Ensemble Based Emotion Detection Model for Multi-Social Platforms

*Bala A[1], Mrs. Abisoye O. A.[2], Oluwaseun, A. O.[3], & Solomon A. A[4]
[1]Computer science Department, Federal University of Technology, PMB 65 Minna Niger State, Nigeria
[2]Computer Science Department, Federal University of Technology, PMB 65 Minna Niger State, Nigeria
[3]Computer Science Department, Federal University of Technology, PMB 65 Minna Niger State, Nigeria
[4]Computer Science Department, Federal University of Technology, PMB 65 Minna Niger State, Nigeria
[1]abdullahibala1@gmail.com, [2]o.abisoye@futminna.edu.ng, [3]o.ojerinde@futminna.edu.ng & [4]solo.adepoju@futminna.edu.ng

ABSTRACT
In recent years, there is an exponential growth in public generated data such as image, video and text, this is due to the rapid emanation of diverse social media users. This available textual data is frequently adopted and significantly important for extracting information such as user's sentiments, and emotions. Considering the complexity and large amount of textual data, the adoption of various machine learning (statistical models), and deep learning model (neural network) for the analysis of emotion has not yet attained optimum accuracy. Recently, Transformer based Architecture (BERT) are achieving state of art accuracy. Hence, this study adopts an ensemble based model using BERT-Large, LSTM and SVM for detecting user's emotion. The experimental evaluation carried out resulted in an optimum accuracy of 93%.

Keywords: *emotions, ensemble, social media, BERT-large, SVM, LSTM.*

1 INTRODUCTION

In recent year, due to the emanation of various social media platform such as WhatsApp, Facebook, Instagram, and twitters, has resulted to large amount of public available text corpus or dataset (Chiorrini et al., 2021). Considering the fact that social media user tends to feel more comfortable expressing their feelings or opinions during chatting session, hence social media data is tag to be more realistic or rich in emotional context (Chiorrini et al., 2021). Its identify that open contribution or opinion of customer feedback on a particular product can facilitate the quick identification of issues or improvement that is essential for better customer services. Recently, researcher has focus on various techniques that is capably of analyzing user sentiment and emotion in written form automatically. This approach is known has sentiment analysis or emotion analysis (Denemark, 2011).

Furthermore, the analysis of sentiment is described as the process of automating, identifying and deducing user opinion from a written or documented text, this also involve classification of user opinion into either positive, negative of neutral sentiment (Zhang et al., 2018). While emotion analysis is a type of sentiment analysis but with a wider range of classification class, it involves identifying user emotion been express in text format. Based on research it is reveal that the task of emotion analysis is more challenging than that of sentiment analysis, this is due to greater number of possible emotions (classification class) (Chiorrini et al., 2021).

Emotions plays a crucial role in every forms of human communication, which often influence an individual perception of an experience, topic or event. User (customer) opinions and feedback on various products, online or offline can be collated, via various mediums such as comments, reviews, message forums, and polls. The gathered data can take the form of either audio, video or text (Minaee et al., 2019) (Bartneck et al., 2017). However, the analysis of human emotion from textual data is a field of sturdy uses natural language processing (NLP), computational linguistic and text analysis to extract subjective information from source materials. The detection of emotion using natural language processing is a sub domain of sentiment analysis which primarily focuses on extracting emotional information of the user (writer) from text data. Emotion detection aims to pinpoint specific emotions such as happiness, sadness, love, and fear (Albu & Spînu, 2022).

Ensemble methods involve the combination of two or more machine learning (ML) or hybridization of ML and deep learning (DL) techniques to achieve better performance and optimal results (Mosavi et al., 2019) (Mosavi et al., 2019). These methods take advantage of the strengths of multiple techniques to improve overall performance. The Ensemble methods may include a prediction unit and an optimization unit for more accurate output. They are flexible and they provide higher capabilities in comparison to single methods, and ensemble method becoming increasingly popular due to their potential (Ardabili et al., 2019). In other words, ensemble base learning involves combining multiple machine learning models which perform intelligent task collectively. Common instance of ensemble models includes

1 Netflix: this system adopts hundreds of machine learning model for predicting possible preferences of movies for user.
2 IBM Watson: this ensemble based model is the first prediction model to won the quiz game of jeopardy.
3 Google: in recent model google adopt ensemble based techniques for developing a neural machine translation system. (Serban et al., 2018).

1.1 RESEARCH PROBLEM

Emotion analysis is tag as one of the most challenging task in terms of opinion mining, due to the broader possible emotion class in comparison to sentiment analysis. Various heuristic and statistic based approach has been proposed for emotion mining from text corpus, but the performance is lower than that of deep learning approach in term of accuracy (Chiorrini et al., 2021). Recently, in the domain of natural language processing various state of art architecture has been design and adopted by researchers for performing emotion classification, which resulted in great improvement in term of accuracy. The BERT, GPT and other model build on transformer architecture are the current state-of-art model. (Albu & Spînu, 2022) adopted the BERT-base and SVM model in an hybridize ensemble techniques to analyze user emotions, its concluded by the researchers that optimum accuracy has not yet been attained because they are better versions of BERT model with larger contextual weights. Hence, this research introduce a multi-social ensemble model based on BERT-large, SVM and LSTM.

1.2 RESEARCH GOAL

This sturdy aims to developed a multi-social emotion detection system using ensemble techniques. The propose techniques model hybridize the Bidirectional encoder from transformer (BERT-large), Support Vector Machine (SVM) and Long Short Term Memory (LSTM) model to achieve utmost emotion classification accuracy. In achieving this goal this sturdy will gathered emotion dataset, train individual model and hybridize the prediction result for ensemble decision making. Finally, the developed ensemble model will be evaluated using standard performance metrics.

2 RELATED WORKS

Considering the research work of (Majeed et al., 2020) which focuses on identifying emotions in Roman Urdu text. Despite Roman Urdu is frequently utilize for information exchange on social media platforms, there is limited research on emotion detection in this language. The primary challenge is the lack of benchmark corpora for detecting user emotion from text, which is crucial for various NLP tasks. The analysis of user emotion from a text corpus has various significant benefit such as improving product quality, dialog systems, investment trends, and mental health. The research developed a comprehensive corpus of 18k sentences from various domains and annotated with 6 different categories to focus on the emotional polarity of Roman Urdu sentences. Based on the experimental result its identify that the SVM prediction model had the best F-1 score measure among the applied algorithms.

(Chiorrini et al., 2021) the researchers investigates the effective use of BERT models thus, the bidirectional encoders for the analysis of user sentiment and emotion detection using data from user Tweet. Two different classifiers were adopted for the two tasks, and the performance of the models was evaluated using real-world tweet datasets. Based on the experiments carried out its revealed that the adopted models were able to achieve 0.92, 0.90 percent accuracy in term of detecting and analyzing user emotion and sentiment.

Its identify by (Albu & Spînu, 2022) that the automatic detection of user emotions for their tweet generated data has numerous real life application. This research transforms an imbalanced data into a balance data by including a neutral class to a benchmark dataset that contain 4 categorical classes of user emotion thus, joy, sadness, fear, and anger. The study adopts SVM and BERT model for the detection and identification of user emotion using the balanced dataset. A novel ensemble model was proposed by merging the BERT and SVM models. The experiments carried out shows that the proposed model achieved an optimum intelligent of 0.91 in recognizing emotions in tweets.

Sentiment analysis approach is examined by (Sinan & Kayaalp, 2021) on Yelp restaurant reviews, IMDB movie reviews, and data collected from Twitter. Word-2-Vec, the BERT model, Bag-of-word along with the TF-IDF feature extraction was the word embedding techniques used by the researchers. The sentiment analysis model was developed using techniques this includes; the LSTM, Naïve Bayesian, CNN and SVM model. Model performance was evaluated using metrics such as, F-1 (F), Precision, ROC curve, and Accuracy. In respect to the result it can be concluded that the word embedding generated by the BERT architecture perform best, and the authors recommend the use of BERT approach to address similar issues.

(Tang *et al*., 2018) introduce Bayesian machine learning model for detecting emotions from social media news articles. This model takes into account the hidden connection or relationship that map two sentences together in a document and can detect emotions at both the document and sentence level. The developed model outperforms the existing method on the level of sentence dis-ambiguity and document dis-ambiguity and sentence-level emotion detection, as shown in experiments on public corpora.

Considering the work of (Kim et al., 2019) introducing a new framework for detecting human emotions in a smart advance cities using the application of Internet of a thing to facilitate the smart environment. The developed approach is named VEmoBar, capable of creating some barrier that can virtually senses human emotions through wireless signals and reflections. A problem is defined to determine the optimal placement of VEmoBars to maximize cumulative accuracy. A novel approach and system initialization are proposed to solve this problem, and the results are evaluated through simulations in various scenarios. The article also discusses future issues and challenges for implementing this framework in smart cities.

In this research thesis the author introduces a prediction model that is capable of predicting human emotion from real time image capturing. The model utilizes the deep CNN and its evaluated against 8 different datasets. However, the researchers thus, (Jaiswal, 2019) minimize the parameters of the network by 909 compared to Vanilla CNN and 509 in respect to current state-of-the-art research. The model achieved 74% accuracy, which is an improvement over previous models with reduced computational complexity. This model can be useful in fields such as elderly care, child therapy, and babysitting where robots need to understand human emotions to provide more customized assistance.

(Vitiugin & Barnab, 2022) This paper describes the system submitted by the WSSC Team to the EmoEvalEs@IberLEF 2021 emotion detection competition. The proposed model combines transformer embedding with topic information and offense features to classify emotions in social media text. The results showed that the model outperforms the benchmark models that are state of art. A weighted average 0.66% F1 score was achieved with 0.67 accuracy.

(Jaiswal et al., 2018) This researcher introduce a new recommendation system which incorporates user's emotions with interests to provide personalized product suggestions. Unlike existing approaches, which rely on past feedback, similarity of other users' buying patterns, or a combination of both, the proposed system does not require a large amount of data. Instead, it captures a user's eye-gaze and facial expressions as they browse a website using an inexpensive webcam. Eye-gaze detection is done by extracting the pupil center of both eyes, and calculating a reference point using a joint probability. Facial expression analysis is done by analyzing landmark points on the face to determine the user's emotion. Both methods work in near real-time, allowing the system to provide intelligent recommendations on-the-fly without the need for user feedback or buying patterns.

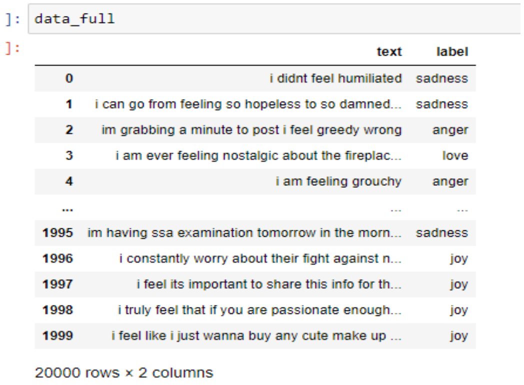
3 RESEARCH METHODOLOGY

This section introduces the methodology adopted for developing, prediction and detecting social media user emotions from textual data. The series of scientific step taking which range from data gathering, data exploration, cleaning, transformation, preprocessing, model training, and model evaluation will be diligently discus in this section. Various conceptual diagram illustrating the internal working technology of the proposed system is critically explained in this section. Diagram such as system architecture, flow chart diagram and use case diagram.

3.1 DATA COLLECTION

This sturdy utilized the available emotion data from a popular data science platform repository (Hugging Face). It’s a company that provides an NLP platform for featuring pre-trained models, fine-tuning libraries, and development tools. The platform is well-known for its transformer-based models like BERT and GPT-2, which have demonstrated excellent performance in multiple NLP tasks. It allows developers to easily integrate NLP into applications and projects through APIs and libraries. Additionally, Hugging Face offers a model hub where users can access pre-trained models, customize them using their own data, and share them with others (Yu et al., 2020) (Wolf et al., 2020).

Furthermore, the adopted dataset was downloaded from the Hugging Face repository using the *datasets* python API module, and all the training, testing, and validation data samples were downloaded into the Google Colab environment in a CSV format. The downloaded data was archive in the Google Drive for future reference and easy importation into the Google Colab for further usage using the python *panda's* library. It was essential to merge the entire dataset into a single data for proper usage as depicted in the Figure 1 below. The dataset includes twenty thousand (20,000) emotional tweet data points, with two (2) columns of features which include the tweet text and the corresponding emotion.



```
]: data_full
]:
```

	text	label
0	i didnt feel humiliated	sadness
1	i can go from feeling so hopeless to so damned...	sadness
2	im grabbing a minute to post i feel greedy wrong	anger
3	i am ever feeling nostalgic about the fireplac...	love
4	i am feeling grouchy	anger
...	...	...
1995	im having ssa examination tomorrow in the morn...	sadness
1996	i constantly worry about their fight against n...	joy
1997	i feel its important to share this info for th...	joy
1998	i truly feel that if you are passionate enough...	joy
1999	i feel like i just wanna buy any cute make up ...	joy

20000 rows × 2 columns

Figure 1: The Emotion Tweets Dataset

The figure 1 shows the sample data point from the downloaded tweet emotions dataset. It's clearly visualize that the *text* column contains various user tweets information while the *label* column contains various type of emotions such has sadness, anger, love, fear and joy. Moreover, the figure shows the total number of columns thus, 2 and the total number available row or data entries thus, 20,000 rows.

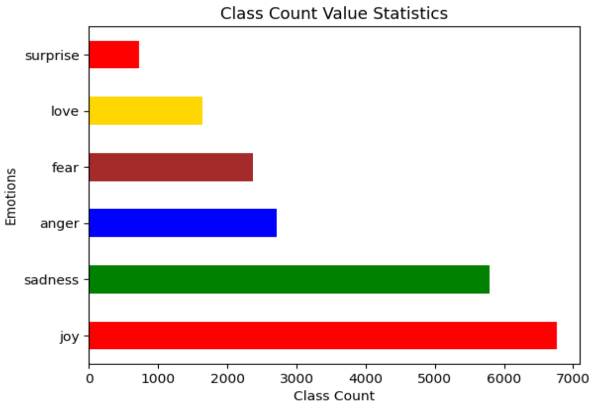


Figure 2. Tweets Emotion Classes Count

3.2 Data Exploration

This subsection include various exploration carry out on the dataset. Every data science research requires comprehensive understanding about the dataset for extraction of meaningful features from the dataset. Based on analysis its identify that the label column contains six (6) different classes, this includes; sadness, anger, love, surprise, fear, and joy. This is visually illustrated using the bar chart below.

Base on the visual plots it's denoted that the dataset is divided in bars base on the *label* (emotion type) columns, and it's also identify that the present classes in the emotion tweet dataset is imbalance (having different number of sample in each emotion label class). Its identify that most of the data-sample belong to the *joy* label class with over 6,000 data sample, which is followed by the *sadness* label data-sample with over 5,000 sample, up to *surprise* label sample with minimum number of data sample of 1,000.

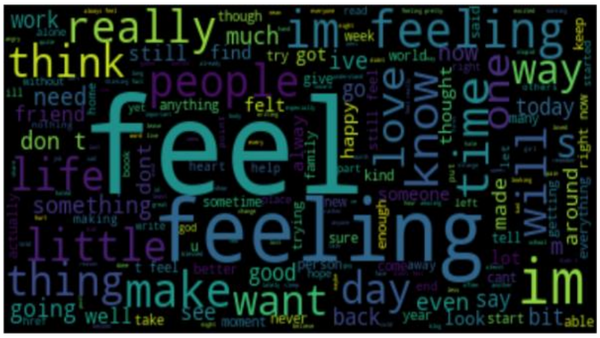
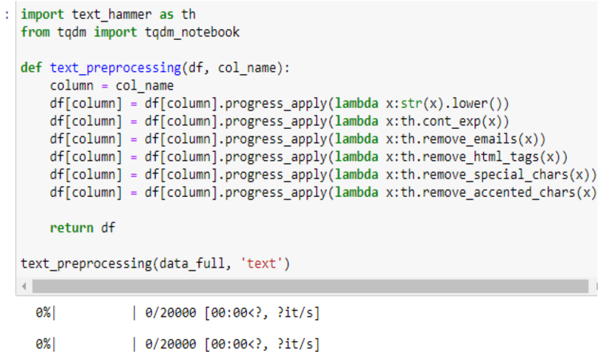


Figure 3. Word Cloud Visual showing Word Frequency

In respect to the word cloud figure above, the frequency of word distribution or frequency in the emotion tweet dataset is explore using the application programming interface for word cloud (Heimerl et al., 2014). that words such as "feel", "feeling", "really", "think", and "make", are the most occurring words in the tweet emotion corpus. The word cloud representation uses text size to indicate how often a word appears in a corpus. Hence, the larger the text size in the figure the more frequently is word appears in the text corpus.

3.3 Data Cleaning & Preprocessing

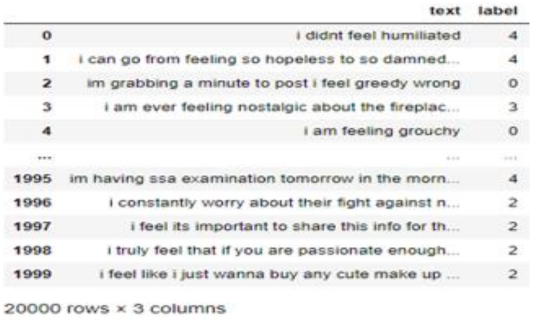
This section is essentially important in the field on data science, machine learning and deep learning. Its ensure that noisy or unwanted content are remove from the raw dataset. Tweet dataset are one of the most challenging data to clean because of the high usage of informal language, emoji's, abbreviation and URLs



```
: import text_hammer as th
  from tqdm import tqdm_notebook

  def text_preprocessing(df, col_name):
      column = col_name
      df[column] = df[column].progress_apply(lambda x:str(x).lower())
      df[column] = df[column].progress_apply(lambda x:th.cont_exp(x))
      df[column] = df[column].progress_apply(lambda x:th.remove_emails(x))
      df[column] = df[column].progress_apply(lambda x:th.remove_html_tags(x))
      df[column] = df[column].progress_apply(lambda x:th.remove_special_chars(x))
      df[column] = df[column].progress_apply(lambda x:th.remove_accented_chars(x)

      return df

  text_preprocessing(data_full, 'text')

    0%|          | 0/20000 [00:00<?, ?it/s]

    0%|          | 0/20000 [00:00<?, ?it/s]
```

Fig 4. Data cleaning using text hammer

(Derczynski et al., 2013). Hence, this sturdy is adopting the text *hammer* python module to address this issues. The figure 4 and 5 show the cleaning process and the resulted clean text header sample.



	text	label
0	i didnt feel humiliated	4
1	i can go from feeling so hopeless to so damned...	4
2	im grabbing a minute to post i feel greedy wrong	0
3	i am ever feeling nostalgic about the fireplac...	3
4	i am feeling grouchy	0
...	...	...
1995	im having ssa examination tomorrow in the morn...	4
1996	i constantly worry about their fight against n...	2
1997	i feel its important to share this info for th...	2
1998	i truly feel that if you are passionate enough...	2
1999	i feel like i just wanna buy any cute make up ...	2

20000 rows × 3 columns

Figure 5. Clean data and Transform Label

Figure 4 illustrates the utilization of the "text hammer" library to conduct multiple data cleaning process. A customized helper function is defined and all major preprocessing steps are defined within this function. The cleaning actions that are considered includes; removal of any mail address, HTML-related text, lowercasing all text, removing of symbols, emoji's and special characters.

Furthermore, the label column is transform into number format thus, anger *is assign 0, 1 is assign to fear, 3 is assign to love, love is assign to 3, four to sadness, and surprise is tag as 5.* The figure 5 show the cleaning text and the transformed label.

3.4 DATA SPLITTING

The final data preprocessing stage include data splitting, thus splitting of dataset into training set and testing set. In this study 70 percent (14,000) of the dataset is considered for model training and 30 percent (6,000) of the remaining data for testing purpose. The figure 6 above shows the code snippet for achieving the data splitting.

3.5 ENSEMBLE BERT-LARGE, SVM AND LSTM

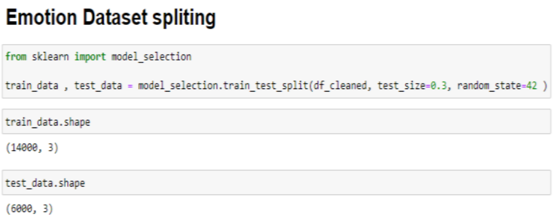
In this section the conceptual diagram of the proposed ensemble model is illustrated, which combines the bidirectional encoder from transformer (BERT-large), Support Vector Machine (SVM), and Long Short-Term Memory (LSTM). The illustration of the proposed model is depicted diagrammatically in detail using Figure 7.

Figure 7 illustrates the conceptual diagrammatic representation of the proposed ensemble model. The model uses sub models for predictions and this includes the BERT-large, SVM, and LSTM model. As depicted in the figure, the dataset is loaded from the local repository into the google colab and training samples are feed into the three models (BERT, LSTM, and SVM) for training. Each machine learning model is trained individually, and the resulting trained models are saved locally on the system directory for application and ensemble usage. The saved models are then loaded, and the prediction outcomes of each model are utilized in an ensemble technique known as voting. The final prediction outcome can be viewed by the user through the user interface.

The flowchart diagram illustrates the systematic process of data flow, which start with data import and ends with model evaluation. The data flow starts by importing the emotion tweet data into the colab environment, then the data is preprocessed, cleaned, and transformed into a standard format that can be understood and process by the ensemble model. It is shown in the data flow that the dataset is split into 70% for training and 30% for testing. The model training includes all three assembled models: The Support Vector Machine, Long Short-Term Memory, and BERT-large model. Finally, the split testing data is fed into the ensemble model for model classification evaluation using standard metrics (accuracy, precision and recall).

3.6 Material and Tool

This sturdy adopt various software tools for the development of the proposed model. Then the entire coding is carried out using the google collaboratory notebook environment and jupyter notebook, because it provides access to GPU and TPU processing power (Alves & Machado Vieira, 2019). Other software tools used Microsoft Visio for drawing the various conceptual diagram. Python is choosing as the programming language of choice due to is richness in data science module.



Emotion Dataset spliting

```
from sklearn import model_selection

train_data , test_data = model_selection.train_test_split(df_cleaned, test_size=0.3, random_state=42 )
```

```
train_data.shape
```

```
(14000, 3)
```

```
test_data.shape
```

```
(6000, 3)
```

Figure 6. Emotion Tweet Data Splitting (70%, 30%)

4 RESULTS AND DISCUSSION

This section explains how the actual implementation and training of the ensemble model is carried out, the various performance evaluation metrics considered includes the Accuracy, precision, recall and F1-score. Finally, the model will be evaluated with the existing sturdy for performance comparison. Each model (SVM, LSTM and BERT) is trained separately and later

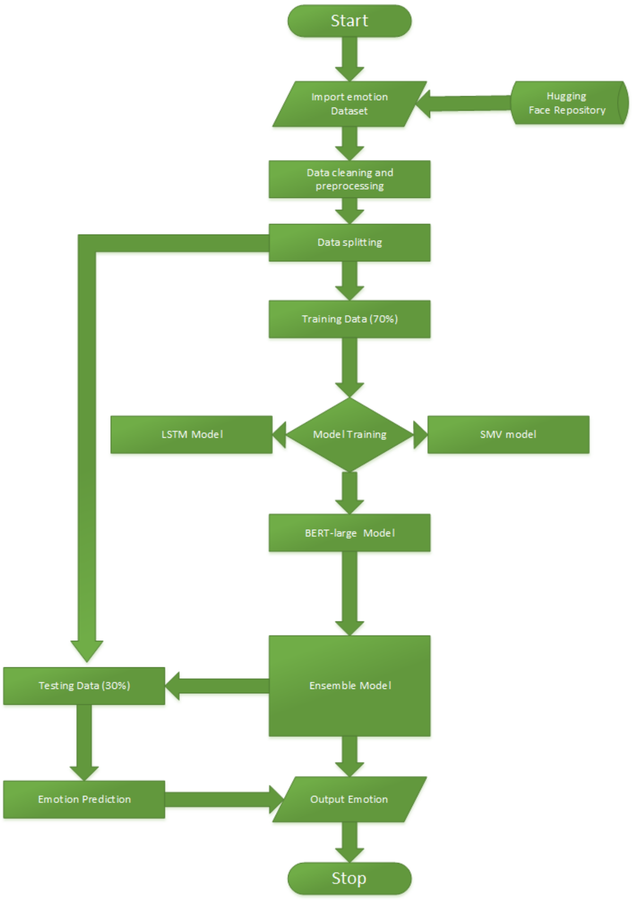
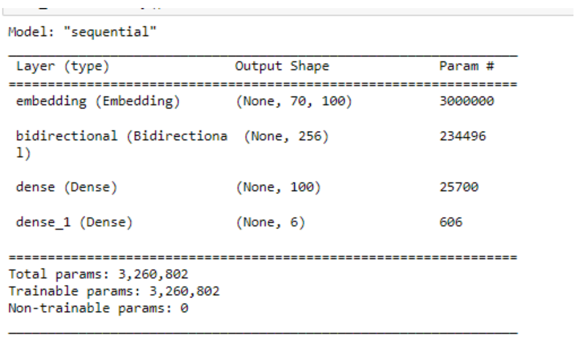


Figure 8. The data flow diagram

ensemble for emotion prediction.



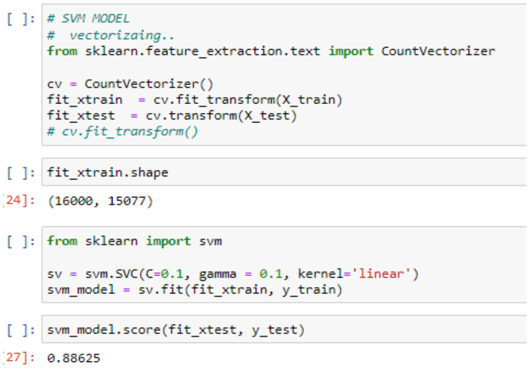
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 70, 100)           3000000

 bidirectional (Bidirectiona  (None, 256)              234496
 l)

 dense (Dense)               (None, 100)               25700

 dense_1 (Dense)             (None, 6)                 606

=================================================================
Total params: 3,260,802
Trainable params: 3,260,802
Non-trainable params: 0
_________________________________________________________________
```

Figure 11 LSTM model summary

4.1 SUPPORT VECTOR MACHINE MODEL

It's essential to carry out some preprocessing on the textual data before it can be utilize by the SVM machine learning model. Hence, this sturdy adopt count vectorization to transform the textual data into numerical dictionary corpus. Based on parameter tuning (cross fold validation) its identify that the best parameter combination is set to c=0.1, gamma = 0.1 and kernel = 'linear'. The preprocessing and training is programmatically achieved using the predefined methods in the sk-learn module. The figure 9 show the experimental implementation.

Based on the figure 9. It's clearly shown that the text column of the emotion dataset is transform into numerical representation using the *fit_transform* method. In this format the system is able to learn a dictionary of words (word map to a unique number), and this is use to train the Support vector machine model. however, an accuracy of 88.6 % is achieved. The classification report is visualizing below by considering the various performance metrics.

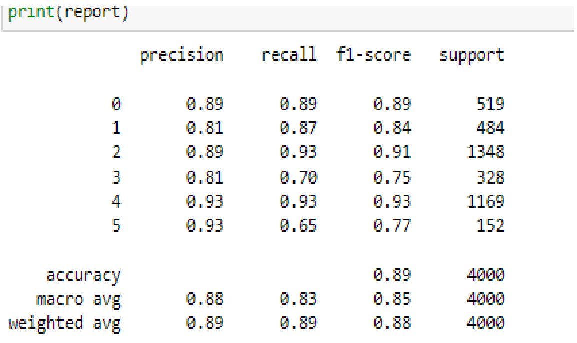


```
[ ]: # SVM MODEL
     #  vectorizaing..
     from sklearn.feature_extraction.text import CountVectorizer

     cv = CountVectorizer()
     fit_xtrain  = cv.fit_transform(X_train)
     fit_xtest  = cv.transform(X_test)
     # cv.fit_transform()

[ ]: fit_xtrain.shape

24]: (16000, 15077)

[ ]: from sklearn import svm

     sv = svm.SVC(C=0.1, gamma = 0.1, kernel='linear')
     svm_model = sv.fit(fit_xtrain, y_train)

[ ]: svm_model.score(fit_xtest, y_test)

27]: 0.88625
```

Figure 9. SVM preprocessing and training

4.2 LONG SHORT TERM MEMORY MODEL

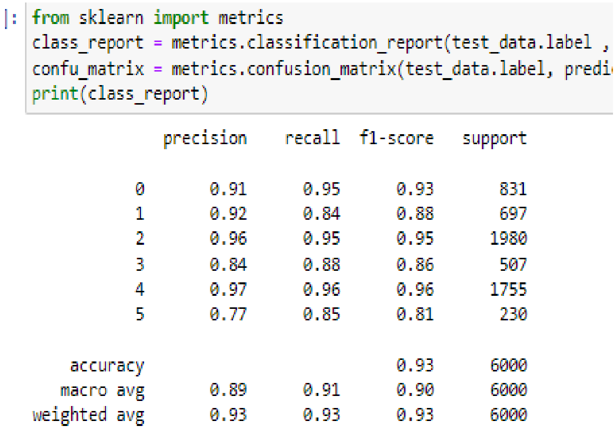
This section requires a separate preprocessing techniques for proper deep learning utilization of text data. The tensor flow API is use in perform text tokenization, sequencing and padding before they are passed into the deep learning architecture.



```
print(report)

              precision    recall  f1-score   support

           0       0.89      0.89      0.89       519
           1       0.81      0.87      0.84       484
           2       0.89      0.93      0.91      1348
           3       0.81      0.70      0.75       328
           4       0.93      0.93      0.93      1169
           5       0.93      0.65      0.77       152

    accuracy                           0.89      4000
   macro avg       0.88      0.83      0.85      4000
weighted avg       0.89      0.89      0.88      4000
```

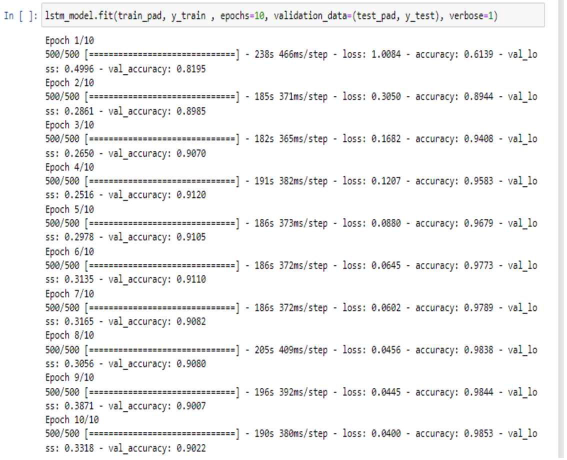
Figure 10 SVM classification Report

The figure shows the number of layer present in the model summary and the actual parameter shape of each layer. Based on the figure and embedding layer comes first where the input parameter and the vector size of each word is define. This is immediately followed by the LSTM layer then a fully connected dense layer and finally a last layer with 6 neurons (output layer)



```
]: from sklearn import metrics
   class_report = metrics.classification_report(test_data.label ,
   confu_matrix = metrics.confusion_matrix(test_data.label, predi
   print(class_report)

              precision    recall  f1-score   support

           0       0.91      0.95      0.93       831
           1       0.92      0.84      0.88       697
           2       0.96      0.95      0.95      1980
           3       0.84      0.88      0.86       507
           4       0.97      0.96      0.96      1755
           5       0.77      0.85      0.81       230

    accuracy                           0.93      6000
   macro avg       0.89      0.91      0.90      6000
weighted avg       0.93      0.93      0.93      6000
```

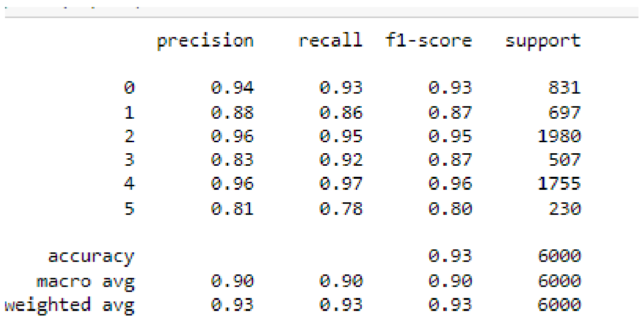
Figure 15 BERT classification report



```
In [ ]: lstm_model.fit(train_pad, y_train , epochs=10, validation_data=(test_pad, y_test), verbose=1)

        Epoch 1/10
        500/500 [==============================] - 238s 466ms/step - loss: 1.0084 - accuracy: 0.6139 - val_lo
        ss: 0.4996 - val_accuracy: 0.8195
        Epoch 2/10
        500/500 [==============================] - 185s 371ms/step - loss: 0.3050 - accuracy: 0.8944 - val_lo
        ss: 0.2861 - val_accuracy: 0.8985
        Epoch 3/10
        500/500 [==============================] - 182s 365ms/step - loss: 0.1682 - accuracy: 0.9408 - val_lo
        ss: 0.2650 - val_accuracy: 0.9070
        Epoch 4/10
        500/500 [==============================] - 191s 382ms/step - loss: 0.1207 - accuracy: 0.9583 - val_lo
        ss: 0.2516 - val_accuracy: 0.9120
        Epoch 5/10
        500/500 [==============================] - 186s 373ms/step - loss: 0.0880 - accuracy: 0.9679 - val_lo
        ss: 0.2978 - val_accuracy: 0.9105
        Epoch 6/10
        500/500 [==============================] - 186s 372ms/step - loss: 0.0645 - accuracy: 0.9773 - val_lo
        ss: 0.3135 - val_accuracy: 0.9110
        Epoch 7/10
        500/500 [==============================] - 186s 372ms/step - loss: 0.0602 - accuracy: 0.9789 - val_lo
        ss: 0.3165 - val_accuracy: 0.9082
        Epoch 8/10
        500/500 [==============================] - 205s 409ms/step - loss: 0.0456 - accuracy: 0.9838 - val_lo
        ss: 0.3056 - val_accuracy: 0.9080
        Epoch 9/10
        500/500 [==============================] - 196s 392ms/step - loss: 0.0445 - accuracy: 0.9844 - val_lo
        ss: 0.3871 - val_accuracy: 0.9007
        Epoch 10/10
        500/500 [==============================] - 190s 380ms/step - loss: 0.0400 - accuracy: 0.9853 - val_lo
        ss: 0.3318 - val_accuracy: 0.9022
```

Figure 11 LSTM training.

The LSTM model is trained for 10 epochs and for every iteration the accuracy, loss and validation accuracy is printed on the console. However, accuracy of 90% is achieved after the 10 iteration of training. The classification report is visualized below, putting into considering the various performance metrics.



```
              precision    recall  f1-score   support

           0       0.94      0.93      0.93       831
           1       0.88      0.86      0.87       697
           2       0.96      0.95      0.95      1980
           3       0.83      0.92      0.87       507
           4       0.96      0.97      0.96      1755
           5       0.81      0.78      0.80       230

    accuracy                           0.93      6000
   macro avg       0.90      0.90      0.90      6000
weighted avg       0.93      0.93      0.93      6000
```

Figure 16 Ensemble classification report

Fig 13 LSM classification report

4.3 BIDIRECTIONAL ENCODER FROM TRANSFORMER (BERT)

The BERT model weight is download from the hugging face repository and fine turned by adding the emotion classification heads using tensforflow layers. The architectural structure is shown in the figure 14 below.

Based on the figure its identify that the model defines two input layer, thus the input id's and the attention mask input with 70 maximum word length. The next layer is the BERT layer follow by a one dimensional max pooling layer, then we have the dense layer, one dropout layer, another dense layer and finally an output layer. After 10 epoch of training the model was able to achieve an accuracy of 93%.

4.4 ENSEMBLE MODEL (SVM, LSTM, BERT)

The three trained model are loaded from the local repository and stack together by comparing their prediction about and selecting the majority vote approach. Hence, these experiment priorities the voting of BERT prediction over other model. Based on the experiment its identify that the ensemble model accuracy combined generate same accuracy as the BERT model.

Generally, the result gotten from the model (SVM, LSTM, BERT) training are summarize in the table below along with the existing study

TABLE 1: MODEL EVALUATION SUMMARY

S/N	Model	Accuracy
1	SVM	83%
2	LSTM	90%
3	BERT-Large	93%
4	Ensemble model (SVM, LSTM, BERT-L)	93%

Base on the result summary in table 1. We can see that both BERT-large and ensemble techniques perform best in term of prediction accuracy of 93%.

TABLE 2: MODEL EVALUATION SUMMARY

S/N	(Albu & Spînu, 2022)		Proposed Model	
1	SVM	84%	SVM	83%
2	-	-	LSTM	90%
3	BERTweet	89%	BERT-Large	93%
4	Ensemble	91%	Ensemble model (SVM, LSTM, BERT-L)	93%

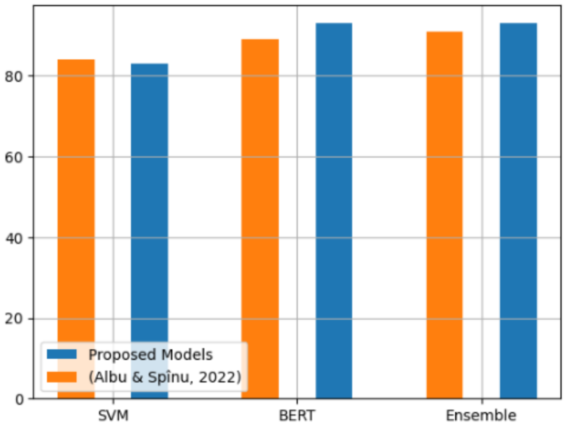


Figure. 17. Performance Comparison (Existing and Proposed model)

Table 2. evaluate the performance comparison between the proposed method and the existing model. Based on the table its identify that the proposed model is more efficient in term of accuracy than the existing model. The visual representation of table II is show in figure 16 below.

5 CONCLUSION

In consideration of the experimental result carried out, thus the outcome in term of accuracy and performance efficiency. The SVM model achieve an accuracy of 83%, while LSTM achieve 90% accuracy, BERT-Large achieve 93% accuracy and the utilization of the three model in an assemble approach also achieve an accuracy of 93%. Its identify that the BERT-large perform best due to his high richness in contextual information. However, the adoption of ensemble approach in respect to BERT-Large alone

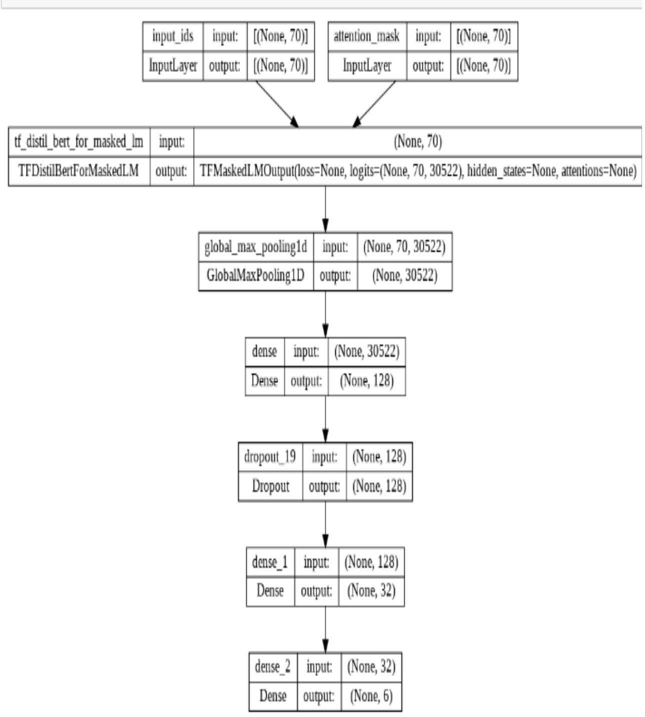

input_ids | input: | [(None, 70)]
InputLayer | output: | [(None, 70)]
attention_mask | input: | [(None, 70)]
InputLayer | output: | [(None, 70)]
tf_distil_bert_for_masked_lm | input: | (None, 70)
TFDistilBertForMaskedLM | output: | TFMaskedLMOutput(loss=None, logits=(None, 70, 30522), hidden_states=None, attentions=None)
global_max_pooling1d | input: | (None, 70, 30522)
GlobalMaxPooling1D | output: | (None, 30522)
dense | input: | (None, 30522)
Dense | output: | (None, 128)
dropout_19 | input: | (None, 128)
Dropout | output: | (None, 128)
dense_1 | input: | (None, 128)
Dense | output: | (None, 32)
dense_2 | input: | (None, 32)
Dense | output: | (None, 6)

Figure 14 BERT model with Emotion Classification heads

does not significantly improve the accuracy. In conclusion the proposed model was able to achieved an optimum accuracy of 93%, and its recommended to used BERT-large has standalone model instead of the developed ensemble techniques for performance reasons (in term of prediction speed during real time usage)

ACKNOWLEDGEMENTS

A short acknowledgement section can be written between the conclusion and the references. Authors may wish to acknowledge the sponsors of the research and others in brief. Acknowledging the contributions of other colleagues who are not included in the authorship of this paper is also added in this section. If no acknowledgement is necessary, this section should not appear in the paper.

REFERENCE

Albu, I. A., & Spînu, S. (2022). Emotion Detection From Tweets Using a Bert and Svm Ensemble Model. *UPB Scientific Bulletin, Series C: Electrical Engineering and Computer Science*, *84*(1), 63–74.

Alves, F. R. V., & Machado Vieira, R. P. (2019). The Newton Fractal's Leonardo Sequence Study with the Google Colab. *International Electronic Journal of Mathematics Education*, *15*(2). https://doi.org/10.29333/iejme/6440

Ardabili, S., Mosavi, A., & Varkonyi-koczy, A. R. (2019). *Advances in Machine Learning Modeling Reviewing Hybrid and Ensemble Methods*. *August*, 1–10. https://doi.org/10.20944/preprints201908.0203.v1

Bartneck, C., Lyons, M. J., & Saerbeck, M. (2017). *The Relationship Between Emotion Models and Artificial Intelligence*. 1–12. http://arxiv.org/abs/1706.09554

Chiorrini, A., Diamantini, C., Mircoli, A., & Potena, D. (2021). Emotion and sentiment analysis of tweets using BERT. *CEUR Workshop Proceedings*, *2841*(April). https://ceur-ws.org/Vol-2841/DARLI-AP_17.pdf.

Denemark, R. A. (2011). Teaching and Learning about Teaching and Learning. *International Studies Review*, *13*(3), 543–545. https://doi.org/10.1111/j.1468-2486.2011.01061.x

Derczynski, L., Ritter, A., Clark, S., & Bontcheva, K. (2013). Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. *International Conference Recent Advances in Natural Language Processing, RANLP*, *September*, 198–206 https://aclanthology.org/R13-1026.pdf.

Heimerl, F., Lohmann, S., Lange, S., & Ertl, T. (2014). Word cloud explorer: Text analytics based on word clouds. *Proceedings of the Annual Hawaii International Conference on System Sciences*, 1833–1842. https://doi.org/10.1109/HICSS.2014.231

Tang, D., Zhang, Z., He, Y., Lin, C., & Zhou, D. (2019). Hidden topic–emotion transition model for multi-level social emotion detection. Knowledge-Based Systems, 164, 426-435. http://wrap.warwick.ac.uk/111420/7/WRAP-hidden-topic-emotion-transition-model-detection-He-2018.pdf.

Jaiswal, S. (2019). Robust real-time emotion detection system using CNN architecture. *Neural Computing and Applications*, *4*. https://doi.org/10.1007/s00521-019-04564-4

Jaiswal, S., Virmani, S., Sethi, V., Partha, K. De, & Roy, P. (2018). *An intelligent recommendation system using gaze and emotion detection https://doi.org/10.1007/s11042-018-6755-1* .

Kim, H., Ben-othman, J., Cho, S., & Mokdad, L. (2019). *ACCEPTED FROM OPEN CALL A Framework for IoT-Enabled Virtual Emotion Detection in Advanced Smart Cities*. 1–7 http://people.uncw.edu/kimh/NETWORK2019.pdf .

Majeed, A., Mujtaba, H., & Beg, M. O. (2020). Emotion detection in roman urdu text using machine learning. *Proceedings - 2020 35th IEEE/ACM International Conference on Automated Software Engineering Workshops, ASEW 2020*, *September 2021*, 125–130. https://doi.org/10.1145/3417113.3423375

Minaee, S., Azimi, E., & Abdolrashidi, A. (n.d.). *Deep-Sentiment : Sentiment Analysis Using Ensemble of CNN and Bi-LSTM Models https://arxiv.org/pdf/1904.04206.pdf*.

Mosavi, A., Salimi, M., Ardabili, S. F., Rabczuk, T., Shamshirband, S., & Varkonyi-Koczy, A. R. (2019). State of the art of machine learning models in energy systems, a systematic review. *Energies*, *12*(7). https://doi.org/10.3390/en12071301

Serban, I. V., Sankar, C., Germain, M., Zhang, S., Lin, Z., Subramanian, S., Kim, T., Pieper, M., Chandar, S., Ke, N. R., Rajeswar, S., de Brebisson, A., Sotelo, J. M. R., Suhubdy, D., Michalski, V., Nguyen, A., Pineau, J., & Bengio, Y. (2018). *A Deep Reinforcement Learning Chatbot (Short Version)*. 1–40. http://arxiv.org/abs/1801.06700

Sinan, M., & Kayaalp, F. (2021). *Sentiment Analysis on Social Media Reviews Datasets with Deep Learning Approach*. *4*(1). https://doi.org/10.35377/saucis.04.01.833026

Vitiugin, F., & Barnab, G. (2022). *Emotion Detection for Spanish by Combining LASER Embeddings , Topic Information , and Offense Features Emotion Detection for Spanish by Combining LASER Embeddings , Topic Information , and Offense*

Features. *January https://www.researchgate.net/profile/Fedor-Vitiugin/publication/358002713_Emotion_Detection_for_Spanish_by_Combining_LASER_Embeddings_Topic_Information_and_Offense_Features/links/61eabdce5779d35951c41656/Emotion-Detection-for-Spanish-by-Combining-LASER-Embeddings-Topic-Information-and-Offense-Features.pdf*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., … Rush, A. (2020). *Transformers: State-of-the-Art Natural Language Processing*. 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6

Yu, S., Chen, Y., & Zaidi, H. (2020). *AVA : A Financial Service Chatbot based on Deep Bidirectional Transformers*. *https://arxiv.org/pdf/2003.04987.pdf*.

Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(4). https://doi.org/10.1002/widm.1253