2023 13th International Conference on Software Technology and Engineering (ICSTE) | 979-8-3503-7147-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICSTE61649.2023.00029

Using Hybrid CNN-LSTM Model for Sentiment Analysis of COVID-19 Tweets

Mary Jane C. Samonte *
School of Information Technology
Mapúa University
Makati City, Philippines
mjcsamonte@yahoo.com

Aderieyan Timothy G. Dela Rosa
School of Information Technology
Mapúa University
Makati City, Philippines
atgdr2000@gmail.com

Lance Joshua C. Rivera
School of Information Technology
Mapúa University
Makati City, Philippines
lancejoshrivera@gmail.com

John Shadrach E. Silo
School of Information Technology
Mapúa University
Makati City, Philippines
jshadsilo@gmail.com

***Abstract*— The COVID-19 pandemic has affected many aspects of everyone's lives, causing problems in health and the economy. People often share their sentiments about the pandemic on social media, particularly on platforms like Twitter. Sentiment analysis involves using various pre-processed tweets and natural language processing (NLP) techniques to classify them as positive, neutral, or negative. It is a demanding task. Platforms with microblogging features have played a crucial role in disseminating information, such as news and emergencies. These data types are rich in sentiments that contribute to the necessary awareness of safety and preparedness. Previous studies have demonstrated that Twitter usage during disasters increases situational awareness and improves disaster response through the tweets of affected users. However, sometimes a single model is insufficient to achieve desired results, necessitating the creation of a hybrid model. This study aimed to develop a sentiment analysis model using a hybrid CNN-LSTM framework to analyze sentiments in tweets during the COVID-19 pandemic. The model utilized a hybrid CNN-LSTM architecture that focused on predicting whether the sentiment was positive or negative. The developed hybrid CNN-LSTM model achieved a testing accuracy of 84.72% and outperformed standalone CNN and LSTM models.**

Keywords*— *Sentiment Analysis, Hybrid CNN-LSTM Model, Microblog, COVID-19.

I. INTRODUCTION

People worldwide with internet access have undoubtedly shared their thoughts and opinions about their lives during the pandemic. There are numerous social media platforms where people can easily express and share their thoughts and opinions. Twitter is one of the most popular choices, allowing users to interact with others through liking, commenting, and messaging [1]. Tweets, with their character limit, can be easily pre-processed and analyzed.

Sentiment refers to the subjective or objective description of a topic, which can be based on factual or false information. Sentiment analysis is an analytical approach used to analyze text and understand people's sentiments towards products, services, celebrities, or political figures [2]. It also extracts qualities such as emotions, attitudes, and sarcasm from sentences [3]. Sentiment analysis is one of the fastest-growing research topics since the 20th century. Twitter, in particular, has gained popularity for sentiment analysis in recent years [4]. Sentiments expressed by users can be classified into categories based on data size and document type using sentiment analysis.

Natural language processing (NLP) is a method used to extract information and it is employed for various purposes, including program activation through translation, responding to voice commands, and real-time data compression. Natural language processing is considered one of the methodologies for learning user sentiments from extracted tweets [5].

One known deep learning algorithm is convolutional neural network (CNN) which is effective in processing data with a grid-like topology [6]. Another technique is the long short-term memory (LSTM) which is a notable type of recurrent neural network (RNN) known for its ability to capture long-term dependencies [7]. The hybrid CNN-LSTM model combines CNN and LSTM algorithms. The CNN part effectively extracts high-level features using max-pooling and convolutional layers, while the LSTM part learns long-term dependencies in sentences [8].

Sentiment analysis is essential for analyzing COVID-19 tweets as it has become an active area of research due to the varied reactions and sentiments expressed online [9]. Governments monitor social media to track people's opinions on policies. Sentiment analysis utilizes various pre-processed tweets and NLP techniques to classify them as positive, neutral, or negative, which is a demanding task [10].

Platforms with microblogging features have played a crucial role in disseminating information, including news and emergencies. Such data is filled with sentiments that enhance the necessary safety and awareness. Previous studies have shown that Twitter usage during disasters increases situational awareness and improves disaster response through tweets from affected users. Authorities can also utilize Twitter for crisis management, disaster relief, and updates. Due to their significant contribution, numerous methods are being employed to uncover hidden insights from tweets [11].

While CNN is capable of detecting features locally in a multidimensional field, it lacks the ability to remember

979-8-3503-7147-5/23/$31.00 ©2023 IEEE
DOI 10.1109/ICSTE61649.2023.00029

previously read values. LSTM, on the other hand, excels at analyzing text by retaining past values. Combining LSTM with CNN can yield better performance in sentiment analysis [12].

This study utilized a hybrid CNN-LSTM model, which outperforms standard deep learning models and previous studies in terms of performance. While using a single model in sentiment analysis is sufficient in most cases, the usage of a hybrid model would be more advantageous to produce better results.

The objective of this study is to create a sentiment analysis model using a hybrid CNN-LSTM framework to analyze sentiments in tweets during the COVID-19 pandemic. Specifically, the study aims to achieve the following: 1) Determine the performance of the hybrid CNN-LSTM model in recognizing the sentiments of people regarding COVID-19 and create an accurate sentiment analysis performance; and 2) Utilize the hybrid CNN-LSTM model in determining sentiment using Word2vec for word embedding.

The study focuses on determining the polarity of sentiments among Twitter users regarding the COVID-19 pandemic. An existing dataset containing tweets from different geographical locations related to COVID-19 was used for this study. The "covid_19_Tweets" dataset was developed by Ankan Deria and it consists of 41,157 tweets; this can be found on Ankan Deria's Kaggle profile [13].

Word2vec and GloVe are popular algorithms for word embedding in natural language processing tasks [14]. While they serve the same purpose of representing words as dense vectors, they have differences in their approaches and advantages. In the field of NLP, the Word2vec is a technique that makes word embeddings; it takes in words from a huge corpus of texts for input and outputs their representation in vector form [15]. In this research, Word2vec was used for word embeddings instead of GloVe due to the effectiveness in capturing local context and rare words, better training performance in analyzing word analogies and similarity, and smaller memory footprint.

II. RELATED LITERATURE

Sometimes, using a single model is not sufficient to achieve the desired results, and creating a hybrid model becomes necessary [16]. According to a study by Gupta et al., [17] while a single machine learning method may be reliable in certain areas, different deep learning approaches have their own advantages and disadvantages. Although CNN alone can perform sentiment analysis, other related studies indicate that its performance is consistently inferior to that of a hybrid model. Initially designed for image recognition, CNN has proven to be adaptable for a wide range of tasks [18].

Various deep learning algorithms have been employed in related studies. Srinidhi proposed a hybrid model called MaLSTM (Manhattan LSTM), which combines an RNN and LSTM fused with SVM for sentiment classification [19]. Essabery et al. [20] also introduced a hybrid model that incorporates C4.5 decision tree, CNN, and FRBS. Additionally, Kurniasari et al. utilized recurrent neural network (RNN) in their study [21]. These studies have demonstrated superior performance compared to other models selected from the respective literature. This positive impact on sentiment analysis provides more options and approaches for conducting sentiment analysis. Incorporating new models can yield better results compared to using traditional ones.

It's important to note that the choice between Word2vec and GloVe depends on the specific task, the available training data, and other factors. GloVe also has its own advantages, such as capturing global word co-occurrence statistics and performing well on certain semantic tasks. Therefore, it's recommended to experiment with both algorithms and evaluate their performance for the particular use case at hand.

Capturing local context: Word2vec, particularly the Skip-gram model, is known for capturing local context information effectively [22]. It focuses on predicting the surrounding context words given the current word. This approach is beneficial when the local context plays a crucial role in determining word meanings and relationships.

Better with rare words: Word2vec tends to perform better than GloVe when dealing with rare words or words with limited occurrences in the training data [23]. It can learn more meaningful representations for infrequent words by leveraging the local context information.

Word analogies and similarity: Word2vec is often praised for its ability to perform well on word analogy tasks and capturing word similarities [24]. It can successfully identify relationships between words, such as "king - man + woman = queen." These analogical reasoning capabilities make Word2vec useful in tasks like word similarity measurement and word analogy completion.

Efficient training: Word2vec is relatively efficient in terms of training time compared to GloVe [25]. It can be trained faster on large corpora due to its simpler architecture and the use of negative sampling techniques.

Smaller memory footprint: Word2vec typically produces smaller word embedding models compared to GloVe [26]. The resulting vectors are usually of lower dimensionality, which reduces the memory footprint required to store and utilize the embedding.

Sentiment analysis is a powerful tool that can predict decision outcomes. During the creation of various COVID-19 vaccines such as Pfizer, Moderna, and AstraZeneca, people express sentiments and opinions about the efficacy and safety of the vaccines on Twitter [27]. Sentiments are collected, preprocessed using natural language processing (NLP), and analyzed using KNN classification. Two important factors in sentiment analysis are polarity and subjectivity. The results show that Pfizer and Moderna received the most positive sentiment from the public, with 47.29% and 46.16% respectively, while AstraZeneca had only 40.08%. This information can assist the government in providing vaccines that people trust.

There are several literatures proved that hybrid algorithms outperformed single ones. In the research made by Dang et al. [28] they experimented by combining SVM, CNN, and LSTM with the use of BERT and Word2vec across eight datasets

composed of reviews and tweets, the experiment showed that the dependability of hybrid models outperformed the tested sentiment analysis models. Fusing SVM with deep learning models produced better results compared to using standalone models for sentiment analysis. The experiment also concluded that the efficiency of the algorithm depends on the quality and characteristics of the dataset.

The LSTM- MCNN-decoder model outperformed all other models, which has an encoder-decoder framework, was developed by Chen et al [29]. It proved that CNN-LSTM-based method using pre-trained embeddings automatically learn features for sentiment analysis of labeled positive or negative reviews or opinions based on the study of Tyagi et al. [30]. The proposed CNN-LSTM method outperformed baseline machine learning methods, achieving an accuracy of 81.20% on the dataset.

In this research, we modified our process by using Word2Vec for word embeddings instead of GloVe which was already used in Tyagi's study and Sosa's study [30] [31]. This study also compared the proposed hybrid model against standalone deep learning models (CNN and LSTM) while Tyagi's study compared their proposed model against machine learning models (SVM, Random Forest, etc.) [30].

III. Methodology

After the dataset has been gathered, it went through data cleaning. After preprocessing the dataset, an array named "tweets_split" was made which then split and appended the tweets where the pre-trained Word2vec model was loaded (GoogleNews-vectors-negative300). The tweets were then transformed into a series of numeric values so that they can be inserted to the hybrid model. The data was split by 70% for training while 30% for testing. The 70:30 split for the training and testing data was chosen because it is the ideal split due to the size of the dataset used. An 80:20 split would be more ideal for larger datasets. The dataset is also fairly large that is why cross validation is not utilized; a simple split was sufficient. The hybrid CNN-LSTM model starts its training at five (5) epochs. Training the hybrid model at ten (10) and fifteen (15) epochs was also done and the researchers compared its results against the 5 epochs hybrid model. After training the hybrid model, a separate standalone CNN and standalone LSTM model were also trained and they share the same values as the hybrid model. The purpose of this is to fairly compare their performance against the hybrid CNN-LSTM model. After the hybrid model's training, evaluation of the results was made. The applied framework's flow would go like this: Data gathering → Data Pre-Processing → Modelling → Evaluation → Comparison. The visualization of this study's framework is shown in Figure 1.

The dataset is multivariate which consists of six (6) columns which are UserName, ScreenName, Location, TweetAt, OriginalTweet, and Sentiment. The column OriginalTweet contains the tweets in their original form when the tweet was posted. The dataset is already pre-annotated. Only the columns OriginalTweet and Sentiment are utilized since the other columns are not necessary for performing the sentiment analysis. The "covid_19_Tweets" dataset labeled with 5 categories: Extremely Positive, Positive, Neutral, Negative, and Extremely Negative. The number of observations in the dataset are shown in Table 1. There were 6,624 tweets labeled as Extremely Positive, 11,422 tweets as Positive, 7,713 tweets labeled as Neutral, 9,917 tweets as Negative, and 5,481 tweets were labeled as Extremely Negative. The labeled Neutral tweets were removed and the Extremely Positive and Extremely Negative were changed to Positive and Negative respectively. The number of Positive tweets is now 18,046 and 15,398 for the Negative tweets which gives a total of 33,444 tweets [13].

The tweets in the dataset were cleansed by removing the @mentions, hashtags, retweets (RTs), hyperlinks, special characters, and numbers. Stop words were removed and the tweets were turned into lowercase. After the data cleaning, the tweets that are labeled Positive and Negative were transformed into 1.0 and 0.0 respectively which has a datatype of int64. An array named "tweets_split" was made which tokenized and appended the tweets. GoogleNews-vectors- negative300.bin.gz was used as the Word2vec model which had a binary of true and a limit of 100,000. The limit is set to 100,000 because it would be very time intensive if we were to load the complete pre-trained Word2vec model. Maximum length of tweet was set to 50 because 50 is the length of the longest tweet in the dataset and a pad sequence was made which had a shape of 33,444 by 50. An embedding layer was made with an input dimension of 100,000, an output dimension of 300, and an input length of 50.

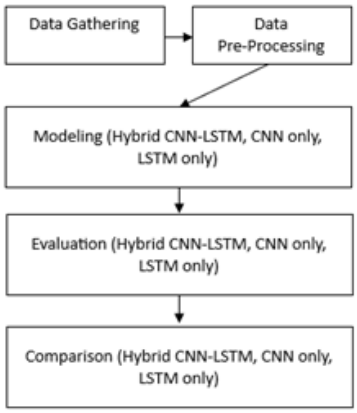

Fig. 1: The Conceptual Framework of the Study.

TABLE I. Annotated Dataset

Number of Tweets	Annotated Sentiment
6,624	Extremely Positive
11,422	Positive
7,713	Neutral
9,917	Negative
5,481	Extremely Negative

The hybrid CNN-LSTM model is made up of an initial convolution layer which will take an input which is the word embeddings. The word embeddings will come from the

Word2vec model. Next, the output will be pooled to a dimension which is smaller which will become the input for the LSTM layer. The intuition in this hybrid model is that the convolution layer will get features locally and the LSTM layer will utilize the ordering of the features to discover the order of text of the input. The parameters used for the model is shown in Table 2. The indicated values were chosen so that our hybrid model would not end up being so complicated and at the same time, would not look too simple. Other than that, these values are appropriate for the dataset that was used. The Conv1D filters is set to sixteen (16) for creating 16 different filters with each having the length equal to the kernel size. Kernel size was set to three (3) for the 1D convolution window's length. Batch size is set to sixty-four (64) so it can read 64 tweets at once during model training. Pool size is set to two (2) for the max pooling window size. Dropout was set to 0.5 because the chances of dropping nodes is 1 in 2 and it is evident in related studies that setting the dropout to 0.5 is the most ideal [29] [30] [31].

TABLE II. SET PARAMETERS OF CNN-LSTM MODEL.

Processes	Values
Epoch	5
Filters	16
Kernel Size	3
Batch Size	64
Pool Size	2
Dropout	0.5

The model is sequential with the embedding layer as the first layer. The embedding layer has an output shape - (None, 50, 300), with 50 being the maximum length of tweet and 300 being the syn0.shape [1] of the Word2vec model. A spatial dropout of 0.25 was placed after it and it has the same output shape as the embedding layer. Next is the Conv1D layer which has 16 filters, a kernel size of 3, relu for its activation, and casual for its padding; it has an output shape of (None, 50, 16) with 50 being the maximum length of tweet and 16 being the number of filters. A max pooling layer 1D was put next with a pool size of 2 with an output shape of (None, 25, 16) with 25 which was halved by this layer from 50 and 16 from the number of filters in Conv1D. A dropout of 0.5 was put after the max pooling layer which has the same output shape as the max pooling layer 1D. Then the LSTM layer is placed with 50 units; this has an output shape of (None, 50) with 50 being the number of LSTM units. Another dropout of 0.5 was also placed after the LSTM layer and it has the same output shape as the LSTM layer. A Flatten layer was placed after the dropout layer and it still has the same output shape as the previous layer. Last was the dense layer with sigmoid for its activation with an output shape of (None, 1) with 1 being the units set for the Dense layer. The model was then compiled with binary crossentropy for its loss since the hybrid model's function is binary classification, adam for its optimizer since adam offers more efficient weights for the neural network weight, and accuracy for its metrics. The summary of the hybrid CNN-LSTM model is shown below in Table 3.

TABLE III. SUMMARY OF THE HYBRID CNN-LSTM MODEL.

Layer (type)	Output Shape	Param #
Embedding	(None, 50, 300)	30,000,000
SpatialDropOut1D	(None, 50, 300)	0
Conv1D	(None, 50, 16)	14,416
MaxPooling1D	(None, 25, 16)	0
Dropout	(None, 25, 16)	0
LSTM	(None, 50)	13,400
Dropout	(None, 50)	0
Flatten	(None, 50)	0
Dense	(None, 1)	51
Total Params:	**30,027,867**	
Trainable Params:	**30,027,867**	
Non-Trainable Params:	**0**	

IV. RESULTS AND DISCUSSION

A. *Hybrid CNN-LSTM Model*

The training dataset was set to 70% while the testing dataset was set to 30%. The batch size was set to 64 and the model was trained in 5 epochs. As part of the analysis, we show training and validation behavior and loss of our model performance in Figure 2 and Figure 3. As shown in Figure 2, the hybrid model's training accuracy (blue) begins at 68.69% and works its way up to 96.79% while testing accuracy (orange) begins at 81.79% and works its way up to 84.72%. While in Figure 3, the hybrid model's training loss (blue) begins at 58.20% going down to 11.00% at the last epoch while the testing loss (orange) begins at 42.62% and ends at 46.52%. The source of error could come from what's called bias error where it is said that bias error occurs when only a few features are utilized in training the model. It is a fact that our hybrid model is not a complicated one and it only needs the tweet and sentiment but it has been proven in Figure 2 that our hybrid model fits well and is neither underfit nor overfit. The hybrid CNN-LSTM model's accuracy plot and loss plot is summarized in the form of a table shown below in Table 4.

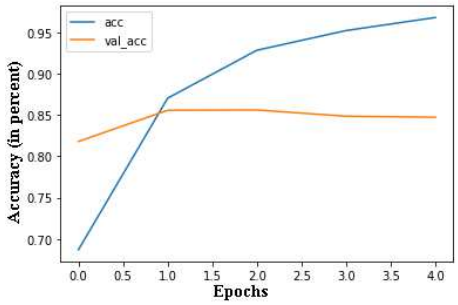

Fig. 2. Hybrid CNN-LSTM model accuracy plot

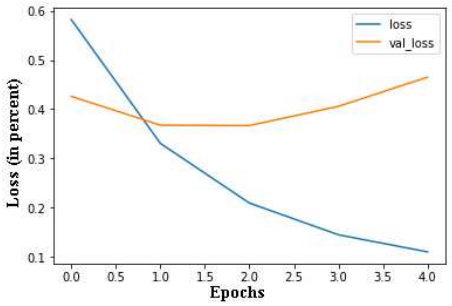

Fig. 3. Hybrid CNN-LSTM model loss plot.

TABLE IV. ACCURACY RESULT OF THE HYBRID CNN-LSTM MODEL.

Epoch	Training Loss	Training Accuracy	Testing Loss	Testing Accuracy
1	58.20%	68.69%	42.46%	81.79%
2	33.06%	87.03%	36.76%	85.57%
3	20.92%	92.83%	36.68%	85.60%
4	14.47%	95.21%	40.58%	84.84%
5	11.00%	96.79%	46.52%	84.72%

In the confusion matrix table for True Negative, 3,896 were predicted Negative and 745 were predicted Positive. For the True Positive, 4,605 were predicted Positive and 788 were predicted Negative. This means that out of the 10,034 tweets in the testing dataset, it correctly predicted the sentiment of 8,501 tweets. The confusion matrix is shown below in Table 5.

TABLE V. CONFUSION MATRIX OF THE HYBRID CNN-LSTM MODEL.

	Predicted Negative	Predicted Positive	Total
True Negative	3,896	745	4,641
True Positive	788	4,605	5,393
Total	4,684	5,350	10,034

Next is the evaluation results (Precision, Recall, and F1 Score) of the hybrid model. Precision is the number of correctly predicted positive/negative labels divided by the total number of predicted positive/negative labels. Recall is the number of correctly predicted positive/negative labels divided by the total number of actual positive/negative labels. F1 score is the mean of precision and recall. Support is the number of actual occurrences of the class from the test dataset. Positive and negative in this case are generic names for the predicted classes. The rows Negative and Positive are generic labels for the predicted classes. The macro average is the average of each class related to precision, recall, and f1 score while the weighted average is calculated by getting the average of all class's F1 scores while considering their support. The table for the Precision, Recall, and F1 Score of the hybrid model is shown below in Table 6.

TABLE VI. EVALUATION RESULTS OF THE HYBRID CNN-LSTM MODEL.

	Precision	Recall	F1-Score	Support
Negative	83%	84%	84%	4,641
Positive	0.86%	0.85%	86%	5,393
Accuracy			85%	10,034
Macro Avg.	85%	85%	85%	10,034
Weighted Avg.	85%	85%	85%	10,034

In terms of the hybrid model predicting new sentence inputs, it performs well when the sentence input is related to the topic of COVID-19, but it sometimes does incorrect prediction when the sentence input is not related to the topic of COVID-19. This is due to the fact that the dataset used is composed of COVID-19 Tweets and the hybrid model was trained from that. Models trained with datasets with no specific topic (ex. random Tweets) may be able to predict more correctly since it is not limited to a specific topic.

Five (5) epochs already seemed to be the most ideal since it already gained good results in accuracy and loss; testing anything higher than five epochs only resulted to further rising of testing loss and decrease in testing accuracy. Due to that, comparing 5 epochs against 10 and 15 epochs was sufficient. The test for 10 and 15 epochs were compared against 5 epochs. Random seed of twenty-four (24) was set to all tests so the shuffling of data is the same.

With 10 epochs, it's shown in Figure 4 that the hybrid model's training accuracy (blue) begins at 68.35% and works its way up to 98.41% while testing accuracy (orange) begins at 82.13% and ends at 82.78%. In Figure 5, the hybrid model's training loss (blue) begins at 58.47% going down to 5.15% at the last epoch while the testing loss (orange) begins at 42.44% and ends at 64.45%. The 10 epoch hybrid model's accuracy plot and loss plot is summarized in the form of a table shown below in Table 7.

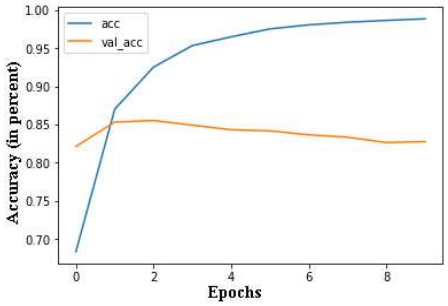

Fig. 4. Hybrid model with 10 epochs accuracy plot.

TABLE VII. ACCURACY RESULT (10 EPOCHS) OF THE HYBRID CNN-LSTM MODEL.

Epoch	Training Loss	Training Accuracy	Testing Loss	Testing Accuracy
1	58.47%	68.35%	42.44%	82.13%
2	33.43%	87.01%	35.80%	85.33%
3	21.35%	92.53%	36.20%	85.53%
4	14.72%	95.35%	41.52%	84.92%
5	11.31%	96.49%	47.86%	84.32%
6	8.90%	97.54%	46.75%	84.17%
7	7.44%	98.06%	53.60%	83.66%
8	6.41%	98.41%	57.10%	83.35%
9	5.54%	98.65%	63.49%	82.65%
10	5.15%	98.86%	64.45%	82.78%

With fifteen (15) epochs, the result is shown in Figure 5. The model's training accuracy (blue) begins at 67.64% and works its way up to 99.37% while testing accuracy (orange) begins at 82.22% and ends at 82.40%. The 15 epoch hybrid model's accuracy plot and loss plot is summarized in the form of a table shown below in Table 8.

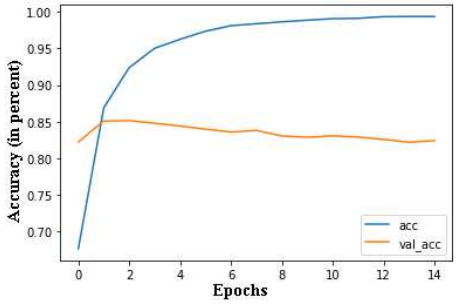

Fig. 5. Hybrid model with 15 epochs accuracy plot.

TABLE VIII. ACCURACY RESULT (15 EPOCHS) OF THE HYBRID CNN-LSTM MODEL.

Epoch	Training Loss	Training Accuracy	Testing Loss	Testing Accuracy
1	58.77%	67.64%	41.80%	82.22%
2	33.70%	86.89%	36.21%	85.09%
3	21.75%	92.39%	38.53%	85.14%
4	15.36%	95.04%	41.48%	84.78%
5	11.70%	96.25%	46.12%	84.40%
6	9.17%	97.36%	49.30%	83.97%
7	7.18%	98.12%	55.06%	83.59%
8	6.36%	98.39%	55.03%	83.81%
9	5.64%	98.65%	57.55%	83.05%
10	4.98%	98.87%	64.02%	82.87%
11	4.48%	99.08%	65.76%	83.06%
12	4.16%	99.12%	63.34%	82.90%
13	3.61%	99.35%	66.85%	82.58%
14	3.37%	99.37%	66.56%	82.19%
15	3.45%	99.37%	67.37%	82.40%

In the confusion matrix shown in Table 9, for ten (10) epochs in True Negative, 3,703 were predicted Negative and 938 were predicted Positive. For the True Positive, 4,603 were predicted Positive and 790 were predicted Negative. This means that out of the 10,034 tweets in the testing dataset, it correctly predicted the sentiment of 8,306 tweets. The evaluation results are shown in Table 10.

TABLE IX. CONFUSION MATRIX (10 EPOCHS) OF THE HYBRID CNN-LSTM MODEL.

	Predicted Negative	Predicted Positive	Total
True Negative	3,703	938	4,641
True Positive	790	4,603	5,393
Total	4,493	5,541	10,034

TABLE X. EVALUATION RESULTS (10 EPOCHS) OF THE HYBRID CNN-LSTM MODEL.

	Precision	Recall	F1-Score	Support
Negative	82%	80%	81%	4,641
Positive	83%	85%	84%	5,393
Accuracy			83%	10,034
Macro Avg.	83%	83%	83%	10,034
Weighted Avg.	83%	83%	83%	10,034

In the confusion matrix shown in Table 11, for fifteen (15) epochs in True Negative, 3,714 were predicted Negative and 927 were predicted Positive. For the True Positive, 4,554 were predicted Positive and 839 were predicted Negative. This means that out of the 10,034 tweets in the testing dataset, it correctly predicted the sentiment of 8,268 tweets. The evaluation results are shown in Table 12.

TABLE XI. CONFUSION MATRIX (15 EPOCHS) OF THE HYBRID CNN-LSTM MODEL.

	Predicted Negative	Predicted Positive	Total
True Negative	3,714	927	4,641
True Positive	839	4,554	5,393
Total	4,553	5,481	10,034

TABLE XII. EVALUATION RESULTS (15 EPOCHS) OF THE HYBRID CNN-LSTM MODEL.

	Precision	Recall	F1-Score	Support
Negative	82%	80%	81%	4,641
Positive	83%	84%	84%	5,393
Accuracy			82%	10,034
Macro Avg.	82%	82%	82%	10,034
Weighted Avg.	82%	82%	82%	10,034

The hybrid CNN-LSTM model that was trained in five (5) epochs had the best validation accuracy of 84.72% versus ten (10) epochs which was 82.78% and 15 epochs which was 82.40%. It was also shown in Table 7 and 8 that the validation loss of the hybrid model with 10 and 15 epochs kept increasing further with more epochs which results in overfitting. At five (5) epochs, the validation loss is kept at 46.52%. At ten (10) and fifteen (15) epochs, they reached a higher training accuracy (98.86% and 99.37% respectively) than the five (5) epochs' 96.79% but the hybrid model with five (5) epochs is still better in terms of validation accuracy and validation loss.

B. Convolutional Neural Network (CNN) Model Result

The results of the created standalone CNN and LSTM model is discussed and compared against the hybrid CNN-LSTM's performance. The standalone CNN and LSTM created in this study is basically just a stripped-down version of the hybrid model and still share the same values for a fair comparison. The summary of the CNN model and LSTM is shown below in Table 13 and Table 14.

TABLE XIII. SUMMARY OF THE CNN MODEL.

Layer (type)	Output Shape	Param #
Embedding	(None, 50, 300)	30,000,000
SpatialDropOut1D	(None, 50, 300)	0
Conv1D	(None, 50, 16)	14,416
MaxPooling1D	(None, 25, 16)	0
Dropout	(None, 25, 16)	0
Flatten	(None, 400)	0
Dense	(None, 1)	401
Total Params:	**30,014,817**	
Trainable Params:	**30,014,817**	
Non-Trainable Params:	**0**	

TABLE XIV. SUMMARY OF THE LSTM MODEL.

Layer (type)	Output Shape	Param #
Embedding	(None, 50, 300)	30,000,000
SpatialDropOut1D	(None, 50, 300)	0
LSTM	(None, 50)	70,200
Dropout	(None, 50)	0
Flatten	(None, 50)	0
Dense	(None, 1)	51
Total Params:	**30,070,251**	
Trainable Params:	**30,070,251**	
Non-Trainable Params:	**0**	

As shown in Figure 6, the CNN model's training accuracy (blue) begins at 60.84% and works its way up to 95.81% while testing accuracy (orange) begins at 77.47% and ends at 84.29%. The CNN model's training loss (blue) begins at 65.50% going down to 14.17% at the last epoch while the testing loss (orange) begins at 50.91% and ends at 42.90%; this is shown in Figure 7. The CNN model's accuracy plot and loss plot is summarized in the form of a table shown in Table 15.

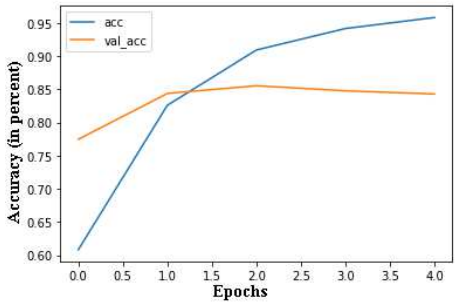

Fig. 6. CNN model accuracy plot.

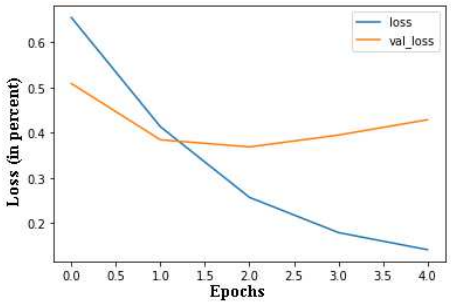

Fig. 7. CNN model loss plot.

TABLE XV. SUMMARY OF THE CNN MODEL ACCURACY RESULT.

Epoch	Training Loss	Training Accuracy	Testing Loss	Testing Accuracy
1	65.50%	60.84%	50.91%	77.47%
2	41.36%	82.59%	38.47%	84.37%
3	25.74%	90.91%	36.89%	85.53%
4	17.95%	94.15%	39.51%	84.77%
5	14.17%	95.81%	42.90%	84.29%

In CNN's confusion matrix shown in Table 16, for True Negative, 3,916 were predicted Negative and 725 were predicted Positive, as given in Table 16. For the True Positive, 4,542 were predicted Positive and 851 were predicted Negative. This means that out of the 10,034 tweets in the testing dataset, it correctly predicted the sentiment of 8,458 tweets. The evaluation results of CNN model are given in Table 17.

TABLE XVI. CONFUSION MATRIX OF THE CNN MODEL.

	Predicted Negative	Predicted Positive	Total
True Negative	3,916	725	4,641
True Positive	851	4,542	5,393
Total	4,767	5,267	10,034

TABLE XVII. EVALUATION RESULTS OF THE CNN MODEL.

	Precision	Recall	F1-Score	Support
Negative	82%	84%	83%	4,641
Positive	86%	84%	85%	5,393
Accuracy			84%	10,034
Macro Avg.	84%	84%	84%	10,034
Weighted Avg.	84%	84%	84%	10,034

C. Long-Short Term Memory Model Results

The LSTM model's training accuracy (blue) begins at 72.91% and works its way up to 98.07% while testing accuracy (orange) begins at 83.31% and ends at 83.81%, as shown in Figure 8. The LSTM model's training loss (blue) begins at 53.54% going down to 7.97% at the last epoch while the testing loss (orange) begins at 40.45% and ends at 54.72%, as given in Figure 9. The LSTM's model's accuracy plot and loss plot is summarized in the form of a table shown below in Table 18.

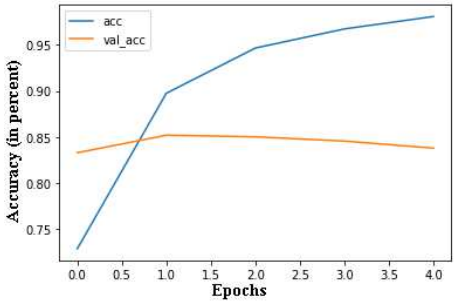

Fig. 8. LSTM model accuracy plot.

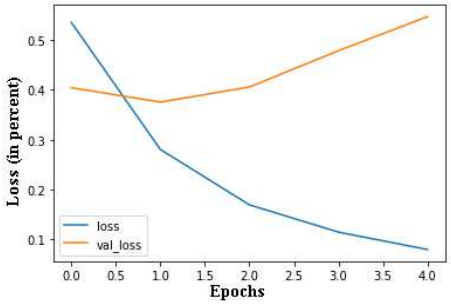

Fig. 9. LSTM model loss plot.

TABLE XVIII. SUMMARY OF THE LSTM MODEL ACCURACY RESULT.

Epoch	Training Loss	Training Accuracy	Testing Loss	Testing Accuracy
1	53.54%	72.91%	40.45%	83.31%
2	28.05%	89.75%	37.56%	85.21%
3	16.94%	94.66%	40.59%	85.03%
4	11.46%	96.72%	47.88%	84.57%
5	7.97%	98.07%	54.72%	83.81%

The LSTM's confusion matrix shown in Table 19, for True Negative, 3,975 were predicted Negative and 666 were predicted Positive. For the True Positive, 4,434 were predicted Positive and 959 were predicted Negative, as illustrated in Table 19. This means that out of the 10,034 tweets in the testing dataset, it correctly predicted the sentiment of 8,409 tweets. The evaluation results of LSTM model are given in Table 20.

TABLE XIX. CONFUSION MARTIX OF THE LSTM MODEL.

	Predicted Negative	Predicted Positive	Total
True Negative	3,975	666	4,641
True Positive	959	4,434	5,393
Total	4,934	5,100	10,034

TABLE XX. EVALUATION RESULTS OF THE LSTM MODEL.

	Precision	Recall	F1-Score	Support
Negative	81%	86%	83%	4,641
Positive	87%	82%	85%	5,393
Accuracy			84%	10,034
Macro Avg.	84%	84%	84%	10,034
Weighted Avg.	84%	84%	84%	10,034

V. CONCLUSION AND RECOMMENDATIONS

The summary of all training and testing results based on 5, 10 and 15 epochs are shown in Table 21, while the summary of all evaluation results is given in Table 22.

TABLE XXI. SUMMARY OF TRAINING AND TESTING OF THE HYBRID CNN-LSTM MODEL IN 5, 10, 15 EPOCHS.

Epochs	Training Loss	Testing Accuracy	Testing Loss	Testing Accuracy
5	11.00%	96.79%	46.52%	84.72%
10	5.15%	98.86%	64.45%	82.78%
15	3.45%	99.37%	67.37%	82.40%

TABLE XXII. SUMMARY OF EVALUATION RESULTS OF THE HYBRID CNN-LSTM MODEL IN 5, 10, 15 EPOCHS.

Number of epochs and labels	Precision	Recall	F1 Score
(5) Negative	83%	84%	84%
(5) Positive	86%	85%	86%
(10) Negative	82%	80%	81%
(10) Positive	83%	85%	84%
(15) Negative	82%	80%	81%
(15) Positive	83%	84%	84%

The performance of the CNN-LSTM hybrid model is best in five (5) epochs and beats the hybrid model that were trained in ten (10) and fifteen (15) epochs in terms of accuracy, precision, recall, and F1 Score. The hybrid model got a training accuracy of 96.79% and a testing accuracy of 84.72%. The five (5) epochs model had the better precision, recall, and F1 score over the ten (10) and fifteen (15) epochs, though they may have better training accuracy, their testing loss increases a lot which does result in overfitting.

Other deep learning techniques used for comparison against the hybrid model is a single CNN and LSTM model. The hybrid CNN-LSTM did perform slightly better than the CNN and LSTM models. The hybrid model got a testing accuracy of 84.72% which is higher than CNN's 84.29% and LSTM's 83.81%. Even though the hybrid model did not have the best training accuracy, it still performed better in the testing dataset. The hybrid model had the best precision for the Negative labels, the best recall in the Positive labels, and the best F1 score for Positive and Negative while LSTM had the best precision for the Positive labels and the best recall in the Negative labels. Even though the hybrid model performed a little better against the standalone CNN and LSTM, it is still better nonetheless. Shown in Table 23 are the training loss, testing loss, training accuracy, and testing accuracy of the CNN, LSTM, and the hybrid model. Shown in Table 24 are their precision, recall, and F1 Score.

TABLE XXIII. SUMMARY OF EVALUATION RESULTS OF ALL MODELS.

	Training Loss	Testing Accuracy	Testing Loss	Testing Accuracy
CNN	14.17%	95.81%	42.90%	84.29%
LSTM	7.97%	98.07%	54.72%	83.81%
Hybrid Model	11.00%	96.79%	46.52%	84.72%

TABLE XXIV. SUMMARY OF EVALUATION RESULTS OF ALL MODELS.

Name of model and labels	Precision	Recall	F1 Score
(CNN) Negative	82%	84%	83%
(CNN) Positive	86%	84%	85%
(LSTM) Negative	81%	86%	83%
(LSTM) Positive	87%	82%	85%
(Hybrid) Negative	83%	84%	84%
(Hybrid) Positive	86%	85%	86%

For future work, the researchers recommend using a larger dataset when training a CNN-LSTM model especially when the objective is to compare it to standalone deep learning techniques. It was shown in this study that the hybrid model was only slightly better than the standalone CNN and LSTM; the results may vary more when a larger dataset is used as this is apparent in related studies where they used a dataset containing millions of tweets [29] [30] [31]. The researchers also recommend that the dataset to be used to have a ratio that is closer to 50/50 on the positive and negative tweets since the ratio in this study is at 54/46 (18,046 Positive tweets and 15,398 Negative tweets) as a closer to 50/50 ratio can impact better on the model's performance. Lastly, the researchers recommend to future researchers to try using Word2Vec with a different limit to see if there are any differences when it comes to performance.

REFERENCES

[1] A.E. Azzaoui, S.K. Singh, and J.H. Park, "SNS big data analysis framework for COVID-19 outbreak prediction in smart healthy city." Sustainable Cities and Society, 71, p.102993, 2021.

[2] M. Wongkar. A. Angdresey. "Sentiment Analysis Using Naive Bayes Algorithm of The Data Crawler: Twitter." In IEEE Fourth International Conference on Informatics and Computing (ICIC) (pp. 1-5), 2019.

[3] M. Wankhade, A.C.S. Rao, and c. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges." Artificial Intelligence Review, 55(7), pp.5731-5780, 2022.

[4] G.A. Ruz, P.A. Henríquez, and A. Mascareño, "Sentiment analysis of Twitter data during critical events through Bayesian networks classifiers." Future Generation Computer Systems, 106, pp.92-104, 2020.

[5] Y. Kang, Z. Cai, C. W. Tan, Q. Huang, and H.Liu, "Natural language processing (NLP) in management research: A literature review." Journal of Management Analytics, 7(2), pp.139-172, 2020.

[6] Z. Shi, W. Yao, L. Zeng, J. Wen, J. Fang, X. Ai, and J. Wen, "Convolutional neural network-based power system transient stability assessment and instability mode prediction." Applied Energy, 263, p.114586, 2020.

[7] J. Qiu, B. Wang, and C. Zhou, "Forecasting stock prices with long-short term memory neural network based on attention mechanism." PloS one, 15(1), p.e0227222 2020.

[8] R. Yao, N. Wang, Z. Liu, P. Chen, and X. Sheng, "Intrusion detection system in the advanced metering infrastructure: a cross-layer feature-fusion CNN-LSTM-based approach." Sensors, 21(2), p.626, 2021.

[9] C.A Melton, O.A. Olusanya, N. Ammar, and A. Shaban-Nejad, "Public sentiment analysis and topic modeling regarding COVID-19 vaccines on the Reddit social media platform: A call to action for strengthening vaccine confidence." Journal of Infection and Public Health, 14(10), pp.1505-1512, 2021.

[10] R. Khan, P. Shrivastava, A. Kapoor, A. Tiwari, and A. Mittal, "Social media analysis with AI: sentiment analysis techniques for the analysis of twitter covid-19 data." J. Crit. Rev, 7(9), pp.2761-2774, 2020.

[11] J. S. Kwan. and K. H. Lim. 2020. Understanding Public Sentiments, B. Jang, M. Kim, G. Harerimana, S.U. Kang, and J.W. Kim, "Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism." Applied Sciences, 10(17), p.5841, 2020.

[12] A. Onan, "Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks." Concurrency and Computation: Practice and Experience, 33(23), p.e5909, 2021.

[13] A. Deria. "covid_19_Tweets." kaggle.com/datasets/ankanderia/covid-19-tweets

[14] D.C. Elton, D. Turakhia, N. Reddy, Z. Boukouvalas, M.D. Fuge, R.M. Doherty, and P.W. Chung, "Using natural language processing techniques to extract information on the properties and functionalities of energetic materials from large text corpora." arXiv preprint arXiv:1903.00415, 2019.

[15] W. Mumbi. "What is Word2Vec?". section.io/engineering-education/what-is-word2vec/ 2021.

[16] J. Drgoňa, J. Arroyo, I.C. Figueroa, D. Blum, K. Arendt, D. Kim, E.P. Ollé, J. Oravec, M. Wetter, D.L. Vrabie, and L. Helsen, "All you need to know about model predictive control for buildings." Annual Reviews in Control, 50, pp.190-232, 2020.

[17] N. Afroz, M. Boral, V. Sharma, and M. Gupta, "Sentiment analysis of COVID-19 nationwide lockdown effect in India." In IEEE 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS) (pp. 561-567), 2021.

[18] J. Tao, and X. Fang, "Toward multi-label sentiment analysis: a transfer learning-based approach." Journal of Big Data, 7, pp.1-26, 2020.

[19] H. Srinidhi, G.M. Siddesh, and K.G. Srinivasa, "A hybrid model using MaLSTM based on recurrent neural networks with support vector machines for sentiment analysis." Engineering and Applied Science Research, 47(3), pp.232-240, 2020.

[20] F. Es-sabery. K. Es-sabery. H. Garmani, and A. Hair, "Sentiment Analysis of Covid19 Tweets Using A MapReduce Fuzzified Hybrid Classifier Based On C4.5 Decision Tree and Convolutional Neural Network." In E3S Web of Conferences (Vol. 297, p. 01052). EDP Sciences, 2021.

[21] L. Kurniasari. A. Setyanto, "Sentiment Analysis using Recurrent Neural Network." In Journal of Physics: Conference Series (Vol. 1471, No. 1, p. 012018). IOP Publishing, 2020.

[22] E.M. Dharma, F.L. Gaol, H.L.H.S. Warnars, and B., Soewito, "The accuracy comparison among word2vec, glove, and fasttext towards convolution neural network (CNN) text classification." J Theor Appl Inf Technol, 100(2), p.31, 2022.

[23] P.L. Rodriguez, and A. Spirling, "Word embeddings: What works, what doesn't, and how to tell the difference for applied research." The Journal of Politics, 84(1), pp.101-115, 2022.

[24] S. Samtani, H. Zhu, and H. Chen, "Proactively identifying emerging hacker threats from the dark web: A diachronic graph embedding framework (d-gef)." ACM Transactions on Privacy and Security (TOPS), 23(4), pp.1-33, 2020.

[25] D. Katsaros, G. Stavropoulos, and D. Papakostas, "Which machine learning paradigm for fake news detection?" In IEEE/WIC/ACM International Conference on Web Intelligence (pp. 383-387), 2019.

[26] S. Tam, R.B. Said, and Ö.Ö. Tanriöver, "A ConvBiLSTM deep learning model-based approach for Twitter sentiment classification". IEEE Access, 9, pp.41283-41293, 2021.

[27] I. Garg, H. Hanif, N. Javed, R. Abbas, S. Mirza, M.A Javaid, S. Pal, R. Shekhar, and A.B. Sheikh, "COVID-19 vaccine hesitancy in the LGBTQ+ population: a systematic review." Infectious Disease Reports, 13(4), pp.872-887, 2021.

[28] C. N. Dang. M. N. Moreno-Garcia. F. Dela Prieta, "Hybrid Deep Learning Models for Sentiment Analysis." Complexity, pp.1-16, 2021.

[29] N. Chen. P. Wang. "Advanced Combined LSTM-CNN Model for Twitter Sentiment Analysis." In 2018 5th IEEE international conference on cloud computing and intelligence systems (CCIS) (pp. 684-687), 2018.

[30] V. Tyagi. A. Kumar. S. Das. "Sentiment Analysis on Twitter Data Using Deep Learning approach." In 2020 2nd IEEE International Conference on advances in computing, communication control and networking (ICACCCN) (pp. 187-190), 2021.

[31] P. Sosa. "Twitter Sentiment Analysis using combined LSTM-CNN Models", 2017.