I.J. Modern Education and Computer Science, **2024, 6, 20-39**
Published Online on December 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2024.06.02

Analyzing Sentiments on Twitter Using Deep Learning Techniques

Aditya Bhushan
Department of Computer Science and Engineering, United College of Engineering & Research, Prayagraj,211011, India
Email: adityabhushan2017@gmail.com
ORCID ID: https://orcid.org/0009-0003-2091-722X

Devanshi Dwivedi
Department of Computer Science and Engineering, United College of Engineering & Research, Prayagraj,211011, India
Email: devanshidwivedi83@gmail.com
ORCID ID: https://orcid.org/0009-0007-0554-9026

Ashutosh Kumar Singh*
Department of Computer Science and Engineering, United College of Engineering & Research, Prayagraj,211011, India
Email: ashuit89@gmail.com
ORCID ID: https://orcid.org/0000-0003-3922-4389
*Corresponding Author

Snehlata
Department of Computer Science and Engineering, United College of Engineering & Research, Prayagraj,211011, India
Email: loginsneha91@gmail.com
ORCID ID: https://orcid.org/0000-0003-0621-1805

Received: 28 February, 2024; Revised: 20 March, 2024; Accepted: 16 May, 2024; Published: 08 December, 2024

Abstract: In today's digital age dominated by social media, understanding public sentiment through Twitter analysis has become imperative. With a staggering 100 million active users on platforms like Twitter and an influx of 572,000 new accounts daily, the vast reservoir of user-generated content underscores the necessity for advanced sentiment analysis tools. This study delves into the realm of sentiment analysis techniques on Twitter, with a particular emphasis on employing Machine Learning (ML) methods. The proposed framework harnesses the power of Natural Language Processing (NLP) and Deep Learning architectures, specifically advocating for a synergistic blend of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. Additionally, it explores the efficacy of traditional ML algorithms such as Support Vector Machines (SVM), Random Forest, and Multi-Layer Perceptron (MLP) in this context. The study's findings illuminate diverse performance metrics across the employed models. While SVM exhibits moderate accuracy, it grapples with challenges in recall and F1-score for sentiment class 1. Conversely, the CNN-LSTM model emerges as a standout performer, boasting impressive accuracy rates of 97% and 98% respectively. Notably, this model excels in sentiment classification across all classes, underscoring its efficacy in discerning nuanced sentiment nuances within tweets. Furthermore, the study underscores the critical importance of judiciously selecting ML algorithms tailored to the intricacies of Twitter sentiment analysis. By leveraging advanced NLP techniques and deep learning architectures, researchers and practitioners can glean deeper insights into the dynamic landscape of public sentiment on social media platforms like Twitter. Such insights hold significant implications for diverse domains, including marketing, brand management, and public opinion analysis.

Index Terms: Sentiment Analysis, Convolutional Neural Network, Deep Learning, Sentiment Classification, Hybrid Models, Machine Learning

1. Introduction

During the age of social platforms supremacy, the analysis of sentiments, emotions conveyed in shared Tweets has become gradually crucial for comprehending common judgement and public sentiments, assessing brand reputation, together with analyzing trends in real-time. Amidst the continually increasing amount of user-created content, the requirement for efficient and successful sentiment analysis tools has strengthened. Social networking platforms has become a cause of various kinds of information and assists the formation, sharing, and knowledge transfer, thoughts, and multimedia content in the user group [1]. The analysis of sentiments in Tweets within the context of social platforms has transcended individual curiosity to become a vital tool for businesses, researchers, and policymakers. The continuous growth of user-generated content on social media platforms emphasizes the need for sophisticated sentiment analysis tools that can efficiently process and interpret this vast and dynamic source of information, providing valuable insights into public sentiments, brand perceptions, and societal trends. These online sites are created to facilitate human interaction as well as networking in internet spaces. As one of the most trending social media, twitter has minimum 100 million involved users [2]. Moreover, 572,000 fresh accounts has been formed on one day, meanwhile on average, 140 million twitter messages are sent daily. Important knowledge is often obscured behind tweeted updates and cannot be smoothly computed through automation [3]. Twitter is an ideal social networking sites for the retrieval of general public thoughts on a particular concern [4]. Shared contents on twitter is beneficial for sentiment analysis, like opinion mining or Natural Language Processing (NLP) [5].

With the continuous growth of user-generated content on social media platforms, the need for sophisticated sentiment analysis tools has become imperative [6]. These tools are designed to efficiently process and interpret the vast and dynamic information on twitter, providing valuable insights into public sentiments, brand perceptions, and emerging societal sentiments.

There are different techniques for sentiment analysis on twitter, Machine Learning (ML) is one of them. The basic framework of Tweet sentiment analysis includes several numbers of steps, as shown in Fig 1 provided the models based on Deep Learning (DL) architecture have attained impressive results in visual perception [7] and voice recognition in the last several years. Sentiment analysis aggregates based on the use of NLP, reviewing tweeted contents, together with biological data to automatically establish and estimate the emotions as well as sentiments of twitter feed [8]. The existing tasks of sentiment analysis are to influence the belief or sentiments of twitter data with positive, negative as well as neutral.

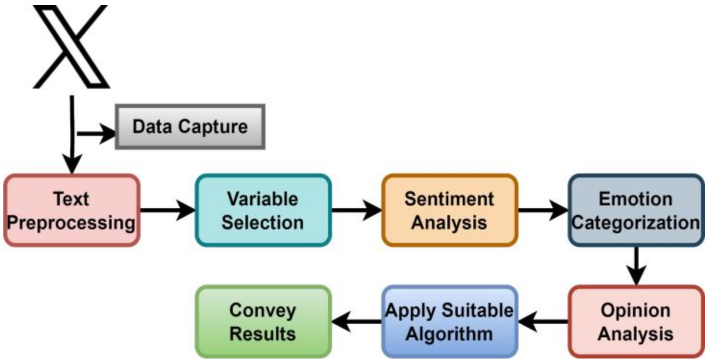

Fig. 1. Framework for Tweet sentiment analysis

Opinion mining is widely available to the public opinion and survey feedback, and also involving healthcare coding data that differs from marketing business to public duty [6]. It can be suitable to compose the textual information, graphic and video elements, vocal recognition, etc. Sentiment mining belongs to similar kind of twitter updates, after collecting this data initially it will divide the input into individual word or sentences, known as tokenization. A cinematic review is like a vocabulary which gives view about the movie and estimates the belief from the public as favorable or unfavorable. Based on this, we can easily determine the clues on whether to consider the movie. The impact of online reviews on marketing strategies cannot be overstated. Positive reviews can serve as powerful endorsements, driving sales and enhancing brand credibility. Conversely, negative reviews can undermine consumer trust and deter potential customers. Understanding the dynamics of online reviews is therefore essential for marketers seeking to leverage this form of social proof to their advantage.

ML depicts a vital task in analysis of sentiments, a NLP role that includes identifying the feelings or sentiments conveyed in textual content, for instance a shared Tweet, feedback, or comment. Supervised learning is a general method in sentiment assessment, where the procedure is trained on a labeled dataset consisting examples of written text with relevant sentiment tags like favorable, adverse, or indifferent [9]. The model learns to determine sequence as well as attributes in the input textual content that are representative of the sentiment classes through the repetitive methods of modifying its parameters rely on the labeled instances. After training, the model can be applied to evaluate novel, unseen text by estimating its sentiment based on the sequences it has learned while training.

In our proposed approach, we applied Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and compare with the other model like Support Vector Machine (SVM), Random forest, Multilayer Perceptron (MLP) and Bidirectional LSTM, which are kinds of supervised ML algorithm. As CNN can be utilized to written data by managing it as a one-dimensional series of word encodings. CNN excel at capturing hierarchical features within the text, enabling them to discern both local nuances and global patterns. They outperform at understanding hierarchical features of textual content, catching both local along with global traits [10]. LSTM are efficient in constructing the ongoing nature of sentences, forming them applicable for sentiment evaluation roles where the framework of words matters [11].

For two class or multiclass sentiment assessment, SVM can be used forming them efficient for roles where there are more than two sentiment classes [12]. Random forest can be employed to sentiment mining tasks, specifically when handling with structured traits or when transparency is crucial. In situation where the associations between input attributes are complex and may not be well-catched by simpler methods, MLP can be used. They are also efficient for both two-class as well as multiclass categorization roles. Bidirectional LSTM can utilize data from both earlier as well as subsequent phrases, enhancing the comprehension of phrase and interactions within the written material [13].

1.1. Application of sentiment analysis on twitter

Tweet sentiment classification has numerous utilizations across various domains because of the knowledge it can give to public beliefs, customer reviews, along with real-time trends. Analyzing emotions in clients tweeted posts permits companies to comprehend customer satisfaction, determine issues, and enhance the quality of their services. Marketers utilize feelings analysis of tweeted messages to assess the successful of their social networking sites [14]. Sentiment analysis is not limited to commercial applications; it also plays a crucial role in social and political discourse, helping to gauge public opinion on various issues and inform policy making. As technology continues to advance, the scope and impact of sentiment analysis are poised to expand, offering even greater opportunities for businesses and society at large.

1.2. Emergence of sentiment analysis in social media

With the rise of social media platforms, the analysis of sentiments expressed in Tweets has become increasingly vital for understanding public opinion, evaluating brand reputation, and tracking real-time trends. As social networking platforms continue to serve as hubs for information sharing and knowledge transfer among user groups, the need for efficient sentiment analysis tools has grown exponentially [1].

1.2.1 Evolution of sentiment analysis in social media

The evolution of sentiment analysis in social media has been propelled by the exponential growth of user-generated content across various platforms. Initially, sentiment analysis focused on understanding individual sentiments expressed in text data, but with the advent of social media, particularly platforms like twitter, the scope expanded to encompass collective sentiments on a larger scale. Despite these advancements, challenges remain in sentiment analysis, including the need to account for cultural and linguistic differences, sarcasm, irony, and context-dependent meanings in language. Researchers continue to explore novel techniques, such as DL and contextual embedding, to address these challenges and further improve the accuracy and robustness of sentiment analysis models [15].

1.2.2 Role of sentiment analysis in understanding public opinion

Sentiment analysis has become indispensable for understanding public opinion dynamics in real time. Social media platforms serve as digital arenas where individuals express their thoughts, emotions, and opinions on a wide range of topics. Analyzing these sentiments provides valuable insights into societal trends, emerging issues, and public reactions to events or policies. In addition to analyzing text data from social media platforms, sentiment analysis is also applied to other types of textual data, including customer reviews, news articles, and survey responses. By aggregating and analyzing sentiment data from diverse sources, businesses, governments, and researchers can gain valuable insights into consumer preferences, market trends, and public opinion on various issues.

1.2.3 Impact on brand perception and reputation management

For businesses and organizations, sentiment analysis is crucial for monitoring brand perception and managing reputation effectively [16]. By analyzing sentiments expressed in Tweets and other social media posts, companies can gauge public sentiment towards their products, services, or brand as a whole. This insight enables them to identify areas for improvement, address customer concerns promptly, and leverage positive feedback for marketing purposes.

1.3. Twitter sentiment analysis

Sentiment classification on tweeted updates is a well-liked and demanding NLP work that includes recognizing the emotions phrased in tweeted contents, which are brief, informal, and may include casual language [17]. Twitter opinion mining has several applications, including comprehending public beliefs, brand tracking, together with monitoring instantaneous feedback to events.

Further innovation in sentiment analysis on Twitter data could involve exploring transformer-based models like BERT (Bidirectional Encoder Representations from Transformers) or GPT (Generative Pre-trained Transformer) to leverage their state-of-the-art capabilities in natural language understanding. These transformer-based models have demonstrated remarkable performance in various NLP tasks, including sentiment analysis, due to their ability to capture complex contextual relationships and semantic nuances in text data. By pre-training on large corpora of text data, transformer models like BERT and GPT can learn rich representations of language that encode deep semantic understanding, allowing them to effectively capture sentiment nuances in tweets. Additionally, these models can be fine-tuned on domain-specific datasets, such as Twitter data, to adapt their learned representations to the specific characteristics of social media text, including informal language, abbreviations, and emojis. By incorporating transformer-based models into sentiment analysis pipelines for Twitter data, researchers can potentially achieve higher accuracy and robustness in sentiment classification, leading to more reliable insights into public opinions and sentiments expressed on the platform. Moreover, exploring the interpretability of transformer-based models and developing methods for explaining their predictions could further enhance trust and understanding in sentiment analysis applications, enabling stakeholders to make informed decisions based on the insights derived from Twitter data. Overall, leveraging transformer-based models represents a promising avenue for advancing sentiment analysis techniques on Twitter and unlocking deeper insights into online discourse and public sentiment.

1.3.1 Limitations in twitter sentiment analysis

The integration of Q-learning in the provided code is deemed impractical due to the inherent disparities between Q-learning's operation in discrete state and action spaces and the continuous nature of text classification tasks. The code is specifically designed for training and evaluating a neural network model tailored to supervised learning in text classification. Attempting to incorporate Q-learning would necessitate substantial modifications to both the problem formulation and the learning algorithm, making it unsuitable for the given context.

Sentiment mining on twitter introduces several challenges, primarily stemming from the platform's character limitations, leading to concise and unique features. This brevity poses difficulties in capturing the complete context and subtext of sentiment [18], while the inclusion of smileys and emoticons adds emotional content that can be contextually ambiguous.

The interpretative variability of these symbols makes it challenging to extract factual sentiment, as individuals may use the same symbols to convey different feelings, and not all emotions are effectively expressed through emojis.

Addressing these challenges is pivotal for enhancing the flexibility and robustness of emotion analysis models for twitter updates. The brevity of Tweets further complicates matters, as users often need to encapsulate their emotions within a limited character count. Traditional models face challenges in accurately capturing the subtleties and nuances of emotions in such constrained contexts. Overcoming this limitation may involve techniques like incorporating contextual embedding, leveraging external knowledge bases, or exploring transformer-based models to enhance the accuracy of emotion analysis in succinct twitter updates.

Table 1. Approach for Tweet sentiment analysis

Methods	Description	Advantages	Limitations
Rule-based	Depend on predefined rules, patterns and practical strategies to analyse the sentiments conveyed in a piece of text.	Easy to understand and transparent, also it requires few labelled data for analysis as compared to the approaches based on ML.	May be less flexible to modification in language trends or variations in emotions expressions.
Machine Learning	Supervised learning models like Naive Bayes, SVM are trained on classified dataset to categorize shared Tweets into emotions categories.	ML techniques, mainly DL models such as LSTM and CNN, can capture to sequence of dataset and adopt complex associations in tweeted posts.	Mainly rely on superiority along with accuracy of training data and it requires classified datasets which may be prolonged along with resource consuming to make, particularly for emotions analysis.
Mixed approaches	Integrates different methods, such as fusion of rule based methods with ML methods, to enhance overall effectiveness.	Mixed approaches can employ the strengths of multiple methods and improve model efficiency.	May enhance complexity and needs thorough integration.

1.3.2 Approaches for twitter sentiment mining

Strategies to assessing feelings and emotions on twitter include applying several approaches and methods to comprehend the emotional tone conveyed in Tweets as shown in Table 1.

1.4. Contributions and section introduction

The major contribution of this paper is as follows:

- The paper introduces a novel approach by combining CNN and LSTM for sentiment analysis on twitter data.
- Beyond the CNN-LSTM combination, the paper explored the effectiveness of various ML algorithms such as SVM, Random forest, CNN, LSTM, Bidirectional LSTM and MLP, providing a comprehensive evaluation of sentiment analysis techniques.
- The research emphasizes the practical significance of sentiment analysis beyond individual curiosity, shedding light on its implications for businesses, and researchers.
- The paper underscored its broad relevance and utility in different domains, highlighting its versatility and potential impact.
- The paper offers insights into the relative effectiveness of different ML algorithms in classifying sentiments in Tweets, providing valuable guidance for practitioners seeking.

The subsequent sections of this paper are structured as follows. Section 2, provides a comprehensive review of twitter sentiment analysis conducted by various authors. Moving forward, Section 3 outlines the detailed framework for different algorithm and proposed model is discussed in Section 4. Section 5 delves into the proposed work and presents the experimental aspects. Finally, Section 6 encapsulates the paper with a concluding summary.

2. Literature Review

Various researches on the application of the CNN-LSTM model for Tweet sentiment analysis are presented here. Grefenstette et al. [19] introduced a hybrid approach that merges the CNN-LSTM model for Tweet sentiment classification, the objective is to apprehend local patterns along with attributes within the input sentence.

Socher et al. [20] put forward with a LSTM framework on emotions classification, which enhance contextual depiction and performs better than many existing systems. Gandhi et al. [21] examine the application of deep neural networks for sentiment mining on tweeted updates. It analyzes the success of various frameworks in adopting local as well as global textual data.

Zheng et al. [22] initiate a novel mixed approach, that integrates a Convolutional Bidirectional LSTM Network and Attention-driven model for precise text categorization. Employing the robustness of Bidirectional LSTM along with CNN model. The approach outperformed usual learning techniques including individual CNN, neural networks, as well as combinations of them.

Huang et al. [23] combines attention-generating with emotion-enhancing factors within an integrated CNN-LSTM framework. The use of this method has the goal to strengthen the model’s capacity to gather significant data, especially personal details, and thus generate better emotional analysis findings. Aydin et al. [24] presents a mixed approach for Tweets sentiment evaluation that utilizes CNN with LSTMs. It intends to employ the advantages of the two frameworks to boost overall efficiency. The suggested framework gathers attributes from various deep neural network approaches for embedding words, integrates these characteristics, and evaluates sentences based on sentiments.

Zimbra et al. [25] describe an integrated approach for Tweet branding sentiment evaluation that combines n-gram evaluation with an evolving Artificial Neural Network (ANN). The technique is intended to improve the preciseness and applicability of sentiment assessment, especially when analyzing brand-related interactions on twitter. Kumar et al. [26] describes that neural networks used for sentiment evaluation are effective tools that can identify complicated connections as well as trends in written information, rendering them ideal for interpreting and categorizing opinions in a wide range of situations, including social networking sites and consumer feedback.

Dang et al. [27] suggests employing mixed DL algorithms for sentiment detection on social networking information, combining SVM, CNN, and LSTM. He performs thorough tests on eight written samples consisting comments and ratings to assess the performance of four mixed approaches for individual models.

Kaur et al. [28] the article aims to clarify well deep ML algorithms operate when used to analyze sentiment tasks using twitter information, giving insight on the advantages, constraints as well as potential improvements in this particular field. Vateekul et al. [29] evaluate the efficacy of deep neural network approaches in sentiment evaluation as employed with thai Tweets. The article may cover pretreatment procedures particular to Thai words, such as stemming, tokenization, or any other type of specific to a language processing needed for efficient sentiment evaluation. Roy et al. [30] in their work used three well-known DL algorithms for Tweets sentiment evaluation, highlighting input pretreatment to improve the model’s precision by reducing interference. A study concluded that the model based on Bidirectional Encoder Representations from Transformers (BERT) outperformed the other alternatives.

3. Design and implementation of algorithms

This section outlines the structural foundation for sentiment analysis on twitter, employing SVM, Random forest, CNN, LSTM, Bidirectional LSTM and MLP algorithms.

3.1. Support vector machine algorithm

One common algorithm used for sentiment analysis with SVM involves training the model using a linear SVM decision boundary shown in Fig 2. This decision boundary aims to classify Tweets into various sentiment classes, as illustrated in Algorithm 1. SVM can be employed in the realm of sentiment mining of tweeted updates [31], SVM can be leveraged as a supervised learning technique. The primary objective of sentiment analysis is to recognize the feelings or sentiment conveyed in a given textual content, like a tweeted message. SVM, an applicable as well as versatile approach, can be trained to categorize Tweets into various sentiment classes, like favorable, adverse, or indifferent. The Equation (1) represents the linear SVM decision boundary for sentiment analysis, where the goal is to classify Tweets into various sentiment classes such as favorable, adverse, or indifferent.

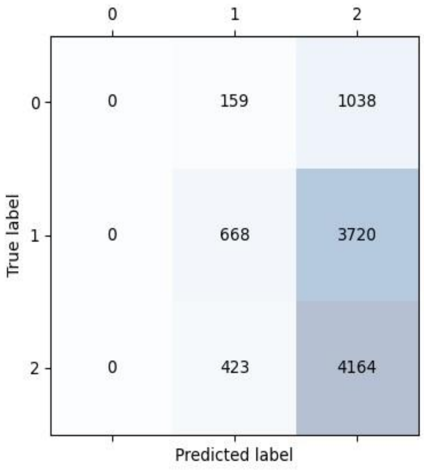

Fig. 2. Performance matrix of SVM

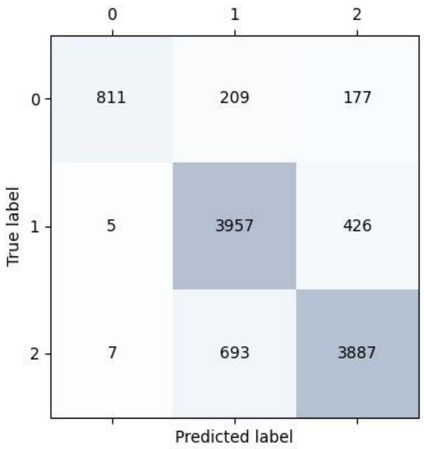

Fig. 3. Performance matrix of random forest

$$f(x) = sign(\sum_{i=1}^{n} \alpha_i y_i x^T x_i + b) \tag{1}$$

Algorithm 1. Support vector machine

```
Input: xtrain, ytrain, xval, yval, xtest
Output: ypred_test_svm
scaler ← StandardScaler()
xtrain_scaled ← scaler.fit transform(xtrain)
xval_scaled ← scaler.transform(xval)
x_test_scaled ← scaler.transform(xtest)
y_train_1d ← np.argmax(ytrain,axis=1)
y_val_1d ← np.argmax(yval, axis=1)
svm_model ← SVC(kernel='linear', C=1.0, random_state=42)
svm_model.fit(xtrain_scaled,y_train_1d)
ypred_test_svm ← svm_model.predict(xtest_scaled)
```

It's worth noting that the success of sentiment analysis using SVM depends on the quality of the training data, feature representation, and the chosen hyperparameters such as the regularization parameter in the SVM model. Additionally, preprocessing steps like tokenization, stemming, or lemmatization might be applied to enhance the effectiveness of sentiment analysis algorithms.

3.2. Random forest algorithm

Random forest is a method of ensemble learning that can be used for evaluation of sentiment uses [32]. The Algorithm 2 forms a grouping of decision trees amid training as well as provide the mode class (for categorization tasks) or the mean prediction (for estimation tasks) of the singular trees. The approach is widely employed in ML techniques because of its reliability, adaptability, as well as capacity to deal with massive data sets. It works extremely well in scenarios with difficult decision boundaries or a high range of attributes. The Equation (2) and Equation (3) represents the classification equation as well as regression equation of Random forest algorithm respectively.

$$y = mode(h1(x), h2(x), \ldots, hn(x)) \tag{2}$$

$$y = \frac{1}{n}\sum_{i=n}^{n} h_i(x) \tag{3}$$

On the other hand, the regression Equation (3), is more suitable for regression tasks, where the goal is to predict a continuous value. In the context of sentiment analysis, regression might not be the typical approach, as sentiment is often treated as a categorical variable (positive, negative, neutral). However, if sentiment scores are represented as continuous values (e.g., sentiment scores on a scale from 0 to 1), you could adapt this regression equation for sentiment analysis shown in Fig 3 by interpreting 'y' as the predicted sentiment score.

Algorithm 2. Random forest

Input: x_{train}, y_{train}, x_{val}, y_{val}, x_{test}
Output: $y_{\text{pred\_test\_rf}}$
scaler ← StandardScaler()
$x_{\text{train\_scaled}}$ ← scaler.fit transform(x_{train})
$x_{\text{val\_scaled}}$ ← scaler.transform(x_{val})
$x_{\text{test\_scaled}}$ ← scaler.transform(x_{test})
rf model ← Classifier(random state=42)
rf model.fit(x_train_scaled, np.argmax(y_{train}))
$y_{\text{pred\_test\_rf}}$ ← rf model.predict(x_$_{\text{test_scaled}}$)

3.3. CNN for text analysis

CNN have shown to be highly efficient and reliable techniques for analyzing written content. CNN were initially created for image analysis but have been effectively extended to identify structural as well as spatial associations throughout data sequences, which makes them useful for natural language processing difficulties [33]. In text mining, CNN perform by organizing filtering over text input patterns, allowing them to automatically identify local characteristics and trends. This trait is very useful for applications such as sentiment evaluation, sorting texts, and data extraction.

The Algorithm 3 provides the efficient and user-friendly tools for researchers and practitioners to leverage the power of CNN in developing advanced models for textual analysis. CNN capacity for identifying essential behavioral characteristics at many levels of abstraction enables them to identify ordered patterns in language, incorporating the two basic information and a high degree concepts as shown in Fig 4. This adaptability enables CNN ideal for managing variable length data and adapting to the changing character of text-based information. As an outcome, CNN have performed comparably to classical models, providing an adaptable yet effective alternative for textual analysis applications.

Algorithm 3. Convolutional neural network

Input: x_{train}, y_{train}, x_{val}, y_{val}, x_{test}
Output: $y_{\text{pred\_test\_cnn}}$
max features ← 20000
embed dim ← 100
num classes ← shape of y_{train}[1]
cnn model ← Sequential()
cnn model.add(Embedding(max features
embed dim, input length=x train.shape[1]))
cnn model.add(Conv1D(filters=32, kernel size=3
padding='same', activation='relu'))
cnn model.add(MaxPooling1D(pool size=2))
cnn model.add(Conv1D(filters=32, kernel size=3
padding='same', activation='relu'))
cnn model.add(MaxPooling1D(pool size=2)
cnn model.add(Flatten())
cnn model.add(Dense(128, activation='relu'))

3.4. Sequential analysis using LSTM

LSTM are essential for collecting serial relationships in textual data and resolving issues linked to recognizing information, connections, and linkages among ordered data. LSTM are specifically developed to detect long-term dependencies in sequential information. Conventional RNNs fail with this type of analysis because of the gradient vanishing issue, which leads them to discard knowledge over extended scores [34]. LSTM employs memory cells along with gates approaches to dynamically safeguard and modify knowledge, allowing them to preserve connections over large sections of the given input series. The implementation of LSTM is illustrated in Algorithm 4, and the corresponding matrix operations are depicted in Fig 5.

LSTM succeeds in identifying consecutive relationships in textual data by overcoming the difficulties of diminishing gradients and offering a reliable framework for preserving and upgrading data over long intervals. Their implementation has considerably enhanced the field of natural language processing, allowing for more precise and context-aware systems for an array of text-related tasks [35]. LSTM are utilized extensively in uses including sentiment evaluation, automated translation, titled entity identification, and modeling of languages.

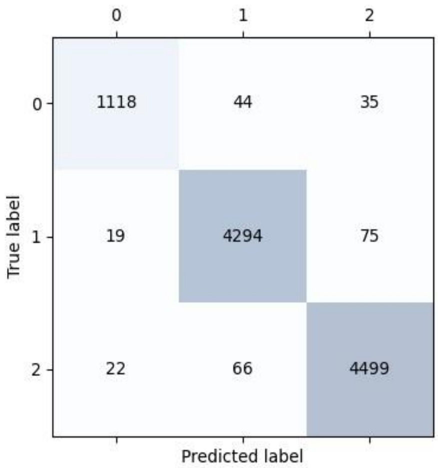

Fig. 4. Performance matrix of CNN

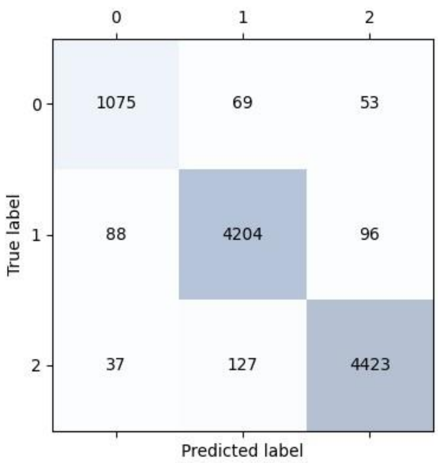

Fig. 5. Performance matrix of LSTM

Algorithm 4. Long short-term memory

Input: x_{train}, y_{train}, x_{val}, y_{val}, x_{test}
Output: $y_{\text{pred\_test\_lstm}}$
max_feature ← 20000
embed dim ← 100
num classes ← shape of $y_{\text{train}}[1]$
lstm_history ← lstm_model.fit(x_{train}, y_{train}, validation_data=(x_val, y_val), epochs=10, batch size=64, verbose=2)
$y_{\text{pred\_test\_lstm}}$ ← np.argmax(lstm_model.predict(x_test))

The LSTM has significantly enhanced various aspects of natural language processing, leading to the development of more precise and context-aware systems for a wide range of text-related tasks. For capturing the sentiment-bearing words' sequential nature LSTM excel allowing for a deeper understanding of the sentiment expressed in a given piece of text.

3.5. Bidirectional LSTM

A complicated recurrent neural network, or RNN, framework referred to Bidirectional LSTM is aimed at gathering pertinent details from patterns of information that were collected previously as well as in the future [36]. The method is particularly useful in sentiment evaluation, where it is essential to comprehend the entire phrase's context. The presented Algorithm 5 and the accompanying matrix operations illustrated in Fig 6 outline the procedural steps for implementing a Bidirectional LSTM model designed for sentiment analysis.

Algorithm 5. Bidirectional LSTM

Input: x_{train}, y_{train}, x_{val}, y_{val}, x_{test}
Output: $y_{\text{pred\_test}}$
max_features ← 20000
embed_dim ← 100
num_classes ← 3
bilstm_model ← Sequential()
bilstm_model.add(Embedding(max_features_embed_dim, input_length=x_train.shape[1]))
bilstm_model.compile(loss='categorical crossentropy' optimizer='adam', metrics=['accuracy'])
bilstm_model.fit($x_{\text{train},}$ y_{train})
validation_data=(x val, y val),
epochs=10, batch_size=64, verbose=2)
$y_{\text{pred test}}$ ← np.argmax(bilstm_model.predict(x test))

It starts by defining the input and output structures, specifying parameters such as the maximum number of features, embedding dimensions, and the number of classes. The embedding layer is added to convert input sequences into dense vectors of fixed size, and the model is compiled with categorical cross-entropy loss, Adam optimizer, and accuracy metrics.

It could be used to process new Tweets, and the final hidden state h_t may be used as a representation of the Tweet for sentiment classification. Equation (4) illustrates the forward pass of a single time step in a Bidirectional LSTM.

3.6. MLP algorithm

MLP is a kind of neural network that includes of multiple layers of interlinked nodes, or neurons. It is a feedforward network, implies that the data flow is acyclic flow [37].

Algorithm 6 Multilayer perceptron

Input: *x*train, *y*train, *x*val, *y*val, *x*test
Output: *y*pred_test
max_features ← 20000
embed_dim ← 100
num_classes ← shape of *y*train[1]
mlp_model ← Sequential()
mlp_model.add(Embedding(max features, embed_dim, input_length=x_train.shape[1]))
mlp_model.add(Flatten())
mlp_model.add(Dense(128, activation='relu'))
mlp_model.add(Dense(128, activation='relu'))
mlp_model.add(Dense(num_classes activation='softmax'))
mlp_model.compile(loss='categorical_crossentropy' optimizer='adam', metrics=['accuracy'])
mlp_history ← mlp_model.fit(x_train, y_train 17: validation_data=(x val, y val), 18: epochs=10, batch size=64, verbose=2)
y*pred*_test_mlp ← np.argmax(mlp_model.predict(x test))

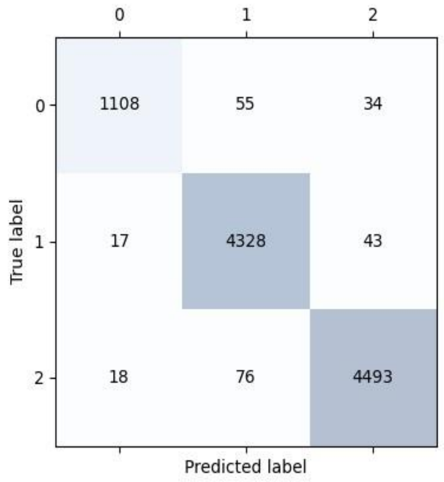

Fig. 6. Performance matrix of Bidirectional LSTM

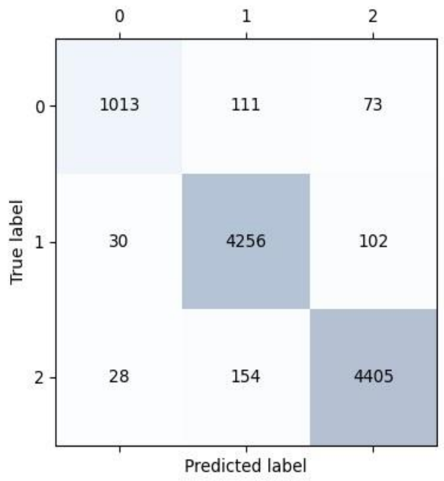

Fig. 7. Performance matrix of MLP

The Equation (5) represents elementary functions in a single layer of a MLP. Zˆ(i) denotes the weighted total of the input attributes for the i-th layer, Yˆ(i) denotes the outcome of i-th layer, attained by employing a non-linear activation function to the weighted total Zˆ(i).

$$Z^i = X^i \cdot W^i + b^i \tag{5}$$

As outlined in Algorithm 6 and depicted in the matrix illustrated in Fig 7, the steps detail the implementation of a MLP it is a feedforward network, meaning that the data flow is acyclic. The feedforward nature ensures that information travels in one direction, from the input layer through the hidden layers to the output layer, without forming cycles. The embedding layer is particularly useful for converting input sequences into a format suitable for neural network processing. The dense layers with non-linear activation’s enable the network to learn complex representations of the input data. The dropout layer helps prevent over fitting during training. The choice of activation functions and the model architecture can be adjusted based on the specific requirements of the task at hand.

4. Proposed integrated CNN-LSTM model

The hybridization of CNN with LSTM as shown in Fig 8, is an effective technique in text mining, providing an advantageous method for identifying the two individual and sequential trends in textual information. The combined approach attempts to capitalize on the positive aspects of both networks, enabling an extensive knowledge of the language’s complexities [38].

CNN, which were initially developed for processing pictures, are efficient for detecting local characteristics and sequences via convolutional filtering techniques. When employed to words, CNN may successfully detect short-term relationships throughout words or paragraphs. LSTM, a type of recurrent neural network, are capable of indicating dependency over time and successive data. This renders LSTM useful for comprehending the semantic connections among letters in a single phrase or throughout a lengthy series.

In the realm of analyzing sentiment on twitter wherein content frequently differentiates itself by briefness, unofficial behavior, along with the use of hashtag and emoticons the mixed CNN-LSTM approach is effective. Tweet data is rapidly changing and varied, necessitating a model that can accurately represent the specific mental state reflected by single posts (caught by CNN) while the deeper context as well sequencing relationships existing across several retweets.

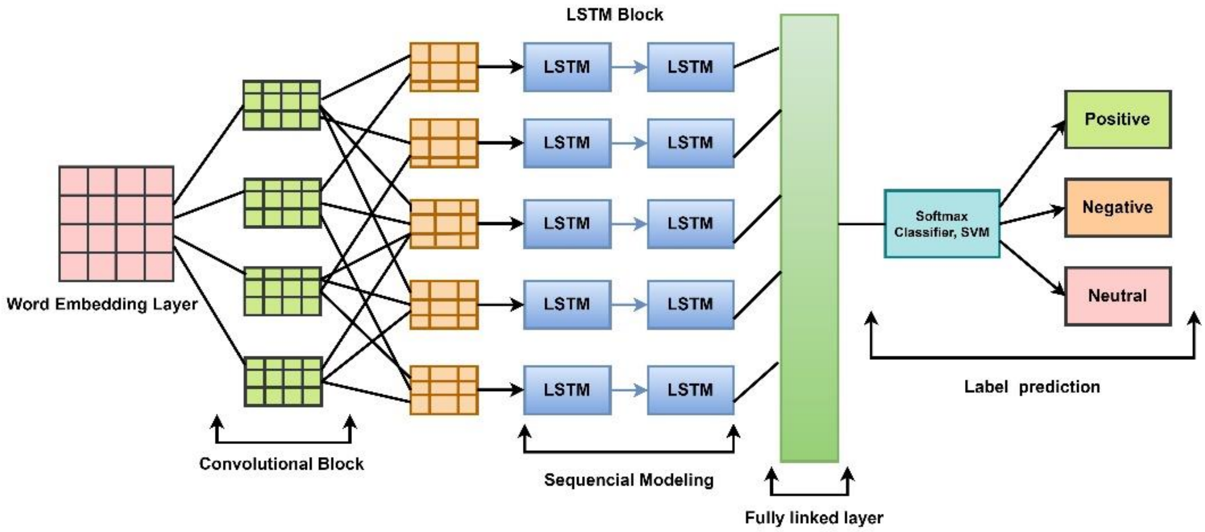

Fig. 8. Integrated CNN-LSTM framework for sentiment analysis

The benefits of applying mixed algorithms for sentiment evaluation on Tweets are numerous. Initially CNN are capable of accurately capturing local emotions among specific messages by considering variables such as particular words or search terms. Furthermore, LSTM assist by grasping the general feeling framework and chronological connections, as well as recognizing how emotions change over a succession of Tweets.

He integrated CNN-LSTM approach to analyzing sentiment on twitter offers an effective structure that considers the platform’s specific peculiarities. By expertly combining the benefits of CNN along with LSTM, this approach provides a sophisticated understanding of feelings, addressing both smooth local trends and the wider sequencing relationships present in the data from twitter.

4.1. CNN component for local feature extraction

The CNN component of the model is responsible for extracting local features and patterns from Tweet text. Through convolutional layers, the CNN identifies important words, phrases, and emoticons that contribute to the sentiment expressed within individual Tweets. This local feature extraction process enables the model to capture the nuanced aspects of sentiment within short, text-based messages.

4.2. LSTM component for temporal modeling

Complementing the CNN, the LSTM component of the model is designed to capture temporal dependencies and sequential relationships across Tweets as shown in Fig 9. By processing Tweet sequences over time, the LSTM discerns evolving sentiments, trends, and shifts in sentiment expression within twitter conversations. This temporal modeling capability enables the model to understand how sentiment evolves and fluctuates across various topics and discussions on the platform.

4.3. Integration and fusion of features

The integrated CNN-LSTM model strategically combines the outputs of the CNN and LSTM components to capture both local nuances and global context in Tweet sentiment. Through feature fusion techniques, the model integrates the extracted local features with the temporal representations learned by the LSTM, thereby achieving a holistic understanding of sentiment dynamics on twitter.

4.4. Mean statistics

In the realm of sentiment evaluation on tweeted updates using a CNN-LSTM approach, the mean performance metrics acquired from analyzing the framework on a test set play a crucial role in understanding the system’s effectiveness in multiple aspects of sentiment mining.

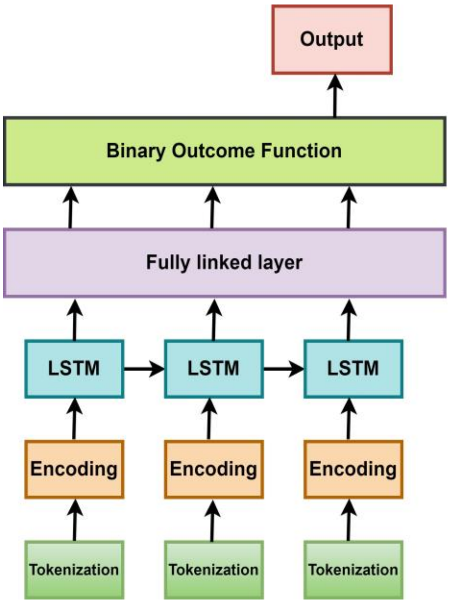

Fig. 9. LSTM network framework for sentiment mining

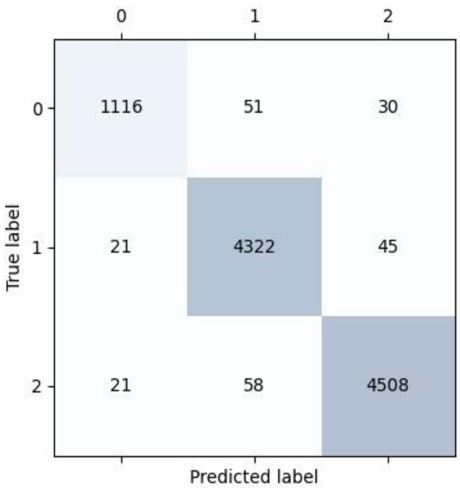

Fig. 10. Performance matrix of LSTM-CNN model

As a broad assessment of the algorithm's overall correctness, is calculated as the proportion of accurately predicted occurrences to the total amount of occurrences. Precision, on the other hand, measures the percentage of precisely anticipated favorable results to all expected positive aspects, indicating how well the model avoids inaccurate predictions. Recall, representing the proportion of precisely anticipated positive outcomes to the overall actual ones, evaluates the model's capacity to identify all favorable cases.

The F1 score, indicating the harmonic average of both recall and precision, offers a balanced assessment considering both false positives and false negatives. These metrics are essential for gauging the model's capability in sentiment analysis tasks, such as accurately classifying Tweets into positive, negative, or neutral sentiments.

The specific values of these metrics, as depicted in the performance matrix of an LSTM-CNN model (Fig 10), depend on factors such as training data, model architecture, and hyper parameters. A thorough analysis of these metrics is vital for comprehensively evaluating the model's performance in sentiment analysis.

4.5. Data processing, model training, and hyperparameters

To conduct sentiment analysis on Twitter data, several crucial preprocessing steps are undertaken to prepare the text for model training. Initially, the raw tweet text is cleaned to remove noise and irrelevant information. This includes removing special characters, punctuation, URLs, and user mentions. Additionally, common stop words are eliminated to reduce dimensionality and focus on meaningful content. Furthermore, tokenization is applied to break down the text into individual words or tokens. Stemming or lemmatization may also be performed to normalize variations of words to their root form, aiding in reducing sparsity and improving model generalization. Emoticons and emojis may also be replaced with textual representations to ensure they are properly interpreted during analysis. Finally, the text is encoded into numerical vectors using techniques like one-hot encoding or word embeddings, allowing it to be processed by machine learning models effectively.

Model training parameters play a crucial role in the performance of sentiment analysis models on Twitter data. When training deep learning architectures such as CNN and LSTM, parameters like learning rate, batch size, and number of epochs are carefully chosen to optimize model convergence and prevent overfitting. The learning rate determines the step size during gradient descent optimization, affecting the speed and stability of convergence. Batch size determines the number of samples processed before updating model parameters, balancing computational efficiency and gradient accuracy. Moreover, the number of epochs defines the number of times the entire dataset is passed through the model during training, influencing the model's ability to capture patterns in the data. These parameters are often tuned through experimentation and validation on separate validation datasets to ensure optimal model performance.

The selection of hyperparameters in sentiment analysis models for Twitter data involves a careful balance between model complexity and generalization ability. For example, in the combined CNN-LSTM architecture, hyperparameters such as the number of convolutional layers, filter sizes, and LSTM units are chosen to capture hierarchical features in the text data while avoiding overfitting. Additionally, dropout regularization may be applied to prevent co-adaptation of neurons and improve model robustness. Hyperparameters in other algorithms like SVM, Random Forest, and MLP are similarly selected based on their impact on model performance metrics such as accuracy, precision, recall, and F1score. Grid search or random search techniques may be employed to systematically explore the hyperparameter space and identify the optimal configuration. Ultimately, the selection of hyperparameters aims to strike a balance between model complexity and computational efficiency while achieving high performance in sentiment classification tasks on Twitter data.

4.6. Mean statistics

In the realm of sentiment evaluation on tweeted updates using a CNN-LSTM approach, the mean performance metrics acquired from analyzing the framework on a test set play a crucial role in understanding the system's effectiveness in multiple aspects of sentiment mining.

As a broad assessment of the algorithm's overall correctness, is calculated as the proportion of accurately predicted occurrences to the total amount of occurrences. Precision, on the other hand, measures the percentage of precisely anticipated favorable results to all expected positive aspects, indicating how well the model avoids inaccurate predictions. Recall, representing the proportion of precisely anticipated positive outcomes to the overall actual ones, evaluates the model's capacity to identify all favorable cases.

The F1 score, indicating the harmonic average of both recall and precision, offers a balanced assessment considering both false positives and false negatives. These metrics are essential for gauging the model's capability in sentiment analysis tasks, such as accurately classifying Tweets into positive, negative, or neutral sentiments.

The specific values of these metrics, as depicted in the performance matrix of an LSTM-CNN model (Fig 10), depend on factors such as training data, model architecture, and hyper parameters. A thorough analysis of these metrics is vital for comprehensively evaluating the model's performance in sentiment analysis.

4.7. Data processing, model training, and hyperparameters

To conduct sentiment analysis on Twitter data, several crucial preprocessing steps are undertaken to prepare the text for model training. Initially, the raw tweet text is cleaned to remove noise and irrelevant information. This includes removing special characters, punctuation, URLs, and user mentions. Additionally, common stopwords are eliminated to reduce dimensionality and focus on meaningful content. Furthermore, tokenization is applied to break down the text into individual words or tokens. Stemming or lemmatization may also be performed to normalize variations of words to their root form, aiding in reducing sparsity and improving model generalization. Emoticons and emojis may also be replaced with textual representations to ensure they are properly interpreted during analysis. Finally, the text is encoded into numerical vectors using techniques like one-hot encoding or word embeddings, allowing it to be processed by machine learning models effectively.

Model training parameters play a crucial role in the performance of sentiment analysis models on Twitter data. When training deep learning architectures such as CNN and LSTM, parameters like learning rate, batch size, and number of epochs are carefully chosen to optimize model convergence and prevent overfitting. The learning rate determines the step size during gradient descent optimization, affecting the speed and stability of convergence. Batch size determines the number of samples processed before updating model parameters, balancing computational efficiency and gradient accuracy. Moreover, the number of epochs defines the number of times the entire dataset is passed through the model during training, influencing the model's ability to capture patterns in the data. These parameters are often tuned through experimentation and validation on separate validation datasets to ensure optimal model performance.

The selection of hyperparameters in sentiment analysis models for Twitter data involves a careful balance between model complexity and generalization ability. For example, in the combined CNN-LSTM architecture, hyperparameters such as the number of convolutional layers, filter sizes, and LSTM units are chosen to capture hierarchical features in the text data while avoiding overfitting. Additionally, dropout regularization may be applied to prevent co-adaptation of neurons and improve model robustness. Hyperparameters in other algorithms like SVM, Random Forest, and MLP are similarly selected based on their impact on model performance metrics such as accuracy, precision, recall, and F1score. Grid search or random search techniques may be employed to systematically explore the hyperparameter space and identify the optimal configuration. Ultimately, the selection of hyperparameters aims to strike a balance between model complexity and computational efficiency while achieving high performance in sentiment classification tasks on Twitter data.

5. Experimental results and discussions

In this section, we outline the experimental design and present the results of our sentiment analysis model. The provided Algorithm 7 represented the training and evaluation steps, employing a neural network model combining LSTM and CNN layers.

5.1. Experimental design

The provided Algorithm 7, outlines the training and evaluation steps for a neural network model designed for sentiment analysis using a combination of LSTM and CNN layers. The process is divided into three main procedures: TrainModel, EvaluateModel, and Main. In the Train Model procedure, the algorithm initializes a neural network model with a sequential architecture. The model incorporates layers for word embedding, one-dimensional convolution with Rectified Linear Unit (ReLU) activation, and max-pooling to capture local patterns in the input data.

Algorithm 7. Hybrid model with CNN-LSTM method

Input: x_{train}, y_{train}, x_{val}, y_{val}, x_{test}
Output: $y_{\text{pred\_test\_mlp}}$
np.random.seed(seed)
K.clear session()
model ← Sequential()
model.add(MaxPooling1D(pool size=2))
max features ← 20000
embed dim ← 100
TrainModel(max_eatures, embed_im)
EvaluateModel()

Additionally, LSTM layer is employed to capture long-range dependencies in the sequence data. The model is concluded with a dense layer using the softmax activation function for sentiment classification. The algorithm compiles the model using categorical cross-entropy loss and the adam optimizer, and then trains the model on the provided training data (xtrain and ytrain) with a specified number of epochs and batch size.

The EvaluateModel procedure assesses the trained model's performance on a separate test dataset (xtest and ytest). It computes predictions for the test data, calculates accuracy, and generates a classification report along with a confusion matrix to provide insights into the model's classification performance.

The Main procedure serves as the entry point, where hyperparameters such as maxfeatures and embeddim are set. It then calls the Train Model and EvaluateModel procedures, executing the complete pipeline for sentiment analysis using LSTM and CNN layers.

The Algorithm 7 processes with three steps: training (x train, y train), validation (x val, y val), and test (x test) sets. The Embedding layer converts integer-encoded sequences (x train) into fixed-size dense vectors, capturing contextual word representations. A subsequent 1D CNN layer extracts local features via convolutional operations and reduces spatial dimensions through MaxPooling1D. The output undergoes LSTM layer processing, capturing long-range dependencies in sequential data. The final layer is a dense layer with softmax activation, producing probability distributions for classification. This architecture is effective for tasks like sentiment analysis or topic classification, with the dense layer units aligning with the output classes.

5.2. Key findings

The high precision and recall values indicate that the integrated CNN and LSTM architecture effectively captures the nuanced patterns in twitter data, distinguishing between different sentiments. The model's ability to maintain high accuracy across sentiment classes suggests its robustness in handling diverse language and expressions found in twitter texts.

The elevated values suggest that the model excels in correctly identifying and distinguishing between different sentiments, showcasing its capability to precisely categorize Tweets into positive, negative, or neutral sentiments.

The integration of CNN and LSTM networks is particularly advantageous. The CNN layers excel in capturing local patterns, such as specific phrases or keywords indicative of sentiment, while LSTM layers contribute to understanding the sequential dependencies and long-term context in Tweets. This combination addresses the unique challenges posed by short, contextually rich twitter texts. The provided data suggests that the CNN and LSTM-based twitter sentiment analysis model exhibits strong learning capabilities, achieving impressive accuracy and low loss values on both training and validation datasets.

These results indicate that the model is likely to perform well in predicting sentiments in real-world twitter data, with the potential for effective integration into sentiment analysis applications.

5.3. Proposed model analysis

The presented sentiment analysis model distributed Tweets for positive, negative as well as neutral sentiments integrating CNN and LSTM networks, exhibited highly favorable results across multiple metrics. The model was evaluated on a dataset comprising three sentiment classes (0, 1, and 2).

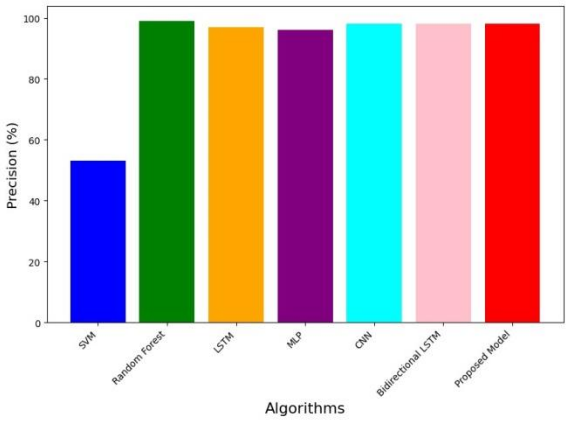

Fig. 11. Precision analysis of discussed algorithms

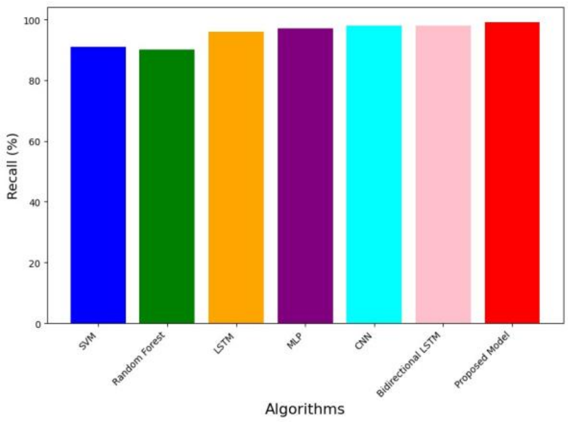

Fig. 12. Recall analysis of discussed algorithms

The performance metrics-Precision, Recall, F1 score, and Accuracy, showcasing their comparative effectiveness in sentiment analysis (Fig 11, 12, 13, and 14), applied to twitter data using various ML techniques and proposed model.

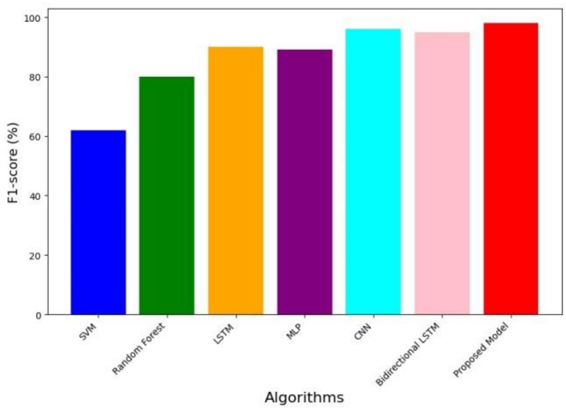

Fig. 13. F1 score analysis of discussed algorithms

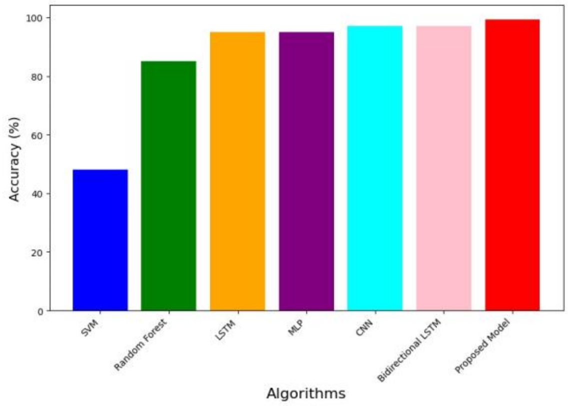

Fig. 14. Accuracy analysis of discussed algorithms

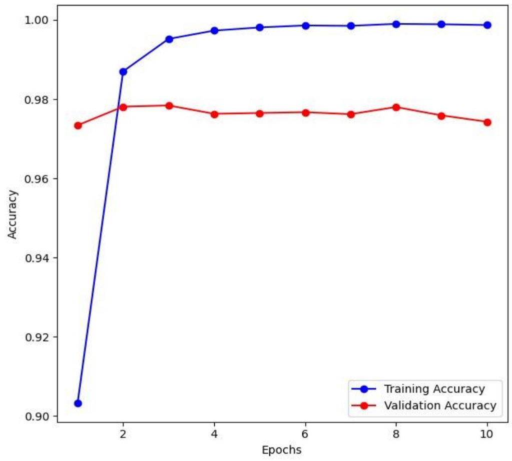

Fig. 15. Training and validation accuracy of model

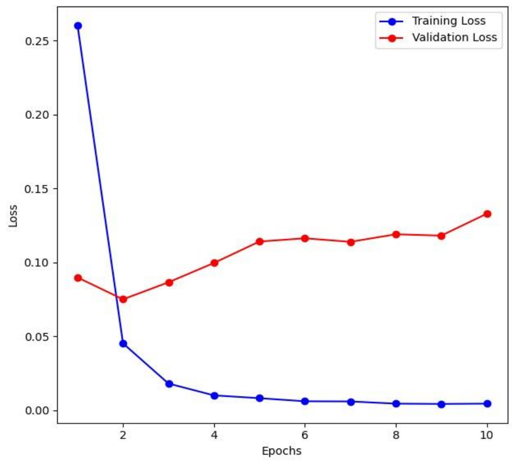

Fig. 16. Training and validation loss of model

The precision values for each sentiment class demonstrated the model's ability to correctly identify instances of a given sentiment, with class 1 and 2 achieving particularly high precision scores of 98%. This suggests that the model has a strong capacity to avoid false positives, crucial for applications where misclassifying sentiment can have significant implications. Recall scores, representing the model's capability to capture all instances of a particular sentiment, were consistently high across all classes. Notably, class 2 achieved an impressive recall of 99%, indicating a robust ability to identify instances of sentiment class 2. The balanced precision and recall values signify the model's effectiveness in handling each sentiment class without bias.

The Table 2, presents the performance metrics, including Precision, Recall, F1-score, Support, and Accuracy, for various sentiment analysis algorithms, namely SVM, Random forest, LSTM, MLP, CNN, Bidirectional LSTM, and the proposed model. Each algorithm is evaluated across three sentiment classes (0, 1, and 2).

The Bidirectional LSTM and the proposed model stand out with exceptional result, reaching up to 98%. These models exhibit a heightened ability to capture sentiment nuances across all classes, demonstrating their effectiveness in sentiment analysis. Notably, the proposed model aligns closely with the Bidirectional LSTM, emphasizing its competitiveness in comparison to established DL models.

Additionally, the CNN-LSTM model, while proficient at capturing local and temporal dependencies within sequential data like text, faces several limitations in the context of Twitter sentiment analysis. These limitations include struggles with nuanced contextual understanding, particularly in interpreting subtle or sarcastic sentiments, as well as challenges posed by the sparse and noisy nature of Twitter data, which may hinder the model’s generalization ability. Furthermore, the lack of interpretability inherent in deep learning models poses challenges in understanding the rationale behind the model’s predictions, potentially leading to issues of trust and ethical use. To address these challenges, future work could focus on mitigating dataset biases through the collection of more diverse data and implementing fair labeling practices. Additionally, efforts to enhance model performance could involve incorporating external knowledge sources, employing ensemble approaches, and leveraging techniques like explainable AI (XAI) and transfer learning to improve interpretability and capture richer contextual information. By addressing these challenges, researchers can develop more robust sentiment analysis techniques for Twitter data, enabling a better understanding of public opinions and sentiments in social media.

Table 2. Classification report for models

Algorithm	Class	Precision (%)	Recall (%)	F1-score (%)	Support	Accuracy
SVM	0	00	00	00	1197	48
	1	53	15	24	4388	
	2	47	91	62	4587	
Random forest	0	99	68	80	1197	85
	1	81	90	86	4388	
	2	87	85	86	4587	
LSTM	0	90	90	90	1197	95
	1	96	96	96	4388	
	2	97	96	97	4587	
MLP	0	95	85	89	1197	95
	1	94	97	96	4388	
	2	96	96	96	4587	
CNN	0	97	92	95	1197	97
	1	98	98	98	4388	
	2	97	99	98	4587	
Bidirectional LSTM	0	97	93	95	1197	98
	1	97	99	98	4388	
	2	98	98	98	4587	
Proposed Model	0	96	93	95	1197	98
	1	98	98	98	4388	
	2	98	98	98	4587	

During the training process, the model’s loss steadily decreases shown in Fig 15, reflecting the diminishing difference between predicted sentiment and actual sentiment in the training data. The initial loss of 0.2604 reduces substantially with each epoch, reaching an impressively low value of 0.0044 by the 10th epoch. This indicates that the model is effectively learning to make accurate sentiment predictions on the training data. Starting at an accuracy of 90.33%, the model achieves an outstanding accuracy of 99.87% by the end of the training process. Representing the validation result of model’s shown in Fig 16, the performance on unseen data, follow a similar trend. The validation loss decreases from 0.0898 to 0.1329, showcasing the model’s ability to generalize well to new twitter data. Validation accuracy remains consistently high, ranging from 97.34% to 97.43%, indicating that the model maintains a robust performance on data it has not been explicitly trained on.

6. Conclusion

In analyzing sentiments on twitter through ML techniques, a comprehensive evaluation of various algorithms was conducted, and accuracy metrics were employed across three sentiment classes (0, 1, and 2). The results indicate distinct performances across the models. SVM, while showing a moderate accuracy of 48%, exhibited challenges in recall and F1-score for sentiment class 1. Random forest, on the other hand, demonstrated a notable improvement with an overall accuracy of 85%, showcasing balanced precision, recall, and F1-scores across sentiment classes. LSTM and MLP outperformed SVM, achieving accuracy rates of 95%, displaying a high score across all sentiment classes. The CNN displayed remarkable efficiency with an accuracy of 97%, excelling in sentiment classification for all classes. Bidirectional LSTM showcased an even higher accuracy of 98%, indicating its proficiency in capturing contextual nuances. The proposed model exhibited competitive results with an accuracy of 98%, aligning closely with the performance of Bidirectional LSTM and CNN. In conclusion, the study highlights the significance of choosing an appropriate ML algorithm for sentiment analysis on twitter. While traditional methods like SVM may exhibit limitations, advanced techniques such as CNN, Bidirectional LSTM, and the proposed model showcase substantial improvements in accuracy and overall performance. The findings underscore the potential of leveraging advanced ML techniques for nuanced sentiment analysis, providing valuable insights into public opinions on twitter.

References

[1] Harleen Kaur, Shafqat Ul Ahsaan, Bhavya Alankar, and Victor Chang. A proposed sentiment analysis deep learning algorithm for analyzing covid-19 tweets. *Information Systems Frontiers*, pages 1–13, 2021.

[2] Sepideh Kaffash, An Truong Nguyen, and Joe Zhu. Big data algorithms and applications in intelligent transportation system: A review and bibliometric analysis. *International Journal of Production Economics*, 231:107868, 2021.

[3] Prabhsimran Singh, Surleen Kaur, Abdullah M Baabdullah, Yogesh K Dwivedi, Sandeep Sharma, Ravinder Singh Sawhney, and Ronnie Das. Is# sdg13 trending online? insights from climate change discussions on twitter. *Information Systems Frontiers*, 25(1):199–219, 2023.

[4] Thanveer Shaik, Xiaohui Tao, Christopher Dann, Haoran Xie, Yan Li, and Linda Galligan. Sentiment analysis and opinion mining on educational data: A survey. *Natural Language Processing Journal*, 2:100003, 2023.

[5] Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, and Rada Mihalcea. Beneath the tip of the iceberg: Current challenges and new directions in sentiment analysis research. *IEEE Transactions on Affective Computing*, 2020.

[6] Patricia LV Ribeiro, Li Weigang, and Tiancheng Li. A unified approach for domain-specific tweet sentiment analysis. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 846–853. IEEE, 2015.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[8] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. Ieee, 2013.

[9] Raj P Mehta, Meet A Sanghvi, Darshin K Shah, and Artika Singh. Sentiment analysis of tweets using supervised learning algorithms. In *First International Conference on Sustainable Technologies for Computational Intelligence: Proceedings of ICTSCI 2019*, pages 323–338. Springer, 2020.

[10] Ishaani Priyadarshini and Chase Cotton. A novel lstm–cnn–grid search-based deep neural network for sentiment analysis. *The Journal of Supercomputing*, 77(12):13911–13932, 2021.

[11] Mia Gerber and Nelishia Pillay. Automated design of the deep neural network pipeline. *Applied Sciences*, 12(23):12215, 2022.

[12] Milan Tripathi. Sentiment analysis of nepali covid19 tweets using nb svm and lstm. *Journal of Artificial Intelligence*, 3(03):151–168, 2021.

[13] Imane Lasri, Anouar Riadsolh, and Mourad Elbelkacemi. 'self-attention-based bi-lstm model for sentiment analysis on tweets about distance learning in higher education. *Int. J. Emerg. Technol. Learn*, 18(12):119–141, 2023.

[14] Shanta H Biradar, JV Gorabal, and Gaurav Gupta. Machine learning tool for exploring sentiment analysis on twitter data. *Materials Today: Proceedings*, 56:1927–1934, 2022.

[15] Marouane Birjali, Mohammed Kasri, and Abderrahim Beni-Hssane. A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowledge-Based Systems*, 226:107134, 2021.

[16] Suraksha Gupta, David Gallear, John Rudd, and Pantea Foroudi. The impact of brand value on brand competitiveness. *Journal of Business Research*, 112:210–222, 2020.

[17] Abdullah Alsaeedi and Mohammad Zubair Khan. A study on sentiment analysis techniques of twitter data. *International Journal of Advanced Computer Science and Applications*, 10(2):361–374, 2019.

[18] Yin Zhang, Jinyang Du, Xiao Ma, Haoyu Wen, and Giancarlo Fortino. Aspect-based sentiment analysis for user reviews. *Cognitive Computation*, 13(5):1114–1127, 2021.

[19] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[20] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015.

[21] Usha Devi Gandhi, Priyan Malarvizhi Kumar, Gokulnath Chandra Babu, and Gayathri Karthick. Sentiment analysis on twitter data by using convolutional neural network (cnn) and long short term memory (lstm). *Wireless Personal Communications*, pages 1–10, 2021.

[22] Jin Zheng and Limin Zheng. A hybrid bidirectional recurrent convolutional neural network attention-based model for text classification. *IEEE Access*, 7:106673–106685, 2019.

[23] Faliang Huang, Xuelong Li, Changan Yuan, Shichao Zhang, Jilian Zhang, and Shaojie Qiao. Attention-emotionenhanced convolutional lstm for sentiment analysis. *IEEE transactions on neural networks and learning systems*, 33(9):4332–4345, 2021.

[24] Mehmet Umut Salur and Ilhan Aydin. A novel hybrid deep learning model for sentiment classification. *IEEE Access*, 8:58080–58093, 2020.

[25] Manoochehr Ghiassi, James Skinner, and David Zimbra. Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. *Expert Systems with applications*, 40(16):6266–6282, 2013.

[26] Akshi Kumar and Ritu Rani. Sentiment analysis using neural network. In *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, pages 262–267. IEEE, 2016.

[27] Amira Samy Talaat. Sentiment analysis classification system using hybrid bert models. *Journal of Big Data*, 10(1):1–18, 2023.

[28] Gagandeep Kaur and Amit Sharma. A deep learning-based model using hybrid feature extraction approach for consumer sentiment analysis. *Journal of Big Data*, 10(1):5, 2023.

[29] Peerapon Vateekul and Thanabhat Koomsubha. A study of sentiment analysis using deep learning techniques on thai twitter data. In *2016 13th International joint conference on computer science and software engineering (JCSSE)*, pages 1–6. IEEE, 2016.

[30] Arnab Roy and Muneendra Ojha. Twitter sentiment analysis using deep learning models. In *2020 IEEE 17th India Council International Conference (INDICON)*, pages 1–6. IEEE, 2020.

[31] Arepalli Peda Gopi, R Naga Sravana Jyothi, V Lakshman Narayana, and K Satya Sandeep. Classification of tweets data based on polarity using improved rbf kernel of svm. *International Journal of Information Technology*, 15(2):965–980, 2023.
[32] P Karthika, R Murugeswari, and R Manoranjithem. Sentiment analysis of social media network using random forest algorithm. In *2019 IEEE international conference on intelligent techniques in control, optimization and signal processing (INCOS)*, pages 1–5. IEEE, 2019.
[33] Satyabrata Aich, Sabyasachi Chakraborty, and Hee-Cheol Kim. Convolutional neural network-based model for web-based text classification. *International Journal of Electrical & Computer Engineering (2088-8708)*, 9(6), 2019.
[34] Anushree Goud and Bindu Garg. Sentiment analysis using long short-term memory model in deep learning. In *2nd EAI International Conference on Big Data Innovation for Sustainable Cognitive Computing: BDCC 2019*, pages 15–23. Springer, 2021.
[35] Deepak Chowdary Edara, Lakshmi Prasanna Vanukuri, Venkatramaphanikumar Sistla, and Venkata Krishna Kishore Kolli. Sentiment analysis and text categorization of cancer medical records with lstm. *Journal of Ambient Intelligence and Humanized Computing*, 14(5):5309–5325, 2023.
[36] Guixian Xu, Yueting Meng, Xiaoyu Qiu, Ziheng Yu, and Xu Wu. Sentiment analysis of comment texts based on bilstm. *Ieee Access*, 7:51522–51532, 2019.
[37] Hao Sun, Hongyi Wang, Jiaqing Liu, Yen-Wei Chen, and Lanfen Lin. Cubemlp: An mlp-based model for multimodal sentiment analysis and depression estimation. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 3722–3729, 2022.
[38] Shafiul Hasan Rafi, Shohana Rahman Deeba, Eklas Hossain, et al. A short-term load forecasting method using integrated cnn and lstm network. *IEEE Access*, 9:32436–32448, 2021.
[39] Qingfeng Lan, Yangchen Pan, Alona Fyshe, and Martha White. Maxmin q-learning: Controlling the estimation bias of q-learning. *arXiv preprint arXiv:2002.06487*, 2020.

Authors' Profiles

Aditya Bhushan is a M.Tech student in Department of Computer Science and Engineering at United College of Engineering & Research, Prayagraj, India and received his post-graduation degree in MCA from University of Allahabad. His current research interests include Machine Learning, Cyber-Security, IoT, Privacy and Data Analytics. He is having membership of IEEE.

Devanshi Dwivedi is a M.Tech student in Department of Computer Science and Engineering at United College of Engineering \& Research, Prayagraj, India and completed B.Tech in CSE from Dr. A.P.J. Abdul Kalam Technical University (AKTU) Lucknow, India. Her current research interests include Network Security, IoT, and Machine Learning. She is having membership of IEEE.

Ashutosh Kumar Singh has done his Bachelors of Technology (Information Technology) from U.P. Technical University, Lucknow in 2011. He has done his Masters of Technology (Computer Science and Engineering) from Indian Institute of Information Technology and Management Gwalior, India in 2014. He has done Ph.D. in the Department of Computer Science and Engineering, MNNIT Allahabad, India. Currently, he is working as an assistant professor in the Department of Computer Science and Engineering at United college of Engineering & Research Prayagraj, India. He is having membership of IEEE, ACM and CSI. His research interest includes network optimization, information security, IoT, and software defined networking. He is guided more than 3 M.Tech thesis and currently guiding 2 M.Tech thesis. He has published various research papers in reputed journals and conferences.

Snehlata has done her Bachelors of Technology (Computer Science and Engineering) from U.P. Technical University, Lucknow in 2011. She has done M.S. in Cyber Law and Information Security from IIIT-Allahabad, India in 2014. She has done Ph.D. in the Department of Computer Science and Engineering at BBAU (Central University) Lucknow in 2021. Her research interests include image recognition, information security, and cyber forensics. She is having membership of IEEE and CSI. She has published more than 10 research papers in reputed journals and conferences.

How to cite this paper: Aditya Bhushan, Devanshi Dwivedi, Ashutosh Kumar Singh, Snehlata, "Analyzing Sentiments on Twitter Using Deep Learning Techniques", International Journal of Modern Education and Computer Science(IJMECS), Vol.16, No.6, pp. 20-39, 2024. DOI:10.5815/ijmecs.2024.06.02