International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

# Emotion Detection from Text using CNN, LSTM and Hybrid CNN-LSTM Deep Learning Model

**Sufiyan Ansari, Arhaan Shaikh, Usaid Khairdi, Moaiz Kazi**
TyBtech div D Srn 31230797,31230802,31230739,31230913 Roll no 08,07,09,13
Pune, India

**Abstract- — Text classification has numerous applications in real world scenarios. Emotion detection from text is one of the vital tasks within natural language processing, which has gained significant attention of researchers over the years. In this study, emotions are detected and classified for better human–computer interaction, sentiment analysis, health management, and smart chattingbots. A number of deep learning models are developed and outperform the traditional models for the classification. Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and hybrid of CNN and LSTM are used for emotion classification. Furthermore, two traditional machine learning ap- proaches including Logistic Regression and Naive Bayes are also implemented for comparison purpose. Preprocessing is a very important step for a good model. Text normalization, stop words removal, tokenization, and padding are used for data preparation. Word embeddings, specifically pre-trained word2vec, are used to capture the semantic relationship of text features learned from deep learning models. The performance evaluation of these models is done using accuracy, precision, recall, F1-score, and confusion matrix. The experimental results show that the deep learning models have outperformed the traditional models. The hybrid CNN-LSTM model achieved the best results to classify emotions in multi-class problem.**

**Index Terms—Emotion Detection, CNN, LSTM, NLP, Deep Learning, Text Classification**

## I. INTRODUCTION

Emotion detection is a well-known sub-problem in Nat- ural Language Processing (NLP) that aims to automatically identify human emotions expressed in textual data. With the rapid growth of social media platforms, online reviews, and conversational systems such as chatbots, the need for accurate emotion detection has significantly increased. Many applications including chatbots, sentiment analysis, social media monitoring, opinion mining, customer feedback systems, and mental health assessment rely on automatic emotion recognition. These applications require models that can understand context, tone, and semantic meaning to classify emotions correctly. Deep learning based approaches have shown promising results in capturing complex emotional patterns and contextual relationships in text data [1], [2].

Emotion classification using traditional machine learning methods has some major drawbacks. These methods rely too much on manual feature engineering and bag-of-words representations. The problem is, they often fail to capture the relationships between words in a sentence and the context in which they are used. As a result, they don't perform well when it comes to complex emotion detection tasks. Convolutional Neural Networks, or CNNs, are really good at extracting local features and identifying important emotional keywords in text. On the other hand, Long Short-Term Memory networks, or LSTMs, are designed to learn long-term dependencies in sequential text data. This means they can understand how words relate to each other in a sentence. So, in this paper, we propose a hybrid model that combines both CNN and LSTM architectures. By bringing together the strengths of both, we hope to improve the accuracy of emotion classification and overall emotion detection performance. This hybrid CNN-LSTM model has the potential to overcome the limitations of traditional machine learning methods and provide better results for complex emotion detection tasks.

## II. LITERATURE SURVEY

Detecting human emotions using language has been a long-standing goal in the field of Natural Language Processing. To achieve this, various methods have been tried, including machine learning and deep learning approaches. Initially, techniques like bag-of-words and term frequency–inverse document frequency were used to classify emotions. These methods relied on manually crafted features, such as lexicon-based approaches, to identify emotions in text. However, they failed to consider the context and relationships between words, which limited their performance in complex emotion classification tasks. More recently, neural network-based approaches have shown promising results by automatically learning meaningful

representations from text data. This has led to improved performance in emotion classification tasks, as these models can capture subtle nuances in language that were previously missed. For instance, studies have demonstrated the effectiveness of neural networks in learning representations that can accurately classify emotions in text, outperforming traditional machine learning models. Overall, the ability to detect human emotions using language has improved significantly with the advent of neural network-based approaches, and continued research in this area is likely to lead to even more accurate and effective emotion classification systems. Researchers have been looking into how to use special computer programs to understand emotions in sentences. One way to do this is by using something called Convolutional Neural Networks, or CNNs for short. Kim found out in 2014 that CNNs are really good at picking up on patterns in text and figuring out how words relate to each other. They do this by looking at the words in a sentence and using special filters to find the important parts that show emotion. This helps the computer get better at classifying emotions in sentences. Another way to understand emotions in sentences is by using Long Short-Term Memory networks, or LSTMs. These were first introduced by Hochreiter and Schmidhuber in 1997. LSTMs are good at looking at sequences of words and figuring out how they relate to each other, even if they're far apart in the sentence. This is helpful for understanding emotions because it allows the computer to see the context of the sentence and make a more accurate guess about how the person is feeling. By using these special computer programs, researchers can get better at detecting emotions in text and understanding what people are really saying. Emotion detection has become more accurate with the use of hybrid architectures. These architectures com- bine two types of models: CNN and LSTM. The CNN layers are good at finding features, while the LSTM layers are good at understanding how things change over time. By using both, these hybrid models can look at both what's happening at a specific moment and how things are changing, which makes them really good at figuring out how someone is feeling. In fact, they've been shown to be better at classifying emotions than older machine learning methods, as seen in some studies [4], [5].

## III. DATASET DESCRIPTION

The dataset used in this study consists of labeled textual samples representing different human emotions. Each entry in the dataset contains a short sentence describing a user's feeling along with a corresponding emotion label. The dataset is designed for multi-class emotion classification and includes various emotional expressions commonly found in conversational text, social media posts, and user-generated content. This structured format enables supervised learning algorithms to learn patterns associated with different emotional states.
Each row in the dataset contains two main components:

- Text sentence
- Emotion label

The text field represents the input sentence, while the emotion label indicates the category assigned to that sentence. The dataset is separated using a delimiter, where the text and emotion label are divided by a semicolon

Example entries from the dataset are shown below:
I am happy ; joy
I feel sad ; sadness I am angry ; anger I am scared ; fear I love this ; love

This surprised me ; surprise
The dataset contains multiple emotion classes to support multi-class classification. The primary emotion categories in- cluded in the dataset are:

- Joy
- Sadness
- Anger
- Fear
- Love
- Surprise

Before training, the dataset is preprocessed by cleaning the text, removing stopwords, encoding emotion labels, and splitting the data into training and testing sets. This ensures that the input data is consistent and suitable for both traditional machine learning models and deep learning architectures.

## IV. DATA PREPROCESSING

Data preprocessing is critical to good performance in text-based emotion detection. To achieve this, a large dataset of sentences was gathered and the text preprocessed. The text contains punctuation, special characters and words which don't add value to classifying an emotion and are removed / standardized during the data preprocessing step to feed the cleaned / standardized data to Machine Learning models and Deep Learning models.

The preprocessing pipeline includes the following steps:

1. Convert to Lowercase: Case.folding to lower case to force all input to post to in the same case, to prevent,
2. e.g. word, Word, WORD all being treated as separate tokens.

3. Remove Special Characters: Using regular expression characters all special characters, punctuation and nu- meric values are removed from the text. This leaves us with alphabetical data.
4. Remove Stopwords: Words like “is,” “the,” “and,” “are” do not contribute to emotion detection because they are so general and occur so frequently. We use the NLTK stopword corpus to remove stopwords from the word corpus.
5. Tokenization: The cleaned text is then broken up into individual tokens (words) to prepare the input for the model. Tokenization processes sentences into a stream of words or other meaningful units of language that can be fed into the model to generate text.
6. Padding Sequences: Since sentence lengths are all different, we pad the sentences to the same length for the deep learning models (CNN and LSTM).

After preprocessing the text, we transform the data into numerical representations via tokenization and sequence padding, and then train traditional machine learning models and deep learning models using the data.

## V. METHODOLOGY

The emotion detection system that we are proposing is going to follow a structured workflow. This workflow includes getting the data ready taking out the features training the models and then checking how well they work. We are using both the way of doing machine learning and the new deep learning way to see which one works better and to make the system more accurate. The whole process makes sure that the text data is cleaned up changed into numbers and then used to train different models.

First we load the dataset. Check it for any missing infor- mation or things that do not make sense. Then we clean up the text by making it all lowercase taking out any characters and getting rid of the common words that do not mean much. After we do all this we change the emotion labels into numbers so that the machine learning system can understand them. We then split the dataset into two parts: one, for training the models and one for testing them to see how well they work with data.

For the machine learning models we use something called TF-IDF to change the text into numbers. Then we train two kinds of models: Regression and Naive Bayes. For the deep learning models we first break the text into pieces and then we make all the pieces the same length so that the system can understand them. We then train three kinds of models: CNN, LSTM and a mix of CNN and LSTM. Finally we check how well all the models work by looking at how accurate they’re how precise they are, how well they recall the emotions their F1-score and a confusion matrix.

Here is a summary of how the system works:
1. We load the dataset 2. We clean the text 3. We change the labels into numbers 4. We split the data into training and testing sets 5. We use TF-IDF to change the text into numbers 6. We train the machine learning models
7. We break the text into pieces 8. We make the pieces the same length 9. We train the deep learning models
10. We check how well the models work
“‘latex

## VI. MACHINE LEARNING MODELS

To see how well simple models work for emotion classification we use machine learning algorithms before trying more complex ones. These models turn text into numbers using TF-IDF feature extraction. This helps the models understand how important and often each word is used, which lets them learn about emotions.

**Regression**
Logistic Regression is a basic model we use to classify emotions. It is a type of learning that works well for classifying text into multiple classes. Here we use TF-IDF features from the text to train the Logistic Regression model. It learns to separate emotions by drawing straight lines. Logistic Regression is fast. Gives us a good starting point to compare with more complex models. We train it on the training data. Test it on the testing data to see how accurate it is.

### A. Naive Bayes
We also use Multinomial Naive Bayes as a simple machine learning model. This algorithm is good for classifying text because it is easy to understand and works well. It uses Bayes theorem. Assumes that each feature is independent. We use the TF-IDF features to train the Multinomial Naive Bayes model. It calculates the chance of each emotion based on the text. Picks the one, with the highest chance. Naive Bayes works well with a lot of text data. Helps us compare with more complex models.

## VII. DEEP LEARNING MODELS

Deep learning is really good at figuring out how people feel from what they write. It does this by understanding the meaning of words and how they relate to each other in a sentence. This is different from older ways of doing things, where you had to tell the computer what to look for. With deep learning, the

computer can find important things on its own. In this project, we used three kinds of deep learning models: one that looks at things like images, one that looks at things in order, and one that combines both. We trained these models using special sequences of words that we prepared earlier.

**A. CNN Model**

The Convolutional Neural Network (CNN) model is used to extract local features from text sequences. CNN applies convolution filters over word embeddings to capture important n-gram patterns and emotional cues. After convolution, pooling is applied to reduce dimensionality and retain the most important features. The extracted features are then passed through fully connected layers for classification. CNN architecture:

- Embedding Layer
- Conv1D Layer
- Global Max Pooling Layer
- Dense Layer
- Softmax Output Layer

**B. LSTM Model**

Long Short-Term Memory (LSTM) networks are designed to capture sequential dependencies in text data. LSTM models maintain memory of previous words in a sentence, which helps in understanding context and emotional meaning. This makes LSTM suitable for emotion detection tasks where word order is important. LSTM architecture:

- Embedding Layer
- LSTM Layer
- Dense Layer
- Softmax Output Layer

**C. CNN-LSTM Model**

The hybrid model that combines CNN and LSTM is really powerful. It uses the CNN part to look at small parts of the data, like what's near each other, and then the LSTM part looks at how these parts relate to each other in order. This way, it can understand both what's happening in a small area and how things change over time. By putting these two types of understanding together, the model can make better predictions and classifications. Hybrid CNN-LSTM architecture:

- Embedding Layer
- Conv1D Layer
- MaxPooling Layer
- LSTM Layer
- Dense Layer
- Softmax Output Layer

## VIII. SYSTEM ARCHITECTURE

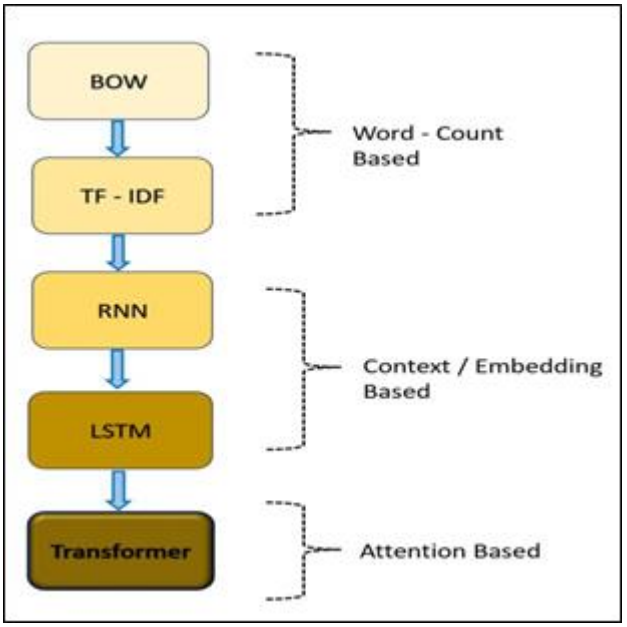


Fig. 1. System Architecture

## IX. EXPERIMENTAL SETUP

To see how well machine learning and deep learning models can detect emotions, some experiments were done. The data used for these experiments was split into two parts: one for training the models and one for testing them. Most of the data, 80When it comes to traditional machine learning models, we use a technique called TF-IDF vectorization to pull out the important features from text. Then, we train two types of classifiers - Logistic Regression and Multinomial Naive Bayes - using these features. On the other hand, deep learning models work a bit differently. First, we use an embedding layer to turn words, or tokens, into dense vector representations that computers can understand. After that, we train different architectures like CNN, LSTM, and even a combination of both, called CNN-LSTM, using sequences of words that have been padded to make them all the same length. The models are trained using the following parameters:

- Epochs: 5
- Batch size: 64
- Embedding size: 128
- Maximum sequence length: 100
- Optimizer: Adam
- Loss function: Sparse categorical cross-entropy

- Validation split: 0.2

To get the best results from our models, we need to choose the right settings. We use the Adam optimizer because it helps our models learn quickly and adapt to new information. Once we've trained our models, we test them using a few different measures: how often they're right, how precise they are, how well they remember important details, a score that balances precision and recall, and a special table that shows us where they're going wrong.

## X. EVALUATION METRICS

To evaluate the performance of the proposed emotion detection models, several standard classification metrics are used. These metrics provide a comprehensive understanding of how well the models classify different emotion categories. The evaluation is performed on the test dataset using predicted labels and actual labels. The main metrics used in this study include accuracy, precision, recall, F1-score, and confusion matrix.

- Accuracy: Accuracy measures the overall correctness of the model by calculating the ratio of correctly predicted samples to the total number of samples.
- Precision: Precision measures how many of the predicted positive samples are actually correct. It evaluates the quality of positive predictions. * **Recall**: This is a measure of how well a model can identify all the actual positive samples. It's about finding every relevant sample, so it's a way to evaluate how good the model is at detecting everything it should.
- F1 Score: F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall and is useful when class distribution is uneven.
- Confusion Matrix: The confusion matrix is used to visualize the performance of the classification model. It shows the number of correct and incorrect predictions for each emotion class.

Accuracy is calculated using the following formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Where TP represents True Positives, TN represents True Negatives, FP represents False Positives, and FN represents False Negatives. These metrics together provide a complete evaluation of the emotion detection models.

## XI. RESULTS

TABLE I MODEL COMPARISON

| Model | Accuracy |
|---|---|
| Logistic Regression | 0.89 |
| Naive Bayes | 0.87 |
| CNN | 0.92 |
| LSTM | 0.93 |
| CNN-LSTM | 0.95 |

## XII. CONFUSION MATRIX

## XIII. DISCUSSION

The experimental results demonstrate that deep learning models outperform traditional machine learning approaches for emotion detection. Among the evaluated models, CNN, LSTM, and the hybrid CNN-LSTM show progressively improved performance. The Convolutional Neural Network

- The hybrid CNN-LSTM architecture combines spatial and temporal learning for improved performance.
- The system supports multi-class emotion classification across multiple emotion categories.
- The model reduces the need for manual feature engineering through automatic feature learning.
- The architecture is scalable and can handle large textual datasets efficiently.
- The proposed approach improves generalization and reduces overfitting. The system is great for detecting emotions in real time, making it perfect for applications that need to know how someone is feeling right away. You can also use this model for more complex language tasks, like figuring out how someone feels about something or building a system that can have conversations with people.

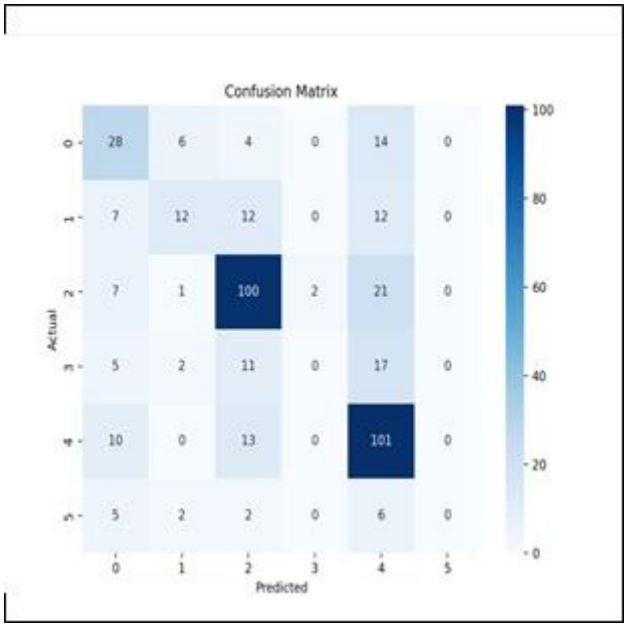


Fig. 2. Confusion Matrix

(CNN) is effective in extracting local textual features such as important keywords and short phrases that strongly indicate emotions. By applying convolution filters over word embeddings, CNN captures n-gram level patterns and emotional cues present in the input sentences. The LSTM model is different, it's all about learning how words in a sentence work together in a sequence. Emotions in sentences often depend on the context and the order of the words, so LSTM networks can remember information from previous words and understand the overall meaning of the sentence. This is helpful because it lets the model capture dependencies between words that are far apart, which is something that CNNs alone can't do very well. The hybrid model that combines CNN and LSTM is really powerful. It uses the CNN part to find important patterns in the data, and then the LSTM part looks at these patterns to see how they relate to each other over time. This way, the model can understand both what's happening in a small area and how things change over time. By putting these two types of models together, we get a system that's better at figuring out what's going on than either of the individual models could be on their own. This means the hybrid model is more accurate and better at classifying things than the CNN or LSTM models used separately.

## XIV. ADVANTAGES

The new approach gets really good results when it comes to classifying things, way better than the old ways of doing machine learning, and that's because it uses some pretty advanced techniques from deep learning.

- CNN layers automatically extract important local features and emotional keywords from textual data.
- LSTM layers capture sequential dependencies and con- textual relationships within sentences.

## XV. APPLICATIONS

- Emotion detection can be used in chatbots to understand user feelings and generate context-aware and personalized responses.
- The system can be applied in Emotion AI applications to automatically recognize human emotions from textual conversations.
- It can be used for social media monitoring to analyze public opinion, detect emotional trends, and understand user reactions.
- The proposed model is useful for sentiment analysis in reviews, comments, and online discussions.
- It can be implemented in customer feedback analysis to identify customer satisfaction and complaints.
- The model can assist mental health monitoring systems by detecting emotional states from user text. It's also really useful in things like recommendation systems, where it can help suggest content to people based on how they're feeling.
- The system is also applicable in virtual assistants for improved human-computer interaction.

## XVI. FUTURE WORK

To make the system work even better, we could use more advanced models like BERT and RoBERTa. These models are really good at understanding the context of what's being said, which would help our system perform even better. To make the model better at recognizing patterns and sorting things into categories, it can be taught with a bigger and more varied set of data. This can help it learn to understand and deal with new and different situations, making it more accurate and reliable.

- Real-time emotion prediction can be implemented for live chat and streaming text applications. You can use this system on the web to find out how people are feeling when they interact with it.

- A mobile application can be developed to enable emotion detection on handheld devices. You can also have emotion detection that works with many languages, which would

- be really helpful for people who speak different languages. We could also look into using other deep learning architectures, like BiLSTM and attention mechanisms, to see if they can help us improve our results.
- The system can be integrated with chatbot frameworks for intelligent conversational agents.

## XVII. CONCLUSION

This paper presented an emotion detection system using deep learning models including CNN, LSTM, and a hybrid CNN-LSTM architecture. The dataset was preprocessed using text cleaning, stopword removal, tokenization, and padding techniques. Traditional machine learning models such as Logistic Regression and Naive Bayes were implemented as baseline approaches, while deep learning models were trained to improve classification performance. Experimental results showed that CNN and LSTM models performed better than traditional methods. The hybrid CNN-LSTM model achieved the highest accuracy by combining feature extraction and sequential learning. The proposed system is efficient, scalable, and suitable for real-world applications such as chatbots, sentiment analysis, and social media monitoring.

## REFERENCES

1. Y. Kim, “Convolutional Neural Networks for Sentence Classification,” EMNLP, 2014.
2. S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, 1997.
3. I. Goodfellow, Y. Bengio, and A. Courville, “Deep Learning,” MIT Press, 2016.
4. D. Jurafsky and J. H. Martin, “Speech and Language Processing,” Pearson, 2020.
5. Y. Goldberg, “Neural Network Methods in Natural Language Process- ing,” Morgan & Claypool, 2017.
6. A. Vaswani et al., “Attention Is All You Need,” Advances in Neural Information Processing Systems, 2017.
7. T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” ICLR, 2013.
8. J. Pennington, R. Socher, and C. Manning, “GloVe: Global Vectors for Word Representation,” EMNLP, 2014.
9. A. Maas et al., “Learning Word Vectors for Sentiment Analysis,” ACL, 2011.
10. R. Socher et al., “Recursive Deep Models for Semantic Compositional- ity,” EMNLP, 2013.
11. F. Chollet, “Deep Learning with Python,” Manning Publications, 2018.
12. S. Ruder, “An Overview of Multi-Task Learning in Deep Neural Net- works,” arXiv preprint arXiv:1706.05098, 2017.
13. J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers,” NAACL, 2019.
14. Z. Yang et al., “Hierarchical Attention Networks for Document Classi- fication,” NAACL, 2016.
15. A. Severyn and A. Moschitti, “Twitter Sentiment Analysis with Deep Convolutional Neural Networks,” SIGIR, 2015.
16. R. Johnson and T. Zhang, “Effective Use of Word Order for Text Categorization,” NAACL, 2015.
17. P. Liu, X. Qiu, and X. Huang, “Recurrent Neural Network for Text Classification,” IJCAI, 2016.
18. S. Lai, L. Xu, K. Liu, and J. Zhao, “Recurrent Convolutional Neural Networks for Text Classification,” AAAI, 2015.
19. M. Peters et al., “Deep Contextualized Word Representations,” NAACL, 2018.
20. Y. Liu et al., “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” arXiv preprint arXiv:1907.11692, 2019.