International Research Journal of Multidisciplinary Scope (IRJMS), 2025; 6(3):1153-1167 IRJMS

Original Article | ISSN (O): 2582-631X DOI: 10.47857/irjms.2025.v06i03.04896

A Hybrid CNN - LSTM Approach for Accident Tweet Classification

Ushnish Pal, Sayandeep Mondal, Md Ashifuddin Mondal*, Rajdip Kundu, Saikat Roy, Subhram Das, Papri Ghosh

Department of Computer Science and Engineering, Narula Institute of Technology, Agarpara, Kolkata, West Bengal, India. *Corresponding Author's Email: ashifuddin.nit@gmail.com

Abstract

Traditional traffic monitoring systems to detect traffic accidents have many problems including high cost to deploy sensors, limited coverage, and data dependability. The purpose of this study is to use social media data to detect road accidents in real time by monitoring tweets. The proposed method uses Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to evaluate and categorize tweets into three categories: Traffic Accident Tweet, Non-Traffic Tweet, and Traffic Information Tweet. This paper compares the proposed hybrid CNN-LSTM model with other Natural Language Processing (NLP) models such as Logistic Regression (LR), Naive Bayes (NB), Support Vector Machine (SVM), and Random Forest (RF) in terms of accuracy, loss, precision, recall and F1 score. The hybrid CNN-LSTM model demonstrated the highest accuracy in detecting accident tweets, achieving an accuracy of 98.5%. The model also showed low loss, indicating reduced error and improved model performance. These results show how well the hybrid deep learning model uses social media to predict traffic incidents accurately in real time. The study emphasizes how social media analytics and artificial intelligence (AI) may be used into traffic monitoring systems to improve emergency response and road safety. The proposed approach provides a more reliable, more affordable and scalable substitute for conventional sensor-based systems.

Keywords: Accident Detection, CNN, LSTM, Traffic Accident, Traffic Tweet, Tweet Classification.

Introduction

Traffic congestion is increasingly becoming a global issue, primarily driven by the rapid rise in population and urbanization (1). Among various contributing factors, non-recurrent events such as road accidents are of particular concern due to the direct threat they pose to human life. It has been reported by the World Health Organization (WHO) that approximately 1.19 million deaths occur annually due to road traffic accidents, making it the leading cause of death for individuals aged 5 to 29. This critical concern has prompted researchers to investigate novel methods for reducing traffic accidents. Traditional traffic monitoring approaches, including sensors and cameras, have encountered limitations such as high operational costs, issues with data reliability, and restricted coverage. These conventional systems often depend on infrastructure that may become inoperative during significant incidents, leading to incomplete data collection. A 5% failure rate in traffic detectors during accidents was reported by the Illinois Department of Transportation (IDOT), which further complicates accident investigation and root cause analysis (2). In light of these challenges, the potential of social media platforms has been explored as an alternative source of real-time traffic information. Due to their massive user bases and rapid dissemination of information, these platforms have proven to be invaluable for gathering traffic-related data. It has been observed that over 500 million tweets are generated daily on Twitter; with a considerable portion providing real-time traffic updates (3). Frequent user-generated posts about traffic conditions make these platforms a rich source for detecting and predicting traffic incidents. In this study, natural language processing (NLP) techniques are employed by integrating Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to classify and identify traffic-related tweets. CNN has been utilized to capture local Textual patterns, while LSTM has been used to model sequential dependencies. Additionally, the Bidirectional Encoder Representations from

(Received 10th April 2025; Accepted 29th June 2025; Published 29th July 2025)

Transformers (BERT) model has been incorporated to extract contextual embedding effectively. Through tweet content analysis, patterns associated with traffic incidents are identified. By focusing solely on linguistic features, a classification accuracy of over 90% in identifying accident-related tweets has been achieved. This approach holds promise in enhancing real-time traffic monitoring and promoting safer roads by leveraging social media data. The classification of accident-related tweets is considered an essential NLP task due to its potential in improving real-time incident detection and emergency response. With the continuous flow of data from platforms like Twitter, significant opportunities have been identified to rapidly detect traffic incidents. Efficiently filtering such tweets to isolate accident-related content can lead to substantial improvements in emergency service responsiveness and road safety.

Various deep learning models such as Logistic Regression, Support Vector Classification (SVC), Multinomial Naive Bayes, and Random Forest have been widely used for Text classification in this area. The effectiveness of Logistic Regression in event detection has been demonstrated, although its ability to capture complex patterns is limited (4). The simplicity and interpretability of Logistic Regression make it suitable for binary classification tasks, especially when combined with feature extraction methods like TF-IDF. SVC has shown robustness in handling high-dimensional Text data, performing effectively in class separation using linear kernels. Despite its straightforward nature, Multinomial Naive Bayes has proven effective on large datasets (4). Random Forest Classification has also been shown to perform well in handling extensive feature sets and noisy data, particularly during large-scale emergencies like traffic accidents (5).

A comparative analysis was conducted to evaluate these algorithms' performance. Logistic Regression achieved the highest classification accuracy, ranging from 32.43% to 58.50%, although it was noted to be less stable than other methods, which maintained average accuracies between 33% and 45% (6). Naive Bayes was found to outperform Random Forest and SVM by 1–2% in average accuracy, but the differences were statistically insignificant, indicating that while Logistic Regression may be optimal, other classifiers perform comparably in multi-class classification scenarios (6).

However, CNN, LSTM networks, and BERT have been identified as the most effective models for Text classification. Significant improvements in classification accuracy have been observed by combining CNN and LSTM, leveraging the unique strengths of both architectures (7). CNN has been used to extract higher-level n-gram features, while LSTM has been employed to capture long-term dependencies, thereby enhancing sentence modelling. This combined approach has been demonstrated to balance dimensionality reduction and feature preservation, optimizing both training efficiency and accuracy, particularly for longer Texts (8). Performance degradation with shorter Texts has been addressed by incorporating original Textual features (8). A notable improvement in accuracy to 97.68%—28.12% higher than traditional methods—has been achieved using this model, along with enhanced training efficiency (9). Evaluations in recent studies have reported precision, recall, and F1-scores reaching up to 98%, indicating a marked improvement over traditional approaches (10). Further advancements have been introduced through an improved Double Channel (DC) mechanism, which enhances the hybrid CNN-LSTM model by addressing limitations related to ambiguous word meanings (11). This mechanism employs dual channels for word- and character-level embedding. Hybrid Attention is utilized to combine output and state at each time step, thereby enhancing generalization and learning capabilities through weighted summation (11). Applications of the CNN-LSTM hybrid model have been demonstrated in a range of classification tasks, such as pre-miRNA identification, high-resolution melting curve classification, and brain tumour detection (12–14).

BERT has gained recognition as a state-of-the-art model in NLP, particularly due to its ability to understand contextual relationships by processing Text bidirectional. Compared to traditional models, BERT has been shown to require minimal pre-processing and feature engineering while delivering superior accuracy (15). For instance, an accuracy of 93.87% has been achieved in sentiment analysis tasks on the IMDB dataset (15). The application of BERT in multi-class Text classification, particularly for short, incoherent,

and code-mixed Texts, has also been investigated, revealing its effectiveness in such challenging scenarios (16). The integration of BERT with Active Learning strategies has been studied, where selectively annotating a small subset of unlabelled data allowed for efficient model fine-tuning. High F1-scores were maintained even when the labelled dataset was reduced by 85%, showcasing the model's robustness and efficiency in Text classification tasks (17).

The remainder of this paper is structured as follows: Section II outlines the proposed methodology of the system; Section III presents the evaluation of model and result analysis; Section IV presents a detailed discussion and comparative analysis of the model's performance; and Section V concludes with insights and future research directions.

Methodology

This section presents the detailed methodology of our proposed accident tweet classification system. This paper proposes a hybrid CNN-LSTM approach for accident tweet classification to assess and classify tweets into three categories: Traffic Accident Tweet labelled as 1, Non-Traffic Tweet labelled as 0, and Traffic Information Tweet labelled as 2.

Overview of the System

The proposed system has been presented in Figure 1. Figure 1 visually represents the overall pipeline of the proposed hybrid accident tweet classification system. It shows the sequence of steps beginning with data collection, followed by pre-processing, classification using CNN-LSTM, and final output. This helps readers grasp the modular structure of the system and understand how raw tweet data is transformed into meaningful classifications. This section illustrates the entire systematic method for identifying traffic-related tweets using a hybrid model that blends CNN and LSTM networks. The procedure begins with the collection of traffic tweet data, followed by data pre-processing. The dataset comprises approximately 10,000 tweets (18). Each tweet was manually annotated and categorized into one of three classes based on the presence of explicit accident-related terminology, location references, and emergency-related keywords. The data pre-processing stage involves cleaning, normalizing, and preparing the data for further analysis. The quality of the data thus enhances, leading to more precise and dependable analysis. The pre-processed data is then loaded into a hybrid CNN-LSTM model designed to evaluate the Text and extract key features. The model analyses the content to identify and categorize significant patterns. The model then categorizes the tweets as Traffic Accident tweets labelled as 1, Non-Traffic Tweets labelled as 0, and Traffic Information tweets labelled as 2. The final output consists of categorized tweets, which provide important insights into traffic-related data.

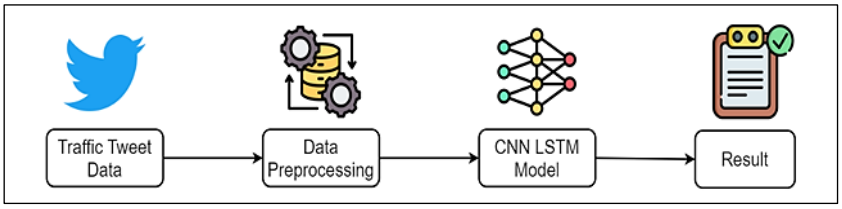

Figure 1: Overview of the System

Overview of the Data Processing Phase

In this section, the pre-processing workflow has been discussed, which is crucial for cleaning and structuring raw data, ensuring accurate and efficient model training and evaluation. An overview of Data Pre-processing phase has been presented in Figure 2. Figure 2 outlines the pre-processing pipeline, starting from Text cleaning (such as lowercasing and punctuation removal) to feature extraction techniques like TF-IDF and BERT embedding. This structured diagram highlights how noise is removed from the raw tweets while preserving informative content necessary for downstream modelling.

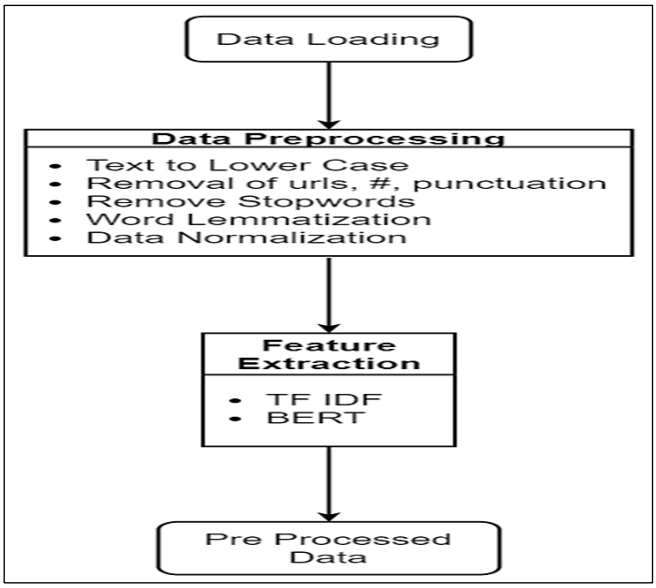

Figure 2: Overview of Data Pre-Processing Phase

The phase begins with the data loading stage, in which raw traffic tweet data is collected from the dataset mentioned before. This stage is critical because it lays the groundwork for all future levels. The quality and completeness of the loaded data have a direct impact on the success of the analysis. Properly loaded data guarantees that no crucial information is lost or misconstrued during the final stages of processing.

Following data loading, the data pre-processing stage is crucial in refining and standardizing raw Text, preparing it for in-depth analysis. This stage involves a number of critical actions, all of which try to change the material into a more consistent and structured manner that analytical models can effectively handle. The initial stage in this process is to convert all Text to lowercase. This seemingly easy activity is critical for guaranteeing consistency throughout the dataset, since it eliminates differences produced by varied capitalizations of the same phrases. This avoids the danger that the analytical model will treat "Apple" and "apple" as two distinct things, improving the analysis's accuracy. Once the Text case is standardized, extraneous elements such as URLs, hash tags, and punctuation marks are eliminated. These components, often present in social media Text, typically do not add useful information for analysis and can instead introduce unnecessary noise. However, certain elements such as location mentions, emoticons, slang, hash tags, and abbreviations were selectively retained or normalized, as they carry important contextual meaning in traffic-related tweets. Location references—such as road names, highways, intersections, or city names (e.g., "I-95," "Main Street," "New York")—were preserved using Named Entity Recognition (NER) techniques to maintain geospatial context, which is critical for real-time traffic applications. Emoticons like "😵," "🚑," or "😠" were mapped to sentiment tokens such as [sad], [ambulance], or [angry], ensuring that their emotional or situational content was not lost during tokenization. Internet slang and abbreviations, commonly found in informal social media language (e.g., "OMG" for "Oh my God," "U-turn," or shorthand like "b4" for "before"), were expanded using a domain-specific slang lexicon to enhance semantic clarity. Hash tags containing meaningful keywords (e.g., "#accident," "#traffic Jam," "#road closed") were retained as informative tokens, while compound hash tags were broken into constituent words using rule-based segmentation (for instance, "#Car Crash" was split into "Car Crash"). This enriched pre-processing pipeline ensured that valuable linguistic and contextual cues embedded in informal tweet language were preserved or transformed appropriately, allowing the model to better interpret and classify accident-related content. By removing these elements away, the model is able to extract more relevant insights by focusing on the Text's primary substance. Additionally, stop words—common words like "and," "the," and "of"—are removed during the pre-processing stage. These words typically do not carry significant meaning in the context of the analysis,

and their removal further reduces noise, distilling the Text down to its most meaningful components. Word lemmatization, or the reduction of words to their base or root forms, is an important component of this stage. For example, words like "running," "ran," and "runs" are all shortened to "run." This procedure not only consolidates different variants of the same term, but it also simplifies the Text, allowing the model to recognize patterns and correlations in the data more easily. Through these thorough pre-processing stages, raw Text is turned into a clean, standardized dataset that is ready for effective analysis, resulting in more accurate and insightful results.

In addition to these activities, data normalization is used to standardize the format and scale of the data. This step is critical to ensuring that the data is consistent and comparable across all records. To avoid complications caused by numerous character encodings, Text can be converted to a standard encoding format, such as UTF-8. Numerical data in the Text is also normalized, either by scaling it to a certain range (for example, 0-1) or by standardizing it to have a mean of 0 and a standard deviation of 1. This normalization procedure harmonizes data from many sources, reduces bias and volatility, and improves the quality and dependability of the study.

Once the data has been pre-processed, the next step is to extract the features. In this stage, the cleaned and standardized language is converted into numerical vectors that represent the data's main properties. Techniques such as TF-IDF (Term Frequency-Inverse Document Frequency) and BERT (Bidirectional Encoder Representations from Transformers) are used for this purpose (19). TF-IDF evaluates the relevance of words in a document in relation to a broader corpus, emphasizing the most informative keywords. BERT, on the other hand, is a deep learning model that generates context-aware embedding based on both the left and right contexts of words in a sentence. Feature extraction approaches are critical for turning Textual input into a format that machine learning models can use efficiently.

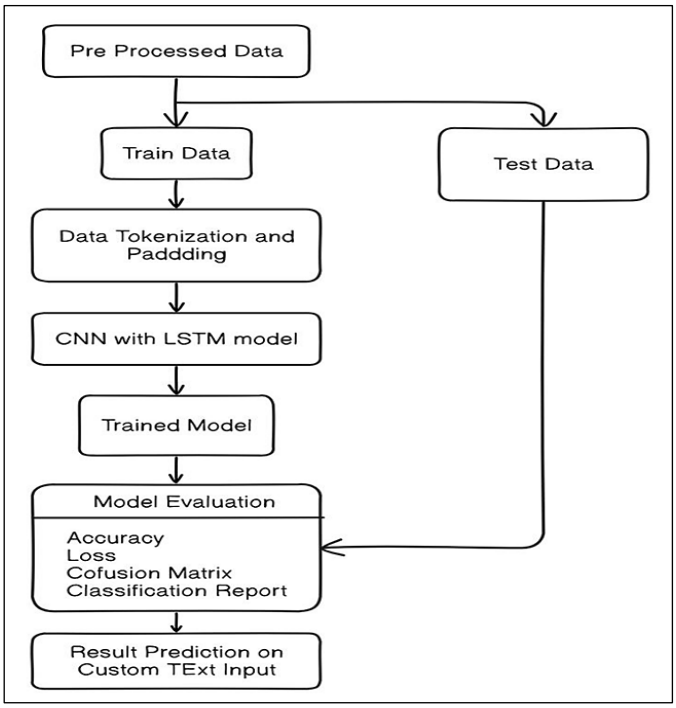

Figure 3: Overview of the Hybrid CNN-LSTM Model

Overview of the Proposed Model

In this section, the working phase of the hybrid CNN-LSTM model has been discussed. This hybrid approach allows the model to recognize both fine-grained characteristics and broader context in sequences, making it extremely useful for applications such as tweet classification. The trained model then assesses and categorizes input data based on the learned patterns. An overview of the hybrid CNN-LSTM model has been presented in Figure 3. As illustrated in Figure 3, the hybrid model architecture includes stages such as tokenization, padding, embedding, CNN-based feature extraction, and LSTM-based sequence

modelling. This figure captures the flow of tweet data through the entire deep learning pipeline, offering a clear overview of how spatial and temporal patterns are handled simultaneously.

The hybrid CNN-LSTM model is an advanced architecture which combines CNN and LSTM to detect both local and long-term correlations in Text, making it very suitable for sorting accident tweets and other sequential data. The procedure starts with tokenization and padding. After tokenization, each tweet is transformed into a sequence of integer indices corresponding to the vocabulary. These sequences are padded to ensure uniform input length, allowing the model to process batches of tweets simultaneously. The padded sequences are then passed through an embedding layer, which maps each word index to a dense, fixed-size vector. This embedding captures semantic similarity between words and prepares the data for convolutional and sequential processing. Data tokenization and padding are important steps in preparing Text data for machine learning algorithms. Tokenization divides Text into smaller parts known as tokens, such as words, sub-words, or characters, and converts the content into a format that models can understand. For example, the phrase "Machine learning is powerful" may be tokenized as ["Machine", "learning", "is", "powerful"]. Each token in the model's vocabulary has its own identifier or vector. Padding ensures that all sequences are of consistent length by adding a specified token, usually zeros, to shorter sequences. Tokenization and padding work together to convert Text into a structured, numerical format that models may use to accomplish tasks like Text classification and sentiment analysis.

After each token in a tweet is assigned an integer index, it is transferred to an embedding layer. This embedding layer transforms these indices into dense vectors of a predetermined size, thereby acting as a lookup table that turns each index into a vector holding semantic meanings and contextual interactions. This embedding enables the model to get a better knowledge of language nuances such as word similarities and grammatical structures, hence improving its ability to detect small changes in meaning and classification accuracy.

After tokens are processed at the embedding layer, the dense word vectors are supplied into a Convolutional Neural Network (CNN). The CNN recognizes patterns within the Text using convolutional filters as sliding windows. These filters are intended to detect local features inside the tweet, such as n-grams or certain word combinations important to the context, such as accident-related data. Pooling layers, such as max-pooling or average pooling, are used to reduce the dimensionality of the data by focusing on the most important features, thereby increasing computational efficiency while retaining essential characteristics that contribute to the model's overall accuracy. In our model, two convolutional layers were used. Each layer consisted of 32 filters with a kernel size of 3, using 'same' padding and ReLU activation. These filters slide across the input embedding to capture local n-gram patterns of size three. The stride was set to 1 to ensure fine-grained scanning of the input. Pooling layers with a pool size of 2 followed each convolution to reduce dimensionality and retain dominant features. These specific hyper parameter choices—number of filters and kernel size—were experimentally selected to balance performance and computational efficiency. For example, increasing the number of filters allows the model to capture a broader range of features, while adjusting the filter size can help in detecting more intricate or larger patterns. A detailed breakdown of the architecture of the CNN layer has been shown in Figure 4. Figure 4 demonstrates the inner workings of the CNN component in the hybrid model. It highlights how convolution and pooling layers are used to extract n-gram features and reduce data dimensionality. The role of multiple filters and ReLU activation is shown as critical for identifying patterns in short tweet Texts.

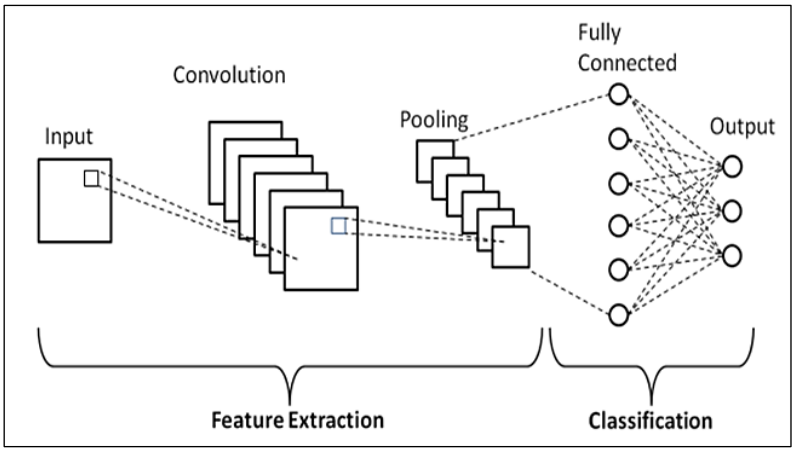

Figure 4: CNN Layer Architecture

Feature maps generated from the CNN are then processed by LSTMs, which handle long-term dependencies with gating mechanisms. Given the feature sequence $h_1, h_1, \ldots, h_m$ produced by the CNN, the LSTM processes this sequence step-by-step, maintaining a hidden state h_t and a cell state c_t at each time step t. The LSTM updates its states using the following equations:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \text{.....[1]}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \text{.....[2]}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \text{.....[3]}$$
$$c_t = f_t . c_{t-1} + i_t . tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \text{.....[4]}$$
$$h_t = o_t . tanh(c_t) \text{.....[5]}$$

where f_t, i_t, o_t, are the forget, input, and output gates, respectively, and σ denotes the sigmoid activation function.

The LSTM layer analyzes feature maps by preserving information over time steps, hence capturing temporal dependencies in tweets. It keeps contextual information from earlier sequence sections while taking into account the influence of later words. This temporal awareness allows the model to grasp how individual attributes interact within the tweet's context. The LSTM improves the model's ability to grasp overall meaning by recalling critical facts over long periods (20), as well as its accuracy in recognizing tiny language distinctions, such as accident-related content against routine traffic data. A detailed breakdown of the architecture of the LSTM layer has been shown in Figure 5. Figure 5 details the internal structure of the LSTM unit, including forget, input, and output gates. These gates help retain or discard information across time steps, enabling the model to understand the contextual flow of a tweet and remember important details across a sequence of words.

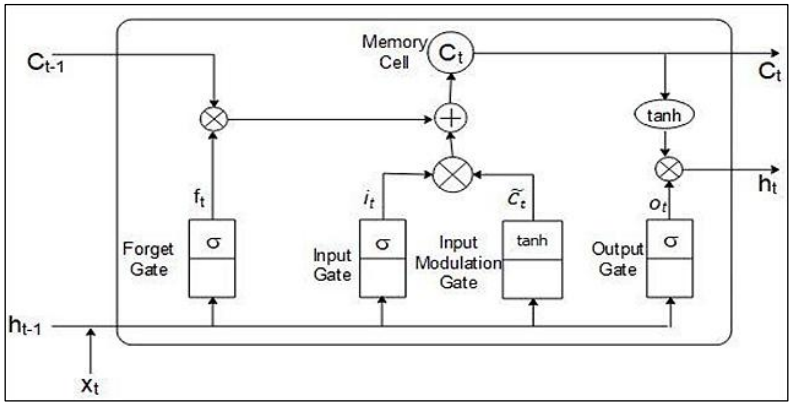

Figure 5: LSTM Layer Architecture

Following the LSTM layer, the output is routed to one or more fully linked (dense) layers. These layers are in charge of learning higher-level representations of the tweet's content using the features and context collected by the CNN and LSTM layers. These deep layers capture abstract patterns, which improves overall categorization accuracy. Activation functions, like as Rectified Linear Units (ReLU), add nonlinearity to the model, allowing it to learn complex correlations and patterns that linear transformations cannot capture. This nonlinearity is crucial for the model to make nuanced decisions and appropriately identify tweets based on complex patterns in the data.

The final layer of the CNN-LSTM model is an output layer, which generates an output vector where each element corresponds to one of the predefined categories (e.g., accident-related, non-accident-related, or normal traffic information). This output vector contains raw scores, or log its, which are then converted into probabilities using a softback activation function. The softback function normalizes these scores so that they sum to one, representing the likelihood of each class. The class with the highest probability is selected as the final prediction, allowing for precise and interpretable tweet classifications based on the patterns learned during training. The categories for classification are: Traffic Accident Tweet [1], Non-Traffic Tweet [0], and Traffic Information Tweet [2]. This categorization procedure is critical in allowing the model to properly distinguish between distinct sorts of tweets based on their content and context. By effectively categorizing tweets, the model ensures that important information, such as accident reports, is distinguished from ordinary traffic updates or irrelevant stuff. This capacity is vital for applications such as real-time traffic monitoring, which requires rapid and reliable information for decision-making and road safety. Once the tweets have been classified, the test dataset is employed to evaluate the model's performance post training. This evaluation assesses the model's generalization, accuracy, precision, recall, and overall effectiveness. The following algorithm shows the workflow of the hybrid CNN-LSTM model.

Algorithm: Hybrid CNN-LSTM Accident Tweet Classification Model

Input: Training Text corpus T.

Output: Classified tweets.

Step 1: Parameter initialization. number of words S, word vector dimension w, sliding window size m, number of projection layer nodes j, iteration number $Enum$, sample error e, threshold i, logarithmic loss function $Loss$.

Step2: Word2vec uses Skip-Gram and CBOW model training to characterize words as vectors:

$$a_j = [c_1, c_2, \dots, c_w] \text{.....[6]}$$

Step 3: Use the CNN sliding window to get the window vector x_k .

$$x_k = [a_k, a_{k+1}, \dots, a_{k+m-1}] \text{.....[7]}$$

Step 4: Maximum pooling after convolution to generate word representation feature λ_k.

$$\lambda_k = (V_{k-m+1}, V_{k-m+2}, \dots, V_k) \text{.....[8]}$$

Step 5: Each α corresponds to λ_T time t, input it into LSTM, and obtain the probability vector o_t.

$$o_t = \sigma(W_o h_{t-1} + U_o \lambda_t + b_o) \text{.....[9]}$$

Step 6: Use the Softmax classifier to classify the output of the LSTM. The probability of classifying x as b is:

$$p(y^i = b | x^i, \theta) = \frac{e^{\theta^T} b^{x^i}}{\sum_{i=1}^{k} e^{\theta^T} b^{x^i}} \text{.....[10]}$$

Step 7: Calculate the loss function$Loss$.

Results

This section presents the evaluation and result analysis of our proposed hybrid CNN-LSTM accident tweet classification system. The result of the proposed system has been compared with other NLP models such as Logistic Regression (LR), Naive Bayes (NB), Support Vector Machine (SVM), and Random Forest (RF).

Evaluation

Model evaluation entails determining accuracy, model loss, and examining the confusion matrix, which comprises true positives, true negatives, false positives, and false negatives. A classification report shows the precision, recall, F1-score, and support for each class. After examination, the model predicts outcomes for both custom and historical Text data. These tests ensure that the model is both reliable and effective at appropriately classifying Text data. A detailed breakdown of the parameters along with their values of the proposed hybrid CNN - LSTM model have been given in Table 1 and Table 2 respectively. Table 3 highlights the performance metrics for hybrid CNN-LSTM model along with other NLP models, while Table 4 displays sample predictions for custom Text inputs.

Table 1: Hyper-parameter of CNN Layer

Layer Type	Parameter	Value
Embedding	Vocabulary Size	20,000
	Embedding Dimension	100
	Input Length	40880
Conv1D (1st Layer)	Number of Filters	32
	Kernel Size	3
	Padding	same
	Activation	relu
MaxPooling1D (1st Layer)	Pool Size	2
Conv1D(2nd Layer)	Number of Filters	32
	Kernel Size	3
	Padding	same
	Activation	relu
MaxPooling1D (2nd Layer)	Pool Size	2
Dense	Number of Units	3
	Activation	softmax
Compilation	Loss Function	categorical_crossentropy
	Optimizer	adam
	Metrics	accuracy

Table 2: Hyper-parameter of LSTM Layer

Layer Type	Parameter	Value
LSTM	Number of Units	100
	Dropout	0.2
	Recurrent Dropout	0.2

Table 3: Evaluation Results of Accident Tweet Classification Models

	CNN LSTM	NB	LG	SVM	RF
ACCURACY	0.97	0.88	0.97	0.97	0.95
Loss	0.23	0.27	0.18	0.66	0.17
Precision	0.98	0.89	0.97	0.97	0.96
Recall	0.96	0.77	0.96	0.96	0.94
F1 Score	0.96	0.78	0.96	0.97	0.95

Table 4: Sample Prediction on a Custom Text Input in the Proposed Model

Text	Prediction Result
UPDATE: Rt. in York County has reopened near Commerce Circle. Detour lifted.	Traffic Information Tweet [2]
Minor injuries reported after vehicle involved in crash at Gayosa Street and N. 16th Street.	Traffic Accident Tweet [1]
I no longer have the energy for meaningless friendships, forced interactions and unnecessary conversations.	Non-Traffic Tweet [0]

Cleared: Crash in Alachua on highway north at Exit CR-, left lane blocked. Content provided by Waze.	Traffic Accident Tweet [1]
Learn more about who the National Weather Service is and what we do to support you!	Non-Traffic Tweet [0]
The left lane of Interstate southbound at mile marker has been reopened. The right lane remains closed.	Traffic Information Tweet [2]
Accident at on ramp of at Scalp Ave.	Traffic Accident Tweet [1]
Multi-vehicle crash in Collier on highway south at Mile Marker, all lanes blocked. Last updated at::36PM.	Traffic Accident Tweet [1]

As part of an accident tweet classification work, the word cloud in Figure 6 depicts the most frequently used words in tweets about accidents. Figure 6 provides a visual summary of the most frequently used words in accident-related tweets. Words like "crash," "highway," "incident," and "vehicle" appear prominently, indicating their high relevance to the target classification. The word cloud reinforces the model's ability to identify these cues for accurate prediction. The larger phrases, such as "highway," "crash," "incident," "cleared," "accident," "construction," and "vehicle," indicate that these terms appear frequently in accident-related tweets, implying that they are important indicators for recognizing and categorizing such tweets.

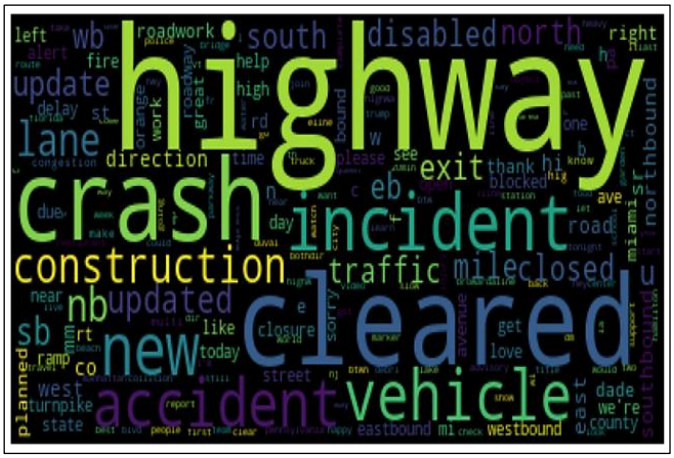

Figure 6: Word Cloud Visualization from Traffic-Accident Tweets

Result Analysis

The Accuracy Comparison of the different Accident Tweet Detection Models has been shown in Figure 7. Figure 7 compares the classification accuracy of various models. The hybrid CNN-LSTM model demonstrates the highest accuracy, validating its superior capability in learning complex patterns from social media Text. This supports the claim that combining CNN and LSTM is more effective than using traditional models alone.

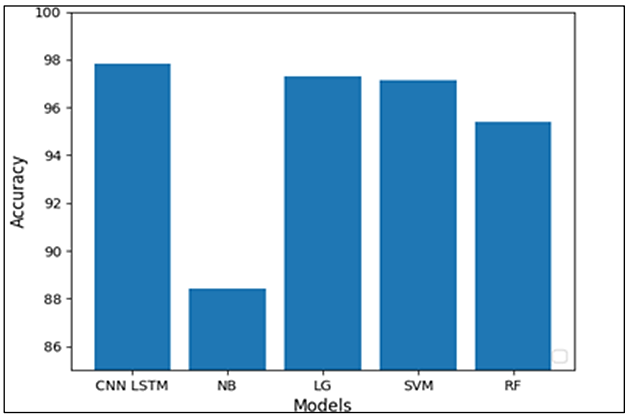

Figure 7: Accuracy Comparison of All Models

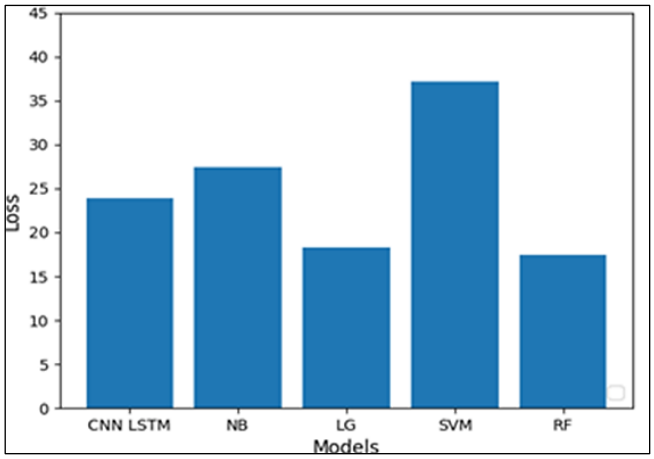

Figure 8: Graphical Representation of Loss of All Models

The Loss Comparison of the different Accident Tweet Detection Models has been shown in Figure 8. As shown in Figure 8, the hybrid CNN-LSTM model and Logistic Regression both achieve low loss values, indicating high reliability in prediction. In contrast, SVM shows a higher loss despite good accuracy, suggesting potential over fitting or sensitivity to feature noise.

The loss comparison shows that SVM has the largest loss, indicating higher prediction mistakes despite its high accuracy, which could be due to over fitting or issues with specific data. Logistic Regression (LG) achieved the lowest loss, indicating extremely accurate predictions. Hybrid CNN-LSTM and Random Forest (RF) showed low loss, indicating effective error reduction. Naive Bayes (NB) exhibited moderate loss, indicating worse prediction accuracy. Overall, Logistic Regression beat others in terms of error minimization, although SVM struggled more with specific data pieces, demonstrating the models' varying error-handling capabilities.

Confusion Matrix Analysis

The confusion matrices in Figure 9 - 13 show that hybrid CNN-LSTM and Logistic Regression excel in classifying Class 0, with 5,021 and 5,076 correct predictions, respectively. However, both models had difficulty differentiating between Classes 1 and 2, with CNN-LSTM producing considerable misclassifications in both categories. Naive Bayes and Random Forest struggle significantly, especially in Class 2, with low classification accuracy. SVM maintains high overall accuracy but exhibits significant misclassification between Classes 1 and 2. Overall, hybrid CNN-LSTM and Logistic Regression surpass Naive Bayes and Random Forest for accurate classification across all classes. A more detailed discussion of each model's confusion matrix follows below each corresponding figure.

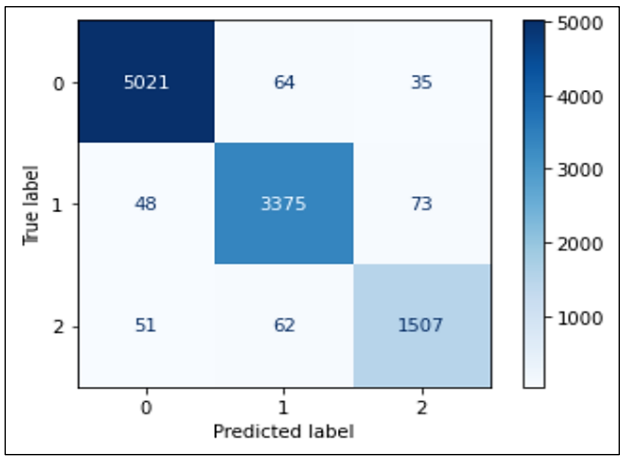

Figure 9: Proposed Hybrid CNN-LSTM Confusion Matrix

Figure 9 illustrates the confusion matrix for the CNN-LSTM model. It shows high accuracy in identifying Class 0 (Non-Traffic Tweets) and strong performance in Classes 1 and 2 as well.

Minor confusion between accident and traffic information tweets indicates areas for potential refinement.

In Figure 10, the Naive Bayes model demonstrates poor classification performance, particularly in distinguishing Classes 1 and 2. This supports the earlier findings that Naive Bayes, while simple and fast, struggles with nuanced distinctions in noisy social media data.

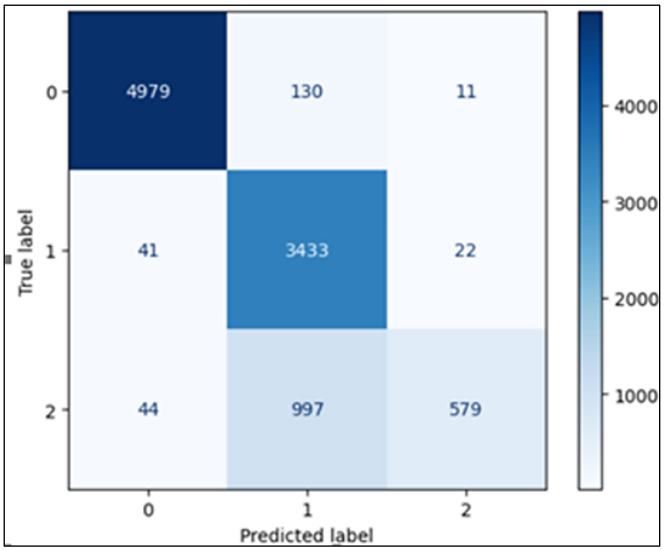

Figure 10: Naive Bayes Confusion Matrix

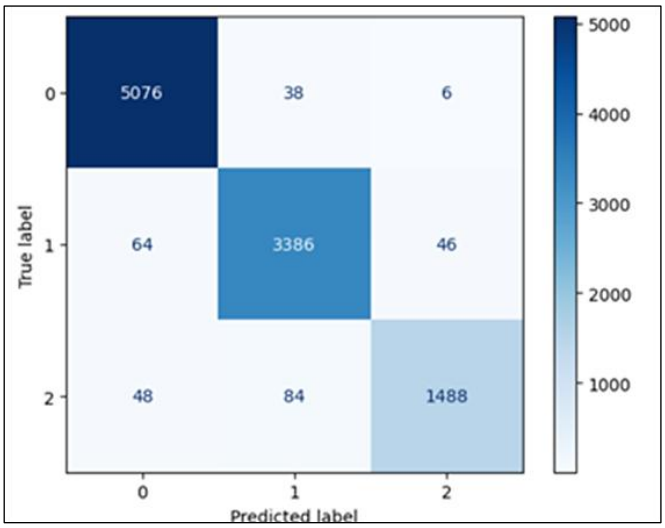

Figure 11: Logistic Regression Confusion Matrix

Figure 11 shows the Logistic Regression model performing very well for Class 0 but with more misclassifications in Classes 1 and 2. Despite its simplicity, Logistic Regression performs comparably to advanced models in certain cases, reinforcing its utility in linearly separable datasets.

Figure 12 reveals that Random Forest performs moderately across all classes but lacks the precision of the hybrid model. While its ensemble nature aids in generalization, it does not handle Textual nuances as effectively as deep learning approaches.

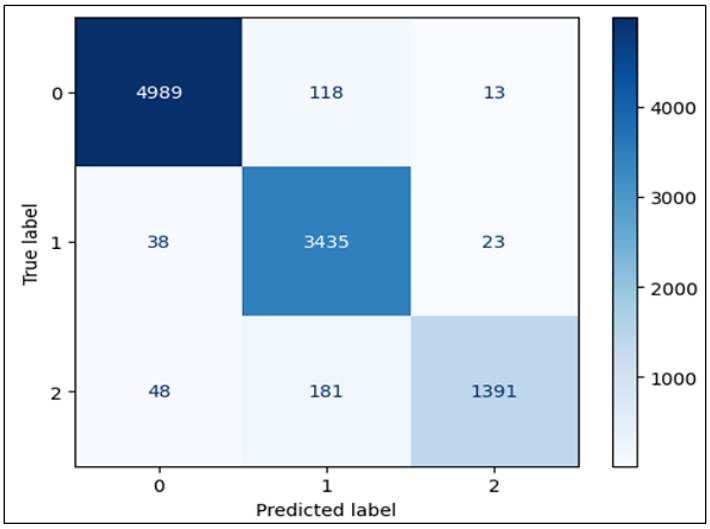

Figure 12: Random Forest Confusion Matrix

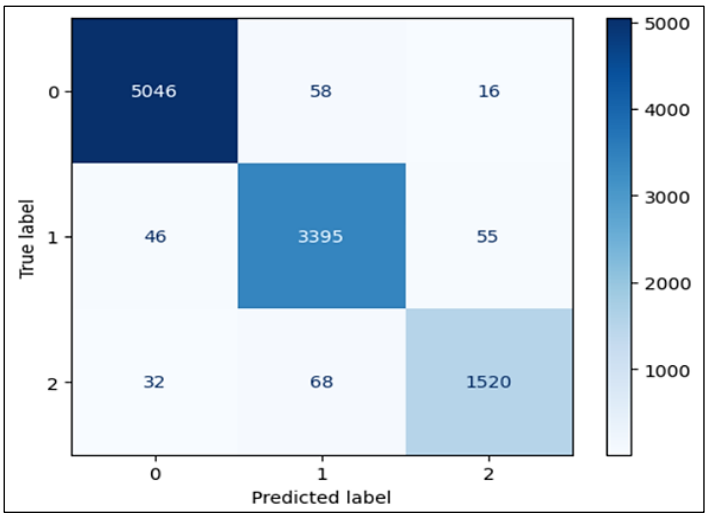

Figure 13: SVM Confusion Matrix

As seen in Figure 13, the SVM model provides high overall classification rates but exhibits notable confusion between Classes 1 and 2. This suggests that while SVM handles high-dimensional feature space well, it may not effectively capture semantic subtleties compared to deep learning models.

A comparative evaluation between the hybrid CNN-LSTM and standalone CNN and LSTM models was also conducted. The hybrid model outperformed standalone models in accuracy and F1-score, indicating that the combination leverages both local feature extraction and temporal dependency handling more effectively than either architecture alone.

Discussion

When comparing accident tweet detection methods, the hybrid CNN-LSTM model achieved the highest accuracy of approximately 98.5%, demonstrating its effectiveness in capturing both local Textual features and temporal dependencies in tweets. Logistic Regression followed closely with an accuracy of around 97%, making it suitable for linearly separable data. SVM also performed well in high-dimensional spaces, achieving 96.5% accuracy, though it showed slightly lower generalization capability compared to Logistic Regression. Random Forest attained about 94% accuracy by leveraging ensemble learning, but struggled to handle complex linguistic patterns as effectively as the hybrid model. Naive Bayes was the least effective, with 88% accuracy, largely due to its simplifying assumptions and limited handling of contextual semantics.

These outcomes align with results from prior studies. For instance, in the past researchers (7, 9) demonstrated that combining CNN and LSTM significantly enhances model performance in

short-Text classification tasks, with reported accuracies up to 97.68%. In another previous study, researchers (10) introduced a Double Channel CNN-LSTM architecture that achieved F1-scores exceeding 98%, reinforcing the hybrid model's capability in extracting both syntactic and semantic patterns.

In addition to the models compared, pre-trained language models such as BERT offer a compelling alternative for accident tweet classification. These models leverage bidirectional attention and deep contextual embeddings to capture word meaning in various linguistic settings. Past researchers (15) reported that BERT outperforms traditional classifiers in sentiment analysis, while another study (16) showed its robustness on short, code-mixed, and noisy Texts like tweets. Although not the main focus of this study, BERT can serve as a strong benchmark in future work. Its ability to generalize across informal and sparse Textual content makes it a promising candidate for comparison with CNN-LSTM, especially in the context of real-time incident detection on social media.

Conclusion

Our study successfully demonstrates the potential of hybrid CNN-LSTM model for identifying accident-related tweets. The model captures both local patterns and long-term dependencies in Text efficiently, thanks to the hybrid structure of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, resulting in high classification accuracy. Our results reveal that hybrid CNN-LSTM is the most effective model in accident tweet detection having an accuracy of 98.5% and an effective error reduction.

The proposed CNN-LSTM approach can be effectively integrated into tools designed for urban planning, traffic control, and emergency response. In traffic control systems, the real-time classification of accident-related tweets can enhance situational awareness by providing immediate and crowdsourced information about ongoing incidents. This allows authorities to make timely decisions regarding traffic signal optimization, route diversions, and congestion management. For urban planning, insights drawn from the spatial and temporal patterns of classified tweets can be used to identify accident-prone zones, informing long-term infrastructure improvements such as redesigning road layouts, adding signage, or modifying lighting and pedestrian crossings. The model can also help assess the effectiveness of past interventions by tracking changes in tweet patterns over time.

In the context of emergency response, real-time detection of accident-related tweets containing location references can enable faster dispatch of ambulances, fire services, and law enforcement, especially when integrated into existing command-and-control systems. Tweets can serve as early indicators of incidents even before official channels are notified, reducing emergency response time. Furthermore, historical tweet analysis can assist agencies in allocating emergency resources more efficiently by identifying peak times and locations for incidents. Overall, this hybrid CNN-LSTM model provides a scalable and adaptive tool that can complement existing traffic intelligence systems, ultimately supporting safer and smarter urban environments.

Abbreviations

CNN: Convolutional Neural Network, LR: Logistic Regression, NB: Naive Bayes, LSTM: Long Short-Term Memory, NLP: Natural Language Processing, ReLU: Rectified Linear Units, RF: Random Forest, SVM: Support Vector Machine, TF-IDF: Term Frequency–Inverse Document Frequency.

Acknowledgement

The authors would like to thank Narula Institute of Technology for their support and resources provided during the course of this research.

Author Contributions

Ushnish Pal: Conceptualization, Methodology, Implementation, Writing Original Draft Preparation, Sayandeep Mondal: Methodology, Data Collection, Writing Original Draft Preparation, Md Ashifuddin Mondal: Proposed Idea, Survey, Conceptualization, Methodology, Supervision, Project Administration, Rajdip Kundu: Data Collection, Writing Original Draft Preparation, Saikat Roy: Writing – Review and Editing, Subhram Das: Manuscript correction, Supervision, Project Administration, Papri Ghosh: Methodology, Supervision, Project Administration.

Conflict of Interest

The authors state that none of their personal ties or known conflicting financial interests might have

seemed to have influenced the work described in this study.

Ethics Approval

Not applicable.

Funding

Not received any funds from any funding agencies.

References

1. Mondal A, Rehena Z. Road traffic outlier detection technique based on linear regression. Procedia Comput Sci. 2020;170:2547-2555.
2. Zhang Z, He Q, Gao J, Ni M. A deep learning approach for detecting traffic accidents from social media data. Transp Res Part C Emerg Technol. 2018;86:580–56.
3. Pandhare KR, Shah MA. Real-time road traffic event detection using Twitter and Spark. International Conference on Inventive Communication and Computational Technologies. 2017; 445-449.
4. McCallum A, Nigam K. A comparison of event models for naive Bayes Text classification. Work Learn Text Categ. 2001;752:41–48.
5. Mohapatra N, Shreya K, Chinmay A. Optimization of the random forest algorithm. Springer Nat. 2020;37:201–208.
6. Pranckevicius T, Marcinkevicius V. Comparison of naive Bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for Text reviews classification. Balt J Mod Comput. 2017;5(2):221–232.
7. Zhou C, Sun C, Liu Z, Lau F. A C-LSTM neural network for Text classification. arXiv. 2015; arXiv: 1511.08630. https://doi.org/10.48550/arXiv.1511.08630
8. Zhou H. Research of Text classification based on TF-IDF and CNN-LSTM. J Phys Conf Ser. 2022;2171:012008.
9. Shoryu T, Wang L, Ma R. A deep neural network approach using convolutional network and long short-term memory for Text sentiment classification. In: Proc IEEE 24th Int Conf Computer Supported Cooperative Work in Design (CSCWD); 2021 May 19–21; Dalian, China. p. 763–768.
10. Liang S, Zhu B, Zhang Y, Cheng S, Jin J. A double channel CNN-LSTM model for Text classification. In: Proc IEEE 22nd Int Conf High Perform Comput Commun; IEEE 18th Int Conf Smart City; 2020. p. 1316-1321.
11. Abdulkadir T, Sen B. A hybrid CNN-LSTM model for pre-miRNA classification. Sci Rep. 2021;11(1): 14125.
12. Ozkok F, Celik M. A hybrid CNN-LSTM model for high resolution melting curve classification. Biomed Signal Process Control. 2022;71(Pt A):103168.
13. Vankdothu R, Hameed A, Fatima H. Brain tumor identification and classification using deep learning based on CNN-LSTM method. Comput Electr Eng. 2022;101:107960.
14. Ismail A, Yusoff R. An efficient hybrid LSTM-CNN and CNN-LSTM with GloVe for Text multi-class sentiment classification in gender violence. Int J Adv Comput Sci Appl. 2022;13(5):853–863.
15. González-Carvajal S, Garrido-Merchán E. Comparing BERT against traditional machine learning Text classification. J Comput Cogn Eng. 2020;1(1):1–10.
16. Prabhu S, Mohamed M, Misra H. Multi-class Text classification using BERT-based active learning. arXiv. 2021. https://doi.org/10.48550/arXiv.2104.14289
17. Ein-Dor L, Halfon A, Gera A, Shnarch E, Dankin L, Choshen L, et al. Active learning for BERT: An empirical study. In: Proc Conf Empirical Methods in Natural Language Processing (EMNLP); 2020. p. 7949–7962.
18. Sina D. Tweets with traffic-related labels for developing a Twitter-based traffic information system [dataset]. Mendeley Data. 2020. https://doi.org/10.17632/c3xvj5snvv.1
19. Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw. 2005;18(5–6):602–610.
20. She X, Zhang D. Text classification based on hybrid CNN-LSTM hybrid model. In: Proc 11th Int Symp Comput Intell Design (ISCID); 2018 Dec 8–9; Hangzhou, China. p. 185–189.