layer with different kernel sizes and perform element-wise dot product operations.

For example, we have the sentence: ‘*Baayee namatti tola ODP abdi fi kallacha qabsoo oromoo!*’

Let say the first convolution with filter size 2, considering two words, ‘baayee’, ‘namatti’, the filter represented by  $2 \times 5$  since our word vector dimension is 5. The convolution overlays across the vectors of ‘baayee’ and ‘namatti’. Then it performs the element-wise dot product operation for all  $2 \times 5$  matrix elements, adds the result and produces a single value number. For instance,  $0.6 \times 0.1 + 0.2 \times 0.1 + \dots + w_{10} \times 0.1 = 0.82$ . We assumed that the weight is initialized randomly which is performed by the system. Now we got the value for the first sequence for the first convolution. Again the convolution moves down one word and overlays across the word vectors of the next words and performs the same operations to get the next value.

So the output of the filter has the form  $s-h+1 \times 1$ , in this case,  $9-1+1 \times 1 = 9$  for the first convolution.  $9-2+1 \times 1 = 8$  for the second convolution and  $9-3+1 \times 1 = 7$  for the third convolution with 3 filters which is illustrated in the architecture above figure 12.

The same operation is performed for each convolution for example for filter size 3, it considers three words at a time and performs the above procedures. Finally, the result from different convolutional channels is concatenated into a single dimension. Before going to the fully connected layer, maximum pooling operation is performed to pick the maximum features and finally fed to the fully connected layer for classification. In additions the detail parameters used along with the CNN is described as the following:

To obtain the feature map  $c$  we add bias and apply activation function. The feature can be mathematically represented as (Kim, 2014):

$$c_i = f(w * s_{i:i+n-1} + b)$$

Filter weights are initialized randomly in the beginning and then tuned through the training process. Where  $w$  is a vector of weights, "\*" refers to the dot product,  $s_{i:i+n-1}$  is a sliding window as illustrated in the above example,  $b \in \mathbb{R}$  is a bias vector, and  $f$  is a non-linear activation function. At each convolutional channel, we apply nonlinear activation function which is called ReLU (Moschitti, et al., 2015) and (Kim, 2014), (Jumayl, et al., 2019). Rectifier linear unit or ReLU is the most widely used nonlinear activation function on CNN. The task of RELU is to avoid negative value.ie.it maps the negative value to zero and returns if the result is positive. ReLU allows producing a non-linear decision boundary. So it can be written as:

$$f(x) = \max(0, x)$$

*It returns x if the value is positive and returns zero if the result is negative.*

The filter is employed to each sequence of words in the sentence that corresponds to the filter size  $\{s_{1:n}, s_{2:n+1}, \dots, s_{s-n+1:s}$  to generate a feature map (Kim, 2014):

$$c(w) = [c_1, c_2, \dots, c_{s-n+1}]$$

**Pooling layer:** In this work max-pooling operation is used as it is extensively used by many researchers the most widely used pooling mechanism. In one thing it allows reducing the size of the feature map as it combines the vectors resulting from the different convolutional windows into a single 1-dimensional vector and at the same time preserving the most relevant feature. Pooling greatly affects the performance of CNN. The pooling operation is used to ideally this vector will capture the most relevant features of the sentence.

$$\hat{c} = \max\{c(w)\}$$

Such operation provides a single feature  $\hat{c}$  for the feature map produced by the particular kernel  $w$ . the other technique is flattening. Flattening mechanism is added to convert the pooled result in to one dimension or single dimension before going to fully connected of the output layer.

**Fully connected layer:** After max-pooling is performed, the concatenated feature vector is fed into a fully connected layer. At this layer, the classification result output is produced. Since our work is a binary classification task, we used sigmoid (Jumayl, et al., 2019) as the activation function and binary cross-entropy as our loss function. Because the Softmax function is used in multiclass classification, whereas sigmoid function is used in binary classification.

**Dropout:** Dropout is a method where randomly selected neurons are dropped during training. They are "dropped-out" randomly (Kim, 2014). This technique is used for preventing the network from overfitting. We used the dropout at every convolutional channel to avoid bias. At the fully connected layer also we used dropout, with parameter 0.1, which means 10% of unnecessary neurons are dropped.

**Training the network:** Training is usually performed using a stochastic gradient descent by randomly selecting some samples from the dataset. Dropout ensures regularization and applied before a fully connected layer. The dropout method assumes that only on the training stage some portion of neurons is removed (dropout rate is set to 0.) that prevents co-adaptation of neurons and leads to learning more robust features and makes model generalize new data well (Srivastava et al., 2014).

Training of the CNN assumes the fine-tuning of the network parameters. This tuning process called backpropagation error. Backpropagation will be applied to compute the gradient of the error function with respect to the filter weights. Adam algorithm (Kingma & Ba, 2014) that is a stochastic gradient descent algorithm is used for optimizing parameters of CNN (updating weights).

### 3.1.2. Long Short Term memory

The main intuition of the LSTM network is that it has the mechanism of long-term memory and accordingly is proficient in handling long-term dependencies.

LSTM has a special structure called cell blocks. These cells are composed of an input gate, the forget gate and the output gate. The figure 2 below emphasis the visualization of the LSTM component.

#### 3.1.2.1. Forget Gate

$$f_t = \sigma(W_f \cdot [X_t, h_t - 1] + b_f), \dots \dots \dots (1)$$

The forget gate is used to forget the unnecessary information. It has a sigmoid layer that takes the previous output at  $h_{t-1}$  and the current input at a time  $x_t$  and outputs the value between 0 and 1.

The main objective of this task is to determine the extent to which a value or information is thrown away or remain in the cell. This can be done by the value form the current input at time  $t$  and the value from the previous hidden

state at time  $t - 1$  are combined in to a single tensor. Then passes through the neural network sigmoid function for transformation. The value from the sigmoid function is squishing between zero and one (0 and 1) because of the sigmoid. After multiplying the number with the internal state, the information to be forgotten or kept in to the cell is determined by, the value which is closer to zero is forgotten and the value which is closer to one is kept in the cell.

*if  $f_t=0$ , the previous internal state is totally forgotten  
 else if  $f_t=1$  will be passed unaltered.*

### 3.1.2.2. Input Gate

$$i_t = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i) \dots \dots \dots (2)$$

The task of the input gate is to decide the extent of new input or value that will be flown into the cell. I other words it determines which of the new input will be updated or ignored. This can be done by receiving the new input and the previously hidden state output passed to another sigmoid layer. Again the output value from the sigmoid is between zero and one due to sigmoid. So the output of the input gate then multiplied with the output of the candidate layer as the following:

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, X_t] + b_c) \dots \dots (3)$$

The candidate vector  $\tilde{C}_t$  is created by neural network hyperbolic tangent (*Tanh*) and is added to the internal state. Now old cell state  $C_{t-1}$  is updated into new cell state  $C_t$  via the following rule:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \dots \dots \dots (4)$$

As we can see from the formula, to obtain the new cell  $C_t$  the old state is multiplied by  $f_t$ , forgetting the value we decided to forget earlier. Then we add the product of  $i_t * \tilde{C}_t$ . This is the new candidate values, mounted by how much we decided to update each state value.

### 3.1.2.3. Output Gate

$$O_t = \sigma(W_o \cdot [h_{t-1}, X_t] + b_o) \dots \dots \dots (5)$$

$$h_t = O_t * \tanh(C_t) \dots \dots \dots (6)$$

The output gate computes what part of the cell is used to compute the output activation function of LSTM and which parts of the cells going to output. This can be done by the cell state is pass through the *tanh* function this squishes the value between -1 and 1. And then multiply it by the output of sigmoid function or gate. By this method, we get the output we need. When relating to our work, it decides whether the polarity is positive or negative.

Our model composed of two stacked LSTM layers or with two LSTM layers with 256 memory units each. This makes the model a deeper more accurate prediction. The same to CNN, we used the embedding layer for LSTM. As mentioned earlier Word embeddings facilitate learned word representations. As reported by many researchers word embedding is has many advantages in extracting complex language features which have been an issue in previous researches (Joshi et al., 2016). The same step as the CNN, after preprocessing and padding is performed and represented in matrix form, this matrix was finally fed as input to the LSTM layer. Then the output of the first LSTM is input to the next LSTM layer. The first LSTM layer provides a sequence output rather than a single value output to the next LSTM layer. That means it provides one output per input time step rather than one output time step for all input time steps. This adds levels of abstraction of input observations over time and representing the problem at different time scales. This approach possibly lets the hidden state at each level to operate at different timescale.

So the additional hidden layers assumed to recombine the learned representation from prior layers and build new representations at high levels of abstraction. The sigmoid function is employed the same with CNN since our work is a binary classification. Dropout regularization is also used as described in the above section to avoid co-adoption and to avoid overfitting. The detail of the network parameter and configuration described in chapter four.

The overview of the proposed architecture of LSTM is depicted in figure 3 below, each comment with different lengths need to have the same length, so the shorter comments are padded with to have an equal size with the longer sentences. The maximum length of the reviews in our dataset is 1344. So, by adding zero to any reviews less than 1344, we make the reviews have equal length. So, there are 1344 time steps in the model for each word and accordingly, each word of the review is being fed to the model at each time step. This further passed to the word embeddings, the word embedding in a case is the dense representation of words, where words with a similar meaning are close to the vector space. By this method, the model learns the relevant feature representation by itself.

The input to the model is a text of fixed length words, where each word is encoded to integers. So we have 1344 time steps, at each time step one word is fed to the model. The word is further entered into the embedding layer with one neuron, in this layer the words are transformed into a real-valued vector of length 256. In this way, 256 features are created. Next, an LSTM layer with 256 neurons is added to the network, each of the features is multiplied by weight for each LSTM cell, where each LSTM cell contains four gates discussed in the above section. Next to the 256 features, the output of the previous time step is also used as an input for the LSTM cells. The LSTM enhances recurrent connections to each other and predicts the series of words in the records. The final layer

is the output layer with two neurons. Here the weighted sum of the 256 outputs of the LSTM layer is taken and a sigmoid activation is added and performs the dot product between features and the weight matrix  $w$  used to predict the value between 0 and 1 for the two class.

#### 4. Results and Discussion

The experiment is done to measure the overall performance of the developed deep learning sentiment analysis model. CNN and LSTM implemented using Keras deep learning library in python. We used evaluation metrics (precision, recall, accuracy and f1 score) to evaluate the performance of the classifiers.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \dots (1)$$

*TP* is the number of true positives: the reviews/comments that are actually positive and estimated as positive.

*TN* is the number of true negatives: the reviews/comments that are actually negative and estimated as negative,

*FP* is the number of false positives: the reviews/comments that are actually negative but estimated as positive,

*FN* is the number of false negatives: the reviews/comments that are actually positive but estimated as negative.

A Precision can be estimated using the following formula (Jumayl, et al., 2019):

$$precision = \frac{TP}{TP+FP} \dots (2)$$

Precision shows how many positive reviews received from the classifier are correct. The greater precision the fewer number of false hits. However, precision does not show whether all the correct answers are returned by the classifier. In order to take into account the latter recall will be used (Jumayl, et al., 2019):

$$recall = \frac{TP}{TP+FN} \dots (3)$$

Recall shows the ability of the classifier to “guess” as many correct answers, (reviews with correct labels) as possible out of the expected.

The more precision and recall the better. However, simultaneous achievement of high precision and recall is almost impossible in real life that is why the balance between two metrics has to be found. *F1* score is a harmonic mean of precision and recall (Jumayl, et al., 2019):

$$f_1 = \frac{2*precision*recall}{precision+recall} \dots (4)$$

In addition to precision, accuracy, and f1score, the neural network is measured by the average loss and accuracy. The loss is calculated on training and validation and its interoperation is how well the model is doing for these two sets. It is a summation of the errors made for each example in training or validation sets. The lowest loss is the best model.

To perform classification, we used Tokenizer from Keras preprocessing python library. The Tokenizer performs the Vectorization of a text corpus into a list of integers. So each integer maps to a value in a dictionary that translates the entire corpus, with the keys in the dictionary being the vocabulary terms themselves. We choose Tokenizer because of many reasons, that we can add the parameter num\_words, which is accountable for setting the size of the vocabulary i.e. the most common num\_words will be then kept. Moreover, we have comments in which each text sequence has a different length of words. To tackle this, Keras has a pad\_sequence() option which simply pads the sequence of words with zeroes.

The results of CNN and LSTM described in the section below:

##### 4.1. Convolutional Neural Network

We got the architecture of the network configuration through try and error and fine-tuning process. The model we proposed using CNN performed well despite our dataset is small. We applied a maximum dropout (0.1). This is a help to remove unnecessary biases from the network. The optimal parameter network architecture is obtained through fine-tuning. After many searches of efforts, we found that the following network configurations perform good results. The following Table 3 shows the CNN network configuration and the two figures fig 4 and 5 show how accuracy increases and the loss decreases with parameters defined respectively.

The confusion matrix of the system is illustrated in the table 4 below.

As we can understand from the table, from the 145 positive reviews, the system correctly classifies 127 and 19 reviews are misclassified. And, from 146 negative reviews, 13 are misclassified and 132 are correctly classified. The precision and recall of the system illustrated in the table 5 below:

Accordingly, the proposed system by CNN achieved an accuracy of 89% and f1 score of 87%.

One of the strengths of the model is its capability to handle the contextually of the words. As mentioned earlier, social media texts lack contextually and it is difficult to deal with it by using the traditional methods. Our proposed model overcomes this problem using deep learning model. This approach learns and extracts features by using different kernels at the same time.

##### 4.2. Long-short Term Memory

Our LSTM model achieved an accuracy of 87.6% and f1 score 87.7% based on the architecture given in the table



6 below. We investigated the following network architecture for LSTM.

The two figures above, fig 6 and 7 show how accuracy increases and average loss decreases with the defined architecture.

The confusion matrix of the LSTM classifier is given in the table 7 and the classification report, i.e. precision and recall is depicted in Table 8 below after training and testing with the network configuration discussed above.

The two tables show the precision, recall, and the confusion matrix or the number of true positive, true negative and false positive and false negative of the LSTM classifier.

In general, CNN can able to handle the longer dependency of the words through different convolutional filters. When the context of the word is used to determine the polarity of the text rather than the probability of the occurrence of the word both CNN and LSTM are the best approaches. In addition, the two deep learning (CNN and LSTM) requires no special feature selection methods, since they discover and learn the relevant features from the text.

The LSTM by its nature has the capability to hold relevant information to the task at hand. This makes it better for text classification and sentiment analysis tasks. But relatively slower computational time than CNN.

## 5. Conclusions and recommendation

The rapid development of social media networks like Facebook, twitter, etc. provides a variety of benefits, in facilitating the way people share their opinion and increase the speed of public comments. Due to this, companies and governments receive high volumes of electronic comments every day. Identifying the polarity of the comments may be valuable input for decision making. Though, a large number of reviews make it difficult for a company or any institutions to react to the opinions rapidly and take appropriate decisions. Therefore, sentiment Analysis has become a major area of research interest in the field of Natural Language Processing and Text Mining to overcome these problems. The sentiment analysis task is under research since the early 2000s. Nevertheless, it is a new area and at an initial state in Afaan Oromoo.

The main drawback of the lexicon-based approach is the inability to detect sentiment words with domain and context specific polarity orientations. In addition, the performance of lexicon-based methods in terms of time complexity and accuracy heavily depends on the number of words in the dictionary, that is, performance decreases significantly with the exponential growth of the dictionary size. Hence, according to the literature review, it was found that the majority of sentiment analysis approaches rely on supervised machine-learning methods. Therefore, it was Long Short Term Memory and Convolutional neural network approaches as far as these methods are state of the art among researchers and they provide meaningful results.

We studied three methods, first Multinomial Naïve Bayes that use Term frequency and inverse document frequency representation and n-gram features for training the classifier. Secondly, Long Short Term Memory deep learning method that uses word embeddings and two different hidden layers to further make precise the polarity of the reviews/comments. Thirdly Convolutional Neural Network deep learning technology that uses word embeddings and applies different convolutional filters and extracts sentiment of the text is studied. Therefore, we aimed to perform experiments and investigate the performance of three different algorithms detecting positive and negative comments. Furthermore, the algorithm which gives the best results is defined.

The experimental results show that our proposed CNN performed an accuracy of 89%. Whereas, the LSTM achieved an accuracy of 87.6%.

In general, in this study shows that LSTM performs slightly less than CNN and MNB. The MNB outperforms both CNN and LSTM, and it is simple and demands fewer resources than both CNN and LSTM. CNN is relatively faster than LSTM and is capable of handling longer text and context of words as LSTM. But, both requires solid computational resources and large amount of training sample.

The system can deal with lengthy comments, as the lengthy comments were a challenge to classify as it was common to find a contradiction in the sentiment expressed and longer expression depends on the meaning of its predecessors. The two deep learning approaches are good at handling indirect comments, but the MNB machine learning approach still has challenge with indirect comments.

The general limitation of the study is that, Social media is an informal means of communication that includes considerable use of slang, malformed words, and colloquial expressions. People use idiomatic expression to express feelings in some cases. So, our system got challenges with idiomatic expressions in some cases.

Based on our work we provided several feature directions:

- We focused on texts, Emoticons and emoji expressions that carry laugh, sad, angry, and happy, love, etc. need to be included and labeled whether emoticon, emoji expression refers to a positive or negative meaning.
- The neural networks LSTM and CNN requires huge data to perform good results. Hence, it is necessary to have a well prepared standard corpus.
- The LSTM and CNN may have a good performance with pre-trained word embeddings (trained on a sufficiently large corpus). Therefore, preparing and trying with pre-trained word embeddings.

- This study considers the review level or document-level sentiment analysis. Others like sentence level can be considered.
- we considered only the binary sentiment classification, in the future multi-scale or multi-class also needs to be considered
- We focused on facebook domain, so different social media networks like twitter, YouTube, etc. also need to be targeted in the future.

## References

- Abate, J., 2019. Unsupervised Opinion Mining Approach for Afaan Oromoo Sentiments, Haramaya University, 2019..
- Abreham, G., 2014. *Opinion Mining from Amharic Entertainment Text*. Addis Abeba: Addis Abeba Universty.
- Akinyi, A. J., 2014. *A Comparative evaluation of sentiment analysis Techniques on Facebook data using three machine Learning algorithms: Naïve Bayes, Maximum Entropy and Support Vector machines*, NAIROBI.: Unpublished.
- Alemu, A., 2018. DEEP LEARNING APPROACH FOR AMHARIC SENTIMENT ANALYSIS UNIVERSITY OF GONDAR.
- Bakliwal, A. et al., 2013. *Sentiment Analysis of Political Tweets*. s.l., s.n.
- Chiavetta, F., Bosco, G. L. & Pilato, G., 2016. *A Lexicon-based Approach for Sentiment Classification of Amazon Books Reviews in Italian Language*. s.l., s.n., pp. 159-170.
- Deriu, J. et al., 2017. *Leveraging large amounts of weakly supervised data for multi-language sentiment classification..* Perth, Australia, s.n.
- Dong, L. et al., 2014. *Adaptive recursive neuralnetwork for target-dependent twitter sentiment classification..* Maryland, s.n.
- Gebremeskel, S., 2010. *Sentiment Mining Model for Opinionated Amharic Texts*. Addis Ababa(Addis Ababa ): Addis Ababa University.
- George and Joseph, 2014. Text Classification by Augmenting Bag of Words (BOW) Representation with Co-occurrence Feature. *IOSR Journal of Computer Engineering (IOSR-JCE)*, e-ISSN: 2278-0661, p- ISSN, Volume 16(Issue 1), pp. PP 34-38.
- Hailong, Wenyan & Bo, 2014. Machine learning and lexicon based methods for sentiment classification: A survey.. *In Web Information System and Application Conference (WISA), 2014 11th*, Volume IEEE., pp. 262-265.
- Huang, M., Cao, Y. & Dong, C., 2016. *Modeling rich contexts for sentiment classification with lstm..* s.l., s.n.
- Joshi et al., 2016. *How do cultural differences impact the quality of sarcasm annotation*. s.l., s.n.
- Jumayl, Robert & Lu, 2019. *Multi-ChannelConvolutionalNeuralNetworkfor Twitter Emotion and Sentiment Recognition*. s.l., Proceedings of NAACL-HLT.
- Kim, Y., 2014. *Convolutional neural networks for sentence classification*. s.l., arXiv preprint arXiv:1408.5882..
- Kingma, D. & Ba, J., 2014. *Adam: A method for stochastic optimization*, s.l.: arXiv preprint arXiv:1412.6980..
- Liu et al., 2018. *LSTM Approach to Short Text Sentiment Classification with word embeddings*. Beijing, s.n., pp. 214-223.
- Liu, B., 2012. *Sentiment Analysis and Opinion Mining*. s.l.:Morgan Claypool Publishers.
- Liu, B., 2012. *Sentiment analysis and opinion mining. Synthesis lectures on human language technologies*. s.l.:Morgan& Claypool Publishers.
- Liu, P., Qiu, X. & Huang, 2016. *Recurrent neural network for text classification with multi-task learning*, s.l.: arXiv preprint arXiv:1605.05101..
- Mengistu, B., 2013. *Sentiment Analysis for Amharic opinionated text*, Addis Ababa: Ababa university.
- Miedema, F., 2018. *Sentiment Analysis with Long Short-Term Memory networks*, s.l.: s.n.
- Moschitti, Severyn and Alessandro, 2015. *Twitter Sentiment Analysis with Deep Convolutional Neural Networks*. s.l., s.n.
- Moschitti, Severyn, A. & Alessandro, 2015. *Twitter Sentiment Analysis with Deep Convolutional Neural Networks*. s.l., s.n.
- Nal et al., 2014. *A convolutional neural network for modelling sentences*. s.l., s.n.
- Ouyang, X., Zhou, P., Li, C. & Liu, L., 2015. *Sentiment Analysis Using Convolutional Neural Network*. s.l., s.n.
- Pang, B., Lee, L. & Vaithyanathan, S., 2002. *Thumbs up? Sentiment Classification using Machine Learning Techniques*. s.l., s.n., pp. 79-86.
- Philemon, W. & Mulugeta, W., 2014. A Machine Learning Approach to Multi-Scale Sentiment Analysis of Amharic Online Posts. *HiLCoE Journal of Computer Science and Technology*, No. 2, Volume Vol. 2, p. 87.
- Popescu and Etzioni, 2005. *Extracting product features and opinions from reviews*. s.l., s.n.
- Qian, Q., Huang, M., Lei, J. & Zhu, X., n.d. *Linguistically regularized lstms for sentiment classification*. s.l., s.n.
- Rios, A. & Kavuluru, 2015. *Convolutional neural networks for biomedical text classification: Application in indexing biomedical articles*. Atlanta, Georgia, s.n.

Severyn, Moschitti & Unitn, 2015. *Training deep convolutional neural network for twitter sentiment classification*. Denver, Colorado, s.n.

Shen, Y. et al., n.d. *Learning Semantic Representations Using Convolutional Neural Networks for Web Search*. 2014, s.n.

Srivastava et al., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, pp. 1929-1958.

Tang et al., 2015. *Document modeling with gated recurrent neural network for sentiment classification*.. Lisbon, Portugal,, s.n.

Tariku, W., 2017. *SENTIMENT MINING AND ASPECT BASED SUMMARIZATION OF OPINIONATED AFAAN OROMOO NEWS TEXT*, Debre Berham: DEBRE BERHAN UNIVERSITY.

Tilahun, T., 2014. Linguistic Localization of Opinion Mining from Amharic Blogs. *International Journal of Information Technology & Computer Sciences Perspectives* © Pezzottaite Journals., p. 895.

Tsungnan et.al., 1996. Learning LongTerm Dependencies in NARX Recurrent Neural Networks. *Transactions on neural networks*.

Vinodhini, G. & Chandrasekaran, R., 2012. Sentiment analysis and Opinion Mining:A survey.. *International Journal of Advanced Resarch in Computer Science and Software Engineerin*, Volume vol. 2, no. 6, pp. pp 1-11.

Wang, X. et al., 2015. Predicting Polarities of Tweets by Composing Word Embeddings with Long Short-Term Memory. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, p. 1343–1353.

Yih et al., 2014. *Semantic Parsing for Single-Relation Question Answering*. Maryland, s.n.

Zhang, X., Zhao, J. & LeCun, Y., 2015. *Character-level convolutional networks for text classification*. Quebec, s.n.

Author	objective	features	model	classes	Selected model	Domain Afaan oromoo Data source	No Dataset
(Tariku, 2017), Msc Thesis, DBU	Assign feature sentiment & summarization	Lexicons +rule based	rule based	+ve&-ve	General lexicon with rule based	ORTO news service	400 reviews
(Abate, 2019) Journal Article	Feature extraction & polarity classification	Lexicon, POS, N-gram	Unsupervised	+ve,-ve &neutral	bigram	(OPDO) official Facebook page, political bloggers page	600 reviews

**Table 1 summary of SA and opinion mining of AO**

Author	Objective	Features	Model	Classes	Selected Model	Domain, Amharic (data source)	No dataset
(Alemu, 2018) Journal Article	Assign docs sentiment	learning word embedding	CNN	+ve, very +ve, extremely +ve, neutral, -ve, very -ve and extremely -ve		Fana BC Facebook page	1600 reviews
(Philemon & Mulugeta, 2014) Journal Article	Assign docs sentiment	Unigram, Bigram & Hybrid valence shifter	NB	-2, -1, 0, +1, +2  (multi-scale)	Naïve Bayes + Bigram	Social media, Product marketing & News	608
(Abreham, 2014) Thesis (AAU)	Assign docs sentiment	BoW & IG	NB, DT & ME	+ve & -ve	Naïve Bayes with Information Gain	EBC, diretube.com & habesha.com	616
(Tilahun, 2014) Journal Article	Assign feature sentiment	Feature + Opinion + Context valence shifter & n-grams-TF-IDF	Rule & Lexicon Based	+ve, -ve & neutral	Feature with adjacent left & right adjective	Hotel, University & Hospital	484
Abebe, (2013), MSc Thesis (UoG)	Assign docs sentiment	BoW, IG	NB and DT	+ve & -ve	Naïve Bayes + Information Gain	Movie review+news	456
(Mengistu, 2013) MSc Thesis (AAU)	Assign docs sentiment	n-grams presence, n-grams frequency & n-grams-TF-IDF	NB, MNB & SVM	+ve, -ve & Neutral	Support Vector Machine with unigram	ERTA.com + FanaBC.com + diretube.com	576
(Gebremeskel, 2010)	Assign docs sentiment	lexicon + Context valence shifter	Rule & Lexicon Based	+ve, -ve & Neutral	General lexicon with valence shifter	Movie review + news doc	303

**Table 2 Summary of Amharic SA and OM**

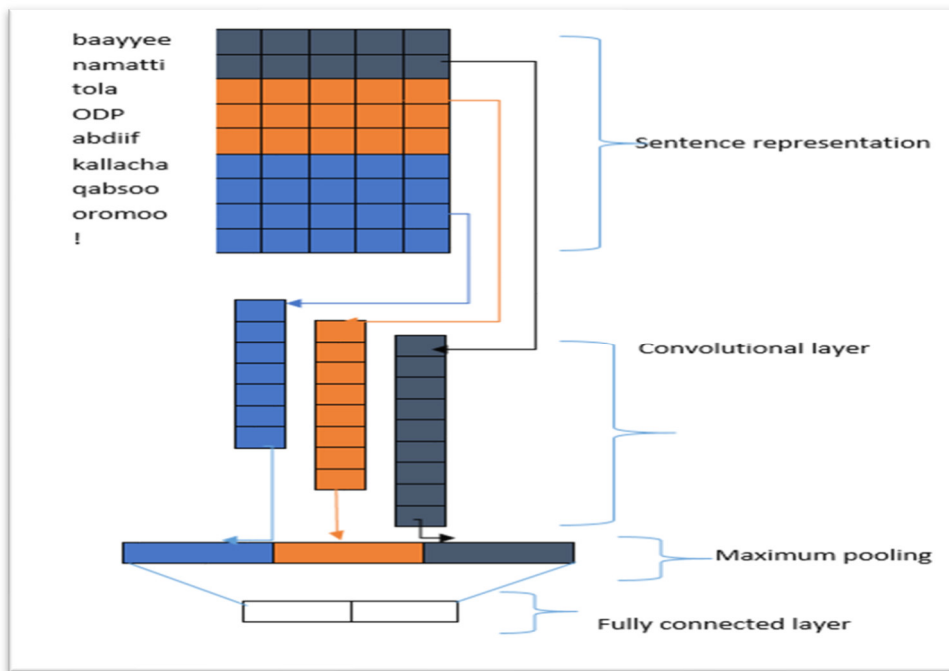


Figure 1 CNN Architecture for AO SA

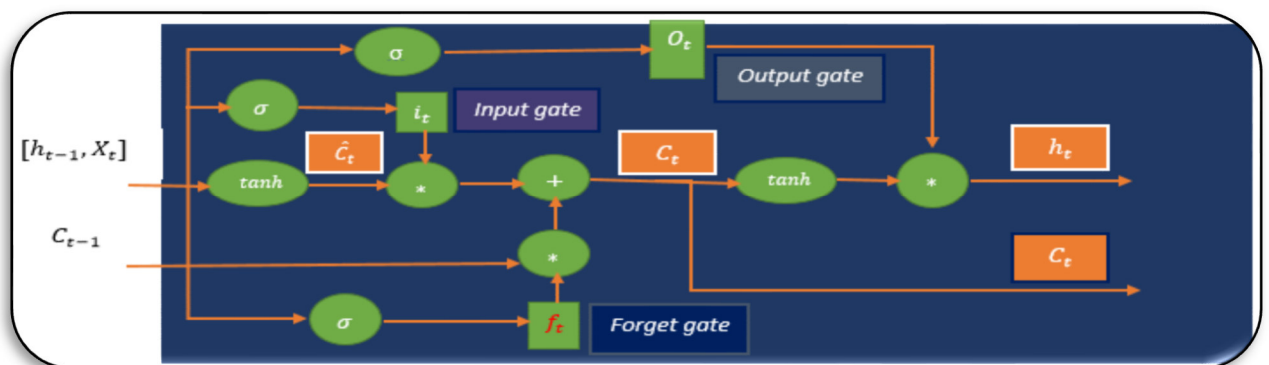


Figure 1 LSTM Network

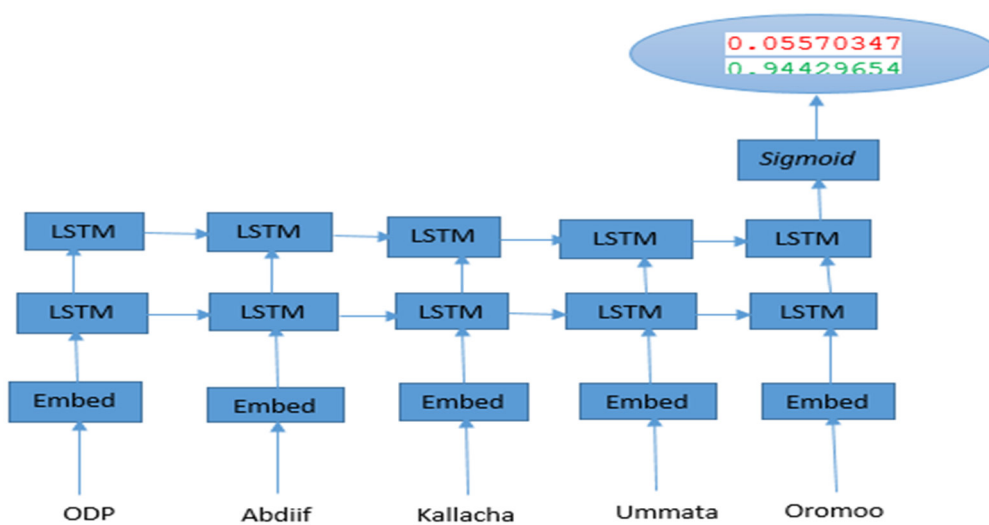
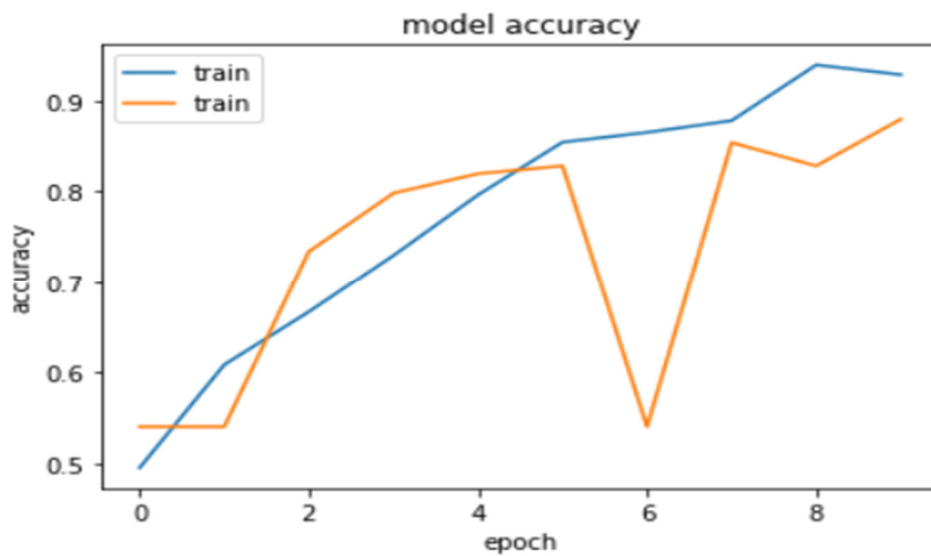


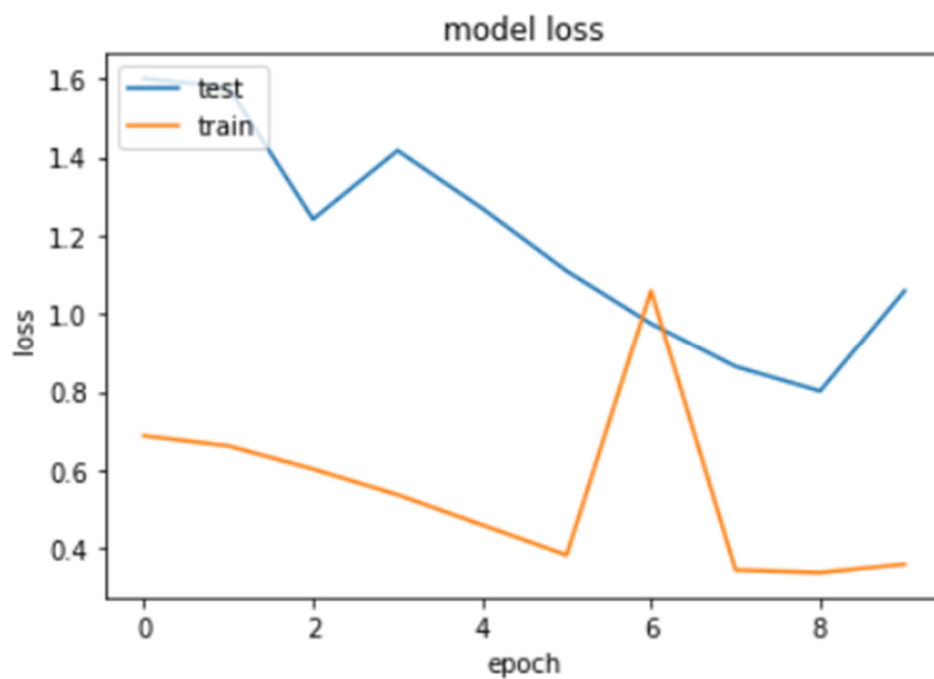
Figure 2 proposed LSTM architecture for AO SA

Hyperparameter training	parameters
Embedding dimension	10
Convolutional filers(kernel size)	1,2, 3,4,5
dropout	0.1, at fully connected layer
Pooling	Max-pooling
Number of filters	64
epochs	10
Learning rate	Default(0.001), beta_1=0.9, beta_2=0.999
Batch size	32

**Table 2 CNN configuration for AO SA**



**Figure 3 model accuracy of CNN**



**Figure 4 model loss of CNN**

True Positive	127
False Positive	13
True Negative	132
False Negative	19

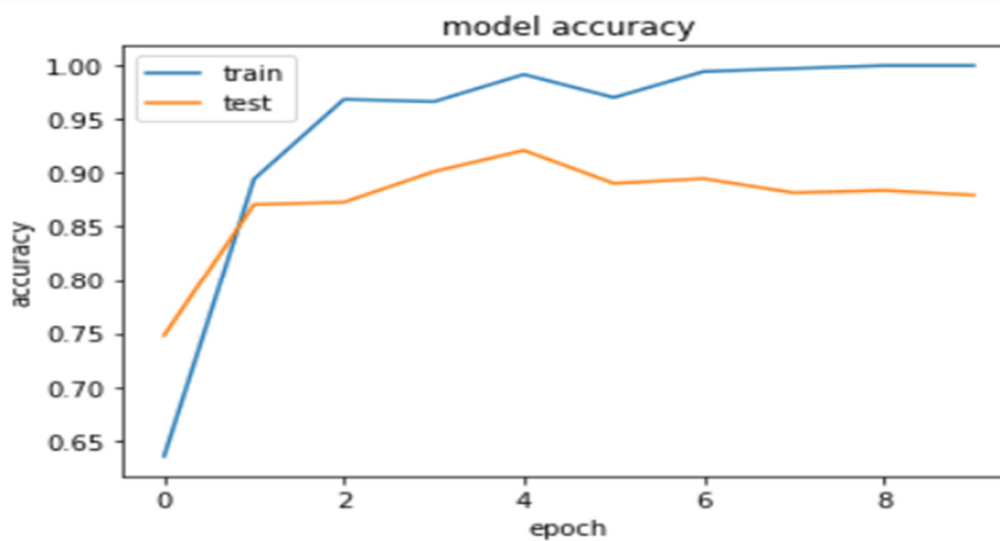
**Table 3 Confusion Matrix of the proposed CNN**

class	Positive	Negative
precision	0.90	0.86
recall	0.87	0.91

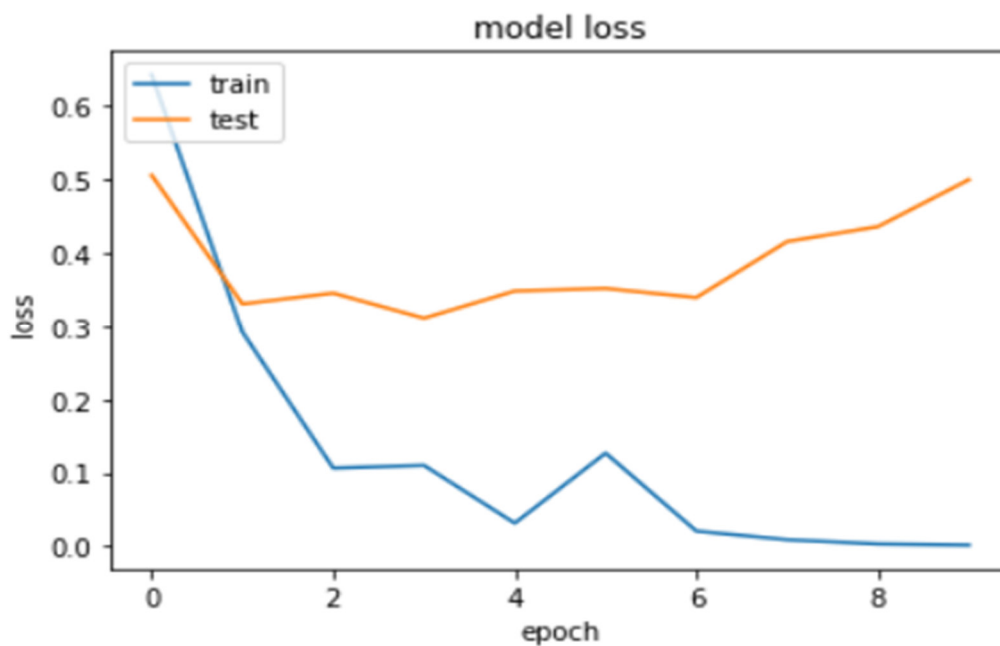
**Table 4 classification report of the proposed LSTM**

Hyper parameter training	parameters
Embedding dimension	256
dropout	0.3 , recurrent dropout =0.2
Memory unit	250 for both LSTM layers
epochs	10
Learning rate	Default(0.001), beta_1=0.9, beta_2=0.999
Batch size	20

**Table 5 LSTM network configuration for AO SA**



**Figure 5 model accuracy with epochs for LSTM**



**Figure 6 model loss of LSTM in each epoch**

True Positive	127
False Positive	18
True Negative	128
False Negative	18

**Table 6 the confusion matrix of LSTM**

Class	Positive	Negative
precision	0.876	0.88
recall	0.876	0.88

**Table 7 precision and recall performance metrics of the LSTM**