applied sciences

Article

Deep Learning Classification of Traffic-Related Tweets: An Advanced Framework Using Deep Learning for Contextual Understanding and Traffic-Related Short Text Classification

Wasen Yahya Melhem, Asad Abdi * and Farid Meziane

Data Science Research Centre, School of Computing, University of Derby, Derby DE22 1GB, UK; w.melhem1@unimail.derby.ac.uk (W.Y.M.); f.meziane@derby.ac.uk (F.M.)
* Correspondence: a.abdi@derby.ac.uk

Featured Application: Enhancing Intelligent Transport Systems by accurately classifying traffic-related social media messages.

Abstract: Classifying social media (SM) messages into relevant or irrelevant categories is challenging due to data sparsity, imbalance, and ambiguity. This study aims to improve Intelligent Transport Systems (ITS) by enhancing short text classification of traffic-related SM data. Deep learning methods such as RNNs, CNNs, and BERT are effective at capturing context, but they can be computationally expensive, struggle with very short texts, and perform poorly with rare words. On the other hand, transfer learning leverages pre-trained knowledge but may be biased towards the pre-training domain. To address these challenges, we propose DLCTC, a novel system combining character-level, word-level, and context features with BiLSTM and TextCNN-based attention. By utilizing external knowledge, DLCTC ensures an accurate understanding of concepts and abbreviations in traffic-related short texts. BiLSTM captures context and term correlations; TextCNN captures local patterns. Multi-level attention focuses on important features across character, word, and concept levels. Experimental studies demonstrate DLCTC's effectiveness over well-known short-text classification approaches based on CNN, RNN, and BERT.

Keywords: short text classification; BiLSTM–TextCNN integration; multi-level attention mechanism; character–word–concept embeddings; traffic-related social media analysis; intelligent transport systems (ITS)

Citation: Melhem, W.Y.; Abdi, A.; Meziane, F. Deep Learning Classification of Traffic-Related Tweets: An Advanced Framework Using Deep Learning for Contextual Understanding and Traffic-Related Short Text Classification. *Appl. Sci.* **2024**, *14*, 11009. https://doi.org/10.3390/app142311009

Academic Editor: Douglas O'Shaughnessy

Received: 29 September 2024
Revised: 7 November 2024
Accepted: 20 November 2024
Published: 27 November 2024

1. Introduction

Cities worldwide are struggling with severe traffic conditions, demanding innovative management strategies to enhance the experience of road users. While traditional sensor-based methods are effective, they face limitations in coverage and high maintenance costs. Collecting data from social media (SM) platforms like X (previously called Twitter) offers a vast, timely, and freely available source of information, including dedicated pages solely discussing traffic, which can support and supplement sensor-based data. By exploiting diverse sources of data, we can study its effects and potentially support and enhance sensory data. Effectively classifying relevant traffic data from the vast streams of short texts on social media could significantly improve traffic flow, as passersby and passengers can submit real-time information, helping authorities make informed decisions based on people's immediate and real time experiences.

Furthermore, pursuing the study of social media to support traffic management is critical as it employs new and advanced deep learning models to process and interpret social media data, enhancing traffic flow management and forecasting capabilities. Ultimately, integrating social media with sensor data could transform traffic monitoring, offering cities a scalable, cost-effective solution that supports traditional methods while opening new pathways for data-driven, urban decision-making.

The volume of short texts being generated daily by social media (SM) platforms introduces opportunities as well as challenges in the field of natural language processing (NLP). Accurately classifying SM posts, especially in safety-critical domains such as traffic, has become increasingly important yet challenging. Unlike long texts, short texts lack the necessary context, hindering accurate interpretation of their meaning. The varied interpretations of words and use of abbreviations or specific terms like "RT" for Route, "WB" for Westbound, and "ETA" for Estimated Time of Arrival in traffic-related tweets further complicate identifying relevant text, highlighting the need to address nuances for effective short text classification. By classifying these data, incidents can be detected early, improving traffic flow and safety through timely interventions. This research enhances Intelligent Transport Systems by integrating social media data as supplementary to sensors for hands-on traffic management.

The first step in utilizing social media for traffic management involves extracting relevant traffic data accurately classified into irrelevant or relevant traffic events. To develop effective approaches for text relevancy, NLP, machine learning (ML), and deep learning (DL) methods should be employed to accurately categorize SM messages. This process is challenged by data sparsity, imbalance, and ambiguity, as well as the use of slang, acronyms, and abbreviations (SAB), further complicating semantic extraction.

Efforts went to classifying short text from social media to identify relevant text using DL methods to resolve issues in short text, such as work on the integration of context-relevant concepts into Convolutional Neural Networks, developing a model (DE-CNN) that surpassed state-of-the-art methods on three classification tasks through an attention mechanism for concept selection [1]. However, it primarily focused on performance improvements via concept integration and did not address challenges related to handling novel concepts or ambiguous short texts lacking clear context. To address the issue of context-relevant features, multi-stage attention was used, outperforming BERT on most datasets. However, it struggled with rare words to capture character-level semantics [2]. A real-time traffic detection system using X data was proposed, comparing various models and embeddings. The proposed approach, RoBERTa, achieved the highest accuracy (97%), but Word2Vec combined with BiLSTM excelled in handling tweet-length limitations by effectively capturing word semantics [3]. The study highlights that in short texts like tweets, Word2Vec with BiLSTM can outperform more complex models that rely on longer context with an F1 of 96%, exceeding BERT, Word2Vec, GloVe, and XLNet. Traditional methods such as Support Vector Machine (SVM) and Naïve Bayes, as well as DL methods such as CNNs and RNNs, have been used widely in text classification [4]. There remain several lingering issues in the text classification problem. First, data sparsity has been addressed using both features from knowledge bases and part-of-speech tags as well as neural networks to transform words into vectors. However, both have limitations where explicit representations face the problem of data sparsity, and vector representations perform poorly on new and rare words, ignoring the is-A relation, which is important for short text understanding [1]. Second, the ambiguity within short, social media-based texts, compared to paragraphs or documents, does not have sufficient contextual information and suffers from limited word count and incomplete semantics, which leads to semantic ambiguity and poses a significant challenge for short text classification [5]. A problem in previous research on short text classification is the limited focus on integrating multiple embedding layers, such as character embedding, word embedding, and concept embedding, in a unified framework. Aiming to overcome the limitations of equal embedding by capturing richer semantic and syntactic information, leading to improved classification accuracy.

Traditional methods like N-gram-based text categorization struggle with short texts due to data sparsity and lack of context, while SVM face challenges with scalability and capturing complex patterns. Naive Bayes suffers from an unrealistic independence assumption and performs poorly with imbalanced or rare data [6]. Existing approaches to traffic-related social media text classification face limitations, including a lack of multi-level embedding, poor context capture, and difficulty handling rare or new terms, reducing classification accuracy. Current models, such as RoBERTa, while achieving high accuracy (97%), struggle with tweet-length limitations and fail to address the semantic ambiguity in shorter contexts [3]. These gaps indicate a need for integrating character, word, and concept embeddings within a unified framework to enhance semantic and syntactic information capture, critical for improved classification [4]. DLCTC's use of multi-level embeddings and Probase [7] directly tackles these issues by enriching feature spaces and resolving semantic ambiguity, which are essential for accurate social media classification.

The integration of various techniques, such as TextCNN, embedding-based methods, and BiLSTM, plays a crucial role in enhancing short-text classification. TextCNN captures semantic features within the text, showing adaptability for short texts. Embedding-based techniques at character, word, and concept levels provide robust semantic representations, helping models understand complex relationships and context. In DLCTC, character embeddings handle rare word issues, word embeddings provide broader contextual meaning, and concept embeddings clarify semantic nuances, enhancing classification accuracy. BiLSTM supports long-term dependencies and context, especially when paired with attention mechanisms that prioritize key parts of the input. This combination—TextCNN for feature extraction, embeddings for comprehensive representation, and BiLSTM for sequential context—creates a unified model leveraging linguistic features and external knowledge, significantly improving classification accuracy and robustness.

While some progress has been made, many models still fall short in handling informal language, abbreviations, and contextual complexity of traffic-related social media data. This study's contributions address these gaps as follows:

1. Enhance classification through multi-level embeddings (character, word, and concept) enriched by Probase to address data sparsity.
2. Introduce an attention mechanism to dynamically weight context-relevant features, improving the handling of word ambiguity.
3. Combine BiLSTM and TextCNN to capture local dependencies and bidirectional context, overcoming limitations in traditional short-text classification models. In response to these challenges, this study introduces the DLCTC model, a unified multi-level embedding framework that leverages character, word, and concept-level representations and attention mechanisms, improving accuracy and robustness for classifying traffic-related social media text.

The remainder of the paper is structured as follows: Section 2 describes an overview of recent works in this field. Section 3 describes the proposed method using a deep learning NLP framework (DLCTC). Our system evaluation is explained in Section 4. Finally, Section 5 concludes the work with a discussion of the results as well as proposing some future research directions.

2. Literature Review

Determining transport-related text relevance from social media often relies on NLP techniques to analyze and interpret textual data. Traditional methods like Bag-of-Words, rule-based, and dictionary-based techniques often fall short due to their lack of semantic understanding and limited keyword coverage [8]. Supervised ML methods, such as SVM, Naïve Bayes, and Random Forest (RF), are commonly used to automate classification [9]. However, DL models like CNN, RNN, and LSTM offer improved semantic enrichment and relationship identification. Dabiri and Heaslip demonstrated that combining word embeddings with CNN, RNN, and LSTM models can effectively classify traffic-related tweets, achieving high precision [10]. Transformer models like BERT have further advanced the field [8], achieving significant results despite challenges with large dictionaries and term ambiguity. Therefore, modern approaches leverage DL and word embeddings for better semantic representation.

2.1. Short Text Classification Methods

Short text classification has seen major developments with the implementation of various DL methods. LSTMs [11,12] are widely used due to their excellent sequential modeling capabilities, particularly for processing word embeddings like SkipGram [13] and GloVe [14]. Recently, the transformer model [15] has gained prominence due to its attention mechanism, assisting modeling long-range dependencies effectively. Pre-trained models (PTMs) [16], which include methods like MLM [17], have additionally enhanced NLP by learning context-based word embeddings and language modeling from large datasets.

Several CNN variants have been developed to address specific challenges in text processing and classification. For text classification and sentiment analysis, TextCNN [18], developed by Kim et al., leveraged static word vectors from the Word2Vec model to classify sentences effectively [19]. In the stock domain, Biswas applied a bidirectional LSTM network with an attention mechanism for stock price classification; this led to a reduction in the number of parameters while maintaining accuracy [20]. Similarly, a Bi-LSTM method that integrates CNN and word embeddings to assign emotion labels to psychiatric texts was developed, enhancing accuracy through the combination of CNN's feature extraction and LSTM's sequential dependency capture [21,22]. These studies demonstrated the importance of building a classification model by taking into consideration the integration of effective feature extraction, such as variations of CNNs, and the importance of capturing sequential dependencies for enhancing the accuracy and robustness of text processing tasks.

Advancements in embedding techniques have also played a crucial role in improving text processing tasks. Character embeddings have been demonstrated in two methods in the literature. Initially, Claveau introduced out-of-vocabulary (OOV) embedding techniques to expand vocabulary handling, especially for rare or unseen words, which is vital in tasks where vocabulary is large and constantly evolving [23]. Later, Wu and Zhang demonstrated the effectiveness of character-level CNN approaches, which capture detailed linguistic nuances by using character-level information, thus improving tasks like parsing and tagging [24]. These developments highlight how CNN variants and hybrid models have been tailored to specific challenges in text classification, sentiment analysis, and other related tasks, showcasing their benefits in different contexts.

2.2. Short Text Classification Methods for Social Media Traffic Data

The analysis of traffic data across various studies highlights the potential of developed methods on ITS. Early studies, such as the study by Yang, Bekoulis, and Deligiannis, treated traffic event detection as a slot-filling problem. They developed a model combining LSTM and CRF, which performed well on Dutch datasets with accuracies above 97% [25]. However, they noted the necessity for enhanced techniques to capture the entire context within each tweet to improve detection accuracy. Meanwhile, Chen explored the use of multi-modal generative adversarial networks (mmGAN) for traffic event detection, integrating data from sensors and social media [26]. Their model demonstrated superior performance compared to others, although they identified the need to apply attention mechanisms to improve the temporal modeling of multi-modal data.

Moving into more recent research, Suat-Rojas combined doc2vec, TF-IDF, and BERT embeddings to classify tweets related to traffic. Despite challenges with informal language and abbreviations, their study showed the evolving methodologies in improving the detection and classification of traffic-related events, highlighting the importance of precise handling of informal text in social media data [27].

Raksachat and Chuawuthai [28] explored methods to improve the classification of road traffic incident messages on X. They focused on balancing imbalanced datasets through techniques like under-sampling, oversampling, Markov chains, and Bi-LSTM [28]. The results indicated that while under-sampling and oversampling did not perform as well as the baseline, the Bi-LSTM method provided the best classification performance, with an F1-score of 0.63, outperforming the baseline by 15.44%. They concluded that Bi-LSTM is particularly suitable for classifying traffic report messages that resemble time series data.

Recent studies have further advanced these methodologies. For instance, Babbar and Bedi utilized X data and deep learning techniques, notably RoBERTa, and achieved a high accuracy of 97% in classifying traffic events [3]. However, they encountered challenges with Named Entity Recognition (NER), as it often misclassified location names, indicating a need for improvements in location extraction from textual data. Moreover, several novel approaches have been proposed to address challenges like semantic sparsity, feature extraction, and label distribution. Chen et al. introduced a knowledge-enhanced, soft-verbalizer-based prompt-tuning method for multi-label short-text classification, which addresses data sparsity and the long-tail distribution of labels by incorporating external knowledge and leveraging separating soft verbalizers. This approach outperformed baseline methods across benchmark datasets by optimizing prompt templates and soft verbalizers, which helped bridge the pre-training and downstream task gaps typical in language models [29]. Liu et al. proposed an approach that combines BERT embeddings with part-of-speech features for dynamic word vector training, capturing long-distance dependencies in text. This integration enriched semantic recognition and achieved improved classification accuracy on the THUCNews dataset, showcasing its effectiveness in handling semantic insufficiency and sparse feature representation [30].

Hua et al. explored a heterogeneous graph-convolution-network-based approach for short-text classification, enhancing feature representation by linking text with entity and word nodes. SHGCN, tested on datasets like AGNews and R52, demonstrated superior performance by combining BERT embeddings with BiLSTM for capturing deep textual relationships [31]. Similarly, Sun et al. advanced text classification with an adaptive segmentation model, designed to adjust segmentation according to text length and embedding importance. This model, which integrates Word2Vec with positional encoding and a co-attention network, provided flexibility in managing long-sequence truncations and further enriched classification accuracy through deep hidden feature extraction [32]. These studies collectively highlight diverse, graph- and embedding-based methods as valuable for enhancing the robustness and precision of short-text classification in NLP tasks.

A crowdsourcing combined with machine learning to reduce irrelevant data, achieving a region extraction accuracy of 80.4%, this work underscores the importance of refining methods for filtering and categorizing relevant traffic data [33]. Azhar et al. developed a deep learning-based traffic accident prediction model using X data, integrating features such as sentiment analysis, emotions, weather, and geo-coded locations [34]. They achieved an accuracy in the range of 80% to 94%. However, they identified gaps in data availability, particularly regarding weather and location, and highlighted the challenge of detecting fake news to improve accuracy.

Finally, Nirbhaya and Suadaa utilized various machine learning techniques, including SVM [35], Naive Bayes, Logistic Regression, LSTM [36], and IndoBERT, to develop a multi-label classification model for detecting traffic incidents in Jakarta from X texts. The best-performing model achieved a 99.10% F1 score and 99.26% accuracy, offering a reliable prediction system for traffic incidents [9].

Finally, compared to previous studies, our research introduces a novel approach to short text classification for traffic-related social media data using a unified framework. Aiming to capture richer semantic and syntactic information, effectively handling the nuances of social media textual data. Unlike prior methods that often rely on single-level embeddings or struggle with rare words and ambiguous short texts, our model uses the influence of external knowledge from resources like Probase to enrich the feature space and overcome data sparsity. Additionally, by merging BiLSTM and TextCNN with a multi-level attention mechanism.

3. Materials and Methods

In this section, we discuss the architecture of the proposed model, important definitions, the problem statement and the algorithms implemented, and the technologies utilized in this study. The study addresses significant challenges in classifying social media messages, particularly data sparsity, imbalance, and ambiguity, which adversely affect classification accuracy. Data sparsity refers to the limited context and number of words typically found in short social media posts, making it difficult to extract meaningful patterns. Class imbalance occurs when the dataset contains significantly more instances of one class than the other, leading to biased model predictions. Ambiguity arises from the informal language, abbreviations, and slang commonly used in social media posts, which complicates the process of understanding the true meaning of messages [37]. The short text classification problem in our study is structured as follows. Given a short text dataset, each short text is represented by a sequence of characters, words, concepts, and sentences. The character set encompasses all unique characters in the dataset, while the vocabulary includes all the distinct words. Each short text is composed of a sequence of words. Additionally, concepts related to these words are extracted from the Probase knowledge base developed by Microsoft Research Asia, located in Beijing, China [7], forming a set of concepts. These characters, words, and concepts are integrated to create the final feature space for each short text. The objective of short text classification is to train a classifier that maps this extended feature space to the label set. This study focuses on the following problems:

1. SM messages about traffic are often brief and informal, leading to issues like data sparsity, ambiguity, and class imbalance, which complicate accurate classification.
2. Current models struggle with short, context-poor text and fail to effectively handle informal language, domain-specific terminology, and abbreviations commonly found in traffic-related messages.
3. There is a need for a model that can integrate multi-level embeddings (character, word, and concept) with an attention mechanism to accurately classify traffic-related messages, enhancing Intelligent Transport Systems (ITS) with reliable real-time information.

The methodological process employed in this study is illustrated in Figure 1.

Figure 1 displays the framework of our approach with the three stages of the proposed model—an innovative approach that integrates character-level, word-level, and context-relevant features with an attention mechanism based on BiLSTM and TextCNN. It includes the embedding layer, representation layer, and output layer. In the embedding layer, in terms of short text conceptualization using the external knowledge base Probase, the short text is transformed into vectors at the character level, word level, and receipt level, respectively. In the representation layer, to obtain the full feature representations, we divide this layer into multiple stages to achieve different levels of feature representation. As a branch, each stage contains the convolutional module, TextCNN module, and attention module. In the output layer, a softmax is used as a classifier. The framework operates in two phases: the offline phase, where the DLCTC model is trained using labeled data, learning to optimize embeddings and network weights, and the online phase, where the trained model processes incoming social media messages as traffic-related or non-traffic-related. The input to the framework consists of social media posts, represented at the character, word, and concept levels, and the output is a binary classification that identifies whether each message is relevant to traffic incidents. This structured approach enables the model to capture linguistic patterns and contextual relationships, significantly improving classification performance compared to existing methods.

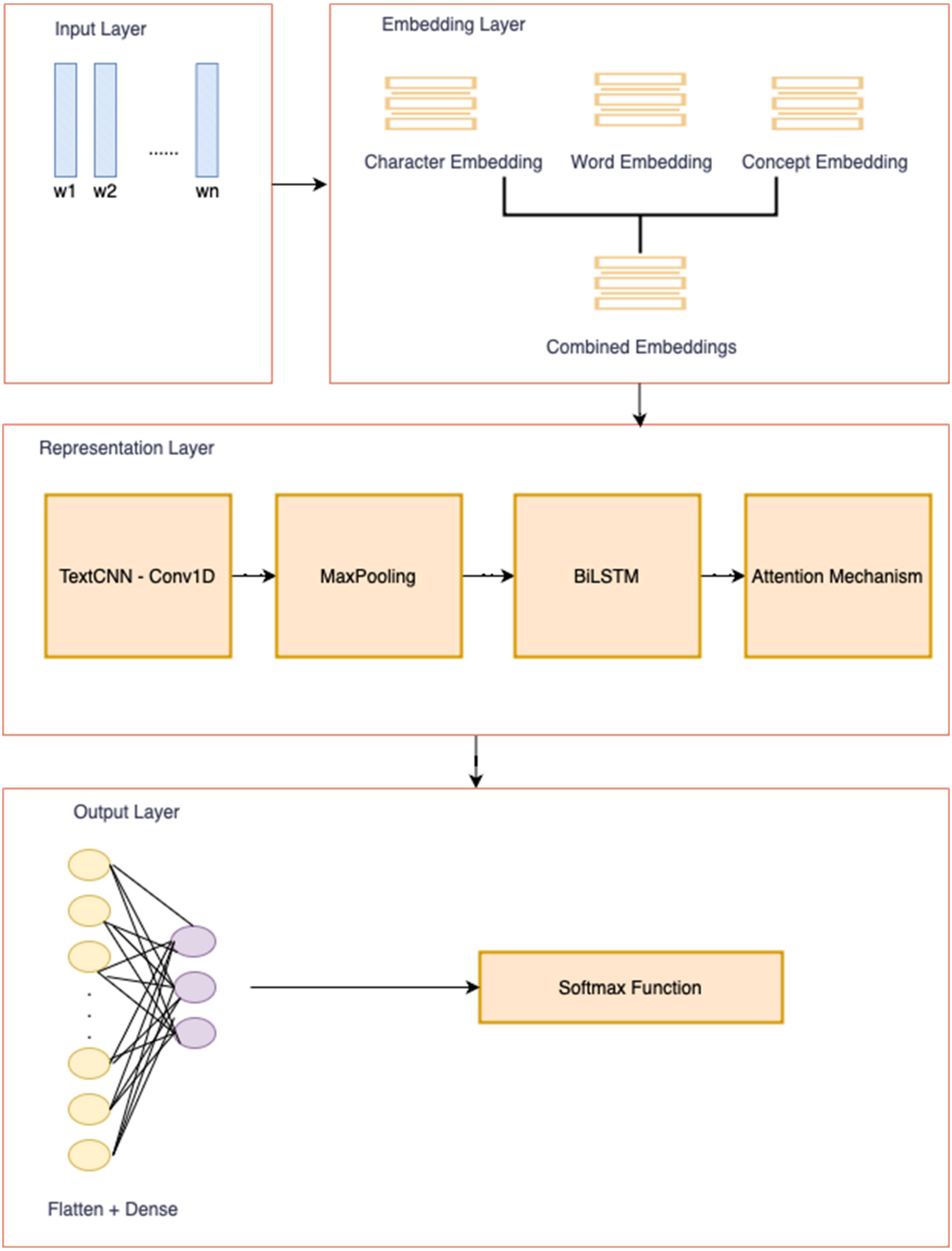

Figure 1. The overall framework of the proposed method.

3.1. Data Preparation

The traffic dataset used in this project was obtained from a combination of the following datasets: GitHub, from the repository titled "Traffic Data Classification Project" available at GitHub (https://github.com/Akanksha242/Traffic-Data-Classification-Project, accessed on 15 October 2024) and collected through the X streaming API; data collected by Dabiri [10], also available on GitHub (https://github.com/sinadabiri/Tweet-Classification-Deep-Learning-Traffic, accessed on 15 October 2024) as well as the X dataset available at Kaggle (https://www.kaggle.com/datasets/mounicapremkumar/traffic-analysis-twitter-dataset, accessed on 15 October 2024). The datasets were combined into one dataset that resulted in 140,000 tweets, stored in a CSV file. Each tweet in the dataset is represented by three parameters: label, X ID, and tweet. The label parameter indicates the classification of the tweet (where 1 denotes a traffic-related tweet and 0 denotes a non-traffic-related tweet), the X ID is a unique identifier for each tweet, and the tweet contains the text content of the tweet. To ensure consistency and compatibility across these datasets, we performed data cleaning and standardization procedures. This involved unifying data formats, resolving discrepancies, and eliminating duplicates. We first split all the tweets in the dataset into training and test sets, with 70% of the tweets for training and the remaining 30% for testing. Table 1 displays a sample of the data.

Table 1. Dataset fields of sample data.

Label	X ID	Tweet
1	s887067566329536512	Disabled Vehicle on Westbound highway WB at Emily Drive. Use caution when traveling through this area.
0	s898251156299927553	New Teacher Lunch & training! Marker wars w/Greta's buzzwords! #PowellNation #ourcougarsourculture
1	s905917482166128641	highway eastbound, all lanes open@ MM, NSP running pace car operations starting near MM.

In this sample, the label column indicates whether the tweet is related to a traffic incident (1) or not (0). A X snowflake unique ID encodes details such as the creation timestamp, datacenter, ID, and machine ID. The timestamp can be decoded to determine the exact time when the tweet or object was created. The tweet column contains the text of the tweet, which includes details about traffic conditions, incidents, or unrelated content. This dataset is valuable for training models to automatically classify tweets based on their relevance to traffic conditions.

Data Preprocessing

In the preprocessing pipeline for the dataset, several essential steps were implemented to prepare the data for the DL and NLP models. First, the tweets were converted to lowercase to ensure uniformity across the dataset. URLs, user mentions, and special characters were removed, as they do not contribute meaningful information to the context of the tweets. Hashtags were removed, but the associated words were retained to preserve context. Tokenization was employed to split the text into individual words or tokens, allowing the analysis of each word separately. A predefined list of English stop words from the NLTK library was used to remove common, non-informative words such as "the" and "is", which are often irrelevant to the core meaning of the tweets. Additionally, lemmatization was applied to reduce words to their base or root forms, ensuring consistency and minimizing dimensionality. Table 2 displays a sample of the pre-processed data.

Table 2. Pre-processed text sample of the tweets.

Tweet	Pre-Processed Text
Disabled Vehicle on Westbound highway WB at Emily Drive. Use caution when traveling through this area.	disable vehicle westbound highway wb Emily Emilye use caution travel area
New Teacher Lunch & training! Marker wars w/Greta's buzzwords! #PowellNation #ourcougarsourculture	New tNewher lunch training marker warGretaa buzzword pollination our cougarsourculture
highway eastbound, all lanes open @ MM, NSP running pace car operations starting near MM.	Highway eastbound lane open nbsp run pace car operation start near mm

Handling informal language such as abbreviations and slang was particularly important for tweets, which often contain shorthand expressions or jargon. The Byte Pair Encoding (BPE) algorithm was used to effectively manage out-of-vocabulary worse-word-up word units, expanding abbreviations into their standard forms. Excess whitespace was also addressed, and emojis were converted into text descriptions using the emoji library to ensure no loss of semantic information. Finally, the processed text was stored in a new column of the dataset and exported as a CSV file, ensuring that the data were properly formatted and cleaned for further analysis or model training.

3.2. DLCTC Model

For the DLCTC model developed in this research, Algorithm 1 is used. The model integrates the following key embeddings: character embedding, word embedding, and concept embedding.

Algorithm 1. Pseudo-code for the DLCTC model for short text classification

Input: Pre-trained PCA-transformed embeddings for character, word, and concepts with FastText: χ_{train}, χ_{test}, Number of training epochs E, Class weights $\boldsymbol{w}_{class}$
Output: Performance metrics: Accuracy, Precision, Recall, F1 Score
$\boldsymbol{\mathcal{X}}_{train} \leftarrow \boldsymbol{Load}(\boldsymbol{train}_{embeddings})$
For (epoch = 1 to E)
    For ($\mathbf{i}$ = 1 to $\boldsymbol{\mathcal{X}}_{train}$)
        Hi←BiLSTM(64)(pool)Hi←BiLSTM(64)(pool)
        Conv1D(256, 3) → BatchNorm Conv1D(256, 4) → BatchNorm Conv1D(256, 5) → BatchNorm
        Ai←MultiHeadAttention(4 heads,64 key dimension) ←MultiHeadAttention(4 heads,64 key dimension)
    End for
    output ← Dense(A, units=1, activation='sigmoid') Output: Trained model weights $\hat{\boldsymbol{\theta}}$.
    Performance loss ← BinaryCrossEntropy(output, labels) UpdateWeights(loss)
    M ← CalculateMetrics(model, validation_data)
End for
return model.weights, M

The model operates in two phases: an offline phase for training the model with labeled social media data and an online phase where the trained model is applied to new messages from the dataset. In the offline phase, the model learns optimal representations and relationships within the training data. In the online phase, these learned representations allow the model to classify incoming messages accurately, providing insights for ITS. This model takes short social media messages as input, represented through multi-level embeddings, and the output of the model is a binary classification that labels each message as either traffic-related or non-traffic-related.

The use of embedding techniques at different levels can effectively capture connections among linguistic units. At the character level, embedding represents characters as vectors in a high-dimensional space, enabling the model to capture their structural and phonetic features. At the word level, embedding represents words as dense vectors in a high-dimensional space, allowing the model to associate words that appear in similar contexts and comprehend a word's meaning based on its use in a sentence. At the sentence level, embedding represents sentences as vectors in a high-dimensional space, enabling the model to capture the overall meaning of a sentence by considering the relationships between individual words and their contexts [4]. The multi-level attention mechanism dynamically assigns weights to the most relevant tokens, emphasizing the critical parts of the text. Finally, the weighted sum of these token embeddings is aggregated to produce a comprehensive sentence-level embedding that represents the entire text, capturing both local token interactions and broader context, which is then used for classification.

Consider the sentence "Crash on I-95 near Exit 24". The DLCTC model processes each word at multiple embedding levels. At the character level, words are broken down into individual characters, capturing morphological details—helpful for interpreting unknown or unusual words. At the word level, each word (e.g., "Crash", "I-95", and "Exit") receives an embedding that reflects its contextual meaning. The concept level links words to broader meanings using Probase, so "Crash" associates with "traffic accident", while "Exit" connects to highway terminology. These embeddings are combined within each token and processed through TextCNN to capture local word associations (like "near Exit") and BiLSTM to understand sequence dependencies (e.g., "Crash on I-95"). A multi-level attention mechanism then dynamically assigns higher weights to contextually relevant tokens, such as "Crash" and "I-95", to emphasize their importance. The result is a comprehensive sentence embedding that represents the sentence's overall meaning, accurately identifying it as a traffic-related event on I-95 near Exit 24.

3.2.1. Character Embedding

The aim of this layer is to map characters in a word to a low-dimensional vector representation by using FastText embeddings; this technique uses skip-gram, which helps in handling out-of-vocabulary words [23]. In this way, the word "I-95" can be broken down into the characters "I", "-", "9", and "5". A character-level embedding model would capture these characters and their patterns. The sequence "I-", followed by a number, is a common format for road or highway designations. The model recognizes that "I" typically stands for "Interstate", and the number represents a specific highway by integrating a knowledge base to map abbreviations to their full forms and synonyms to their related terms. This can provide additional context and improve accuracy. Figure 2 shows the character embedding diagram using FastText.

3.2.2. Word Embedding

Word embeddings here come to represent words in a continuous vector space. Unlike character embedding, which focuses on fine-grained features, word embeddings capture their meanings and relationships based on the co-occurrence of words for semantic understanding, allowing the model to understand and differentiate between similar words as presented in Figure 3. The skip-gram model is used to maximize the probability of a word's surrounding context given the word itself. Equation (1) shows the objective function used, where $\mathcal{T}$ is the total number of words in the corpus, c is the context window size, w_t is the target word at position, t, w_{t+j} are the context words within the window w_t.

$$max\ {}^{1}/_{T} \sum (t=1)^{T} \sum (-c \leq j \leq c, j \neq 0) log\ p\left(w_{(t+j)} \middle| w_t\right) \quad (1)$$

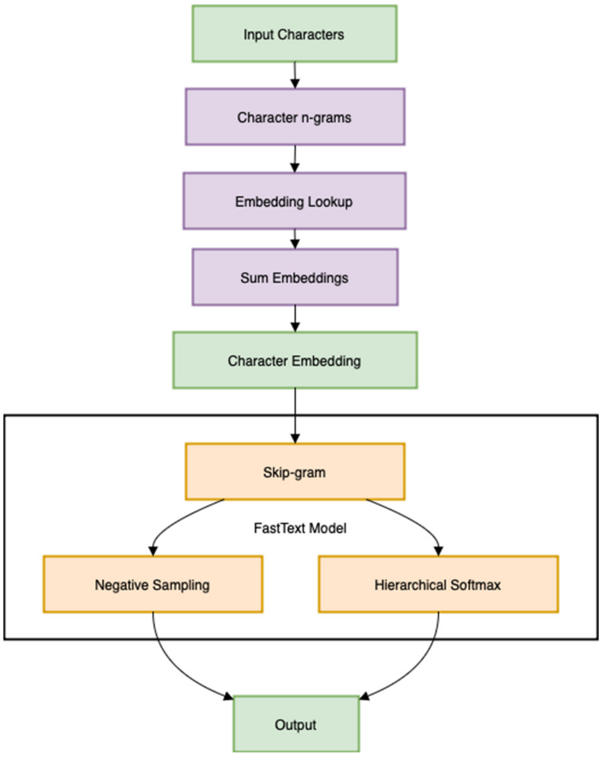

Figure 2. Character embedding framework.

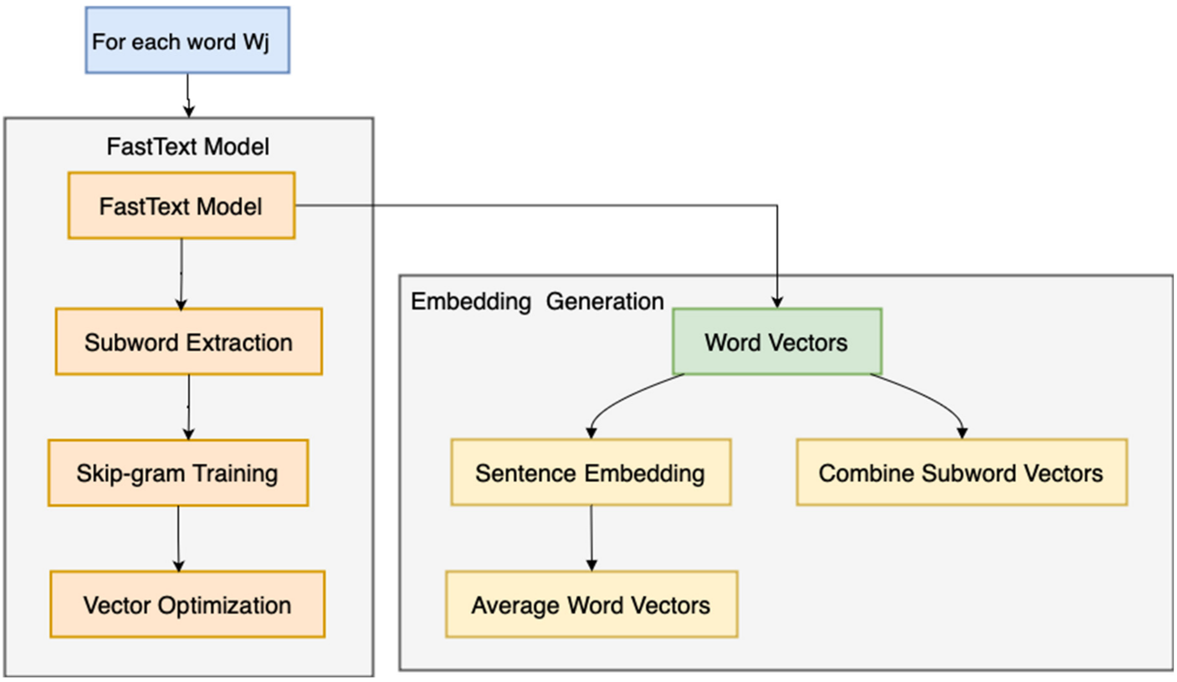

Figure 3. Word embedding framework.

Equation (2) represents the positive score $Score_{pos}$, which is calculated as the dot product between the sum of the subword vectors Z_g, of the input word w_I, and the vector $\mathcal{V}_{w0}$, of the output (context) word w_0. This computation captures the similarity between the input word and its context by aggregating subword information to enhance the word representation.

$$Score_{pos} = \left(\sum_{g \in G_{w_I}} z_g \right)^{\top} v_{w_O} \tag{2}$$

The loss function $\mathcal{L}$ combines the likelihoods of the positive and negative samples, where σ denotes the sigmoid function as given in Equation (3). It aims to maximize the probability of the positive context word by minimizing $-\log\sigma\left(-Score_{neg_i}\right)$, while simultaneously minimizing the probability of negative samples through $-\sum_{i=1}^{k} \log\sigma\left(-Score_{neg_i}\right)$, thereby enhancing the model's discriminative capability.

$$\mathcal{L} = -\log\sigma\left(\mathrm{Score_{pos}}\right) - \sum_{\mathrm{i}=1}^{\mathrm{k}} \log\sigma\left(-\mathrm{Score_{neg_i}}\right) \tag{3}$$

For subword vectors z_g and context word vectors v_w, the embeddings are updated using stochastic gradient descent, where η represents the learning rate. By computing the gradients $\frac{\partial \mathcal{L}}{\partial v_w}$ and $\frac{\partial \mathcal{L}}{\partial z_g}$, the model adjusts the subword and context word vectors to minimize the loss function $\mathcal{L}$, thus refining the embeddings to better capture linguistic relationships in Equations (4) and (5).

$$\mathrm{v_w} \leftarrow \mathrm{v_w} - \eta \frac{\partial \mathscr{L}}{\partial \mathrm{v_w}} \tag{4}$$

$$\mathrm{z_g} \leftarrow \mathrm{z_g} - \eta \frac{\partial \mathscr{L}}{\partial \mathrm{z_g}} \tag{5}$$

The sentence embedding vS is obtained by averaging the embeddings of all words in the sentence, where each word's embedding is the sum of its subword vectors z_g in Equation (6). This process yields a fixed-dimensional representation that encapsulates the semantic content of the entire sentence by aggregating the subword-level information across all N words.

$$vS = {}^{1}\!/_{N} \sum (i=1)^{N} \left[\sum (g \in G_w i) z_g \right] \tag{6}$$

3.2.3. Concept Embedding

The polysemy of words in short texts will lead to the ambiguity problem with just word and character embedding. Therefore, by using concept embeddings, we can distinguish between words based on the context by understanding the variety of concepts for similar words. For example, "crash" belongs to many concepts, such as a computer malfunction or an accident. But it is related to the concept of "accident" because of the occurrence with "South" and "Exit 24". To deal with this issue, we use an external knowledge base, Probase, which contains more concepts than any other knowledge base and has many is-A relationships that can be used to quantify the classification of short texts. Moreover, Probase provides a synonym set and many similar concepts, which help to improve the classification accuracy of short texts. Thus, in terms of the API of the Probase knowledge base, we can obtain the hidden concepts by conceptualizing all words in this short text. For example, the concept set of "Jam" contains "fruit preserves", "situation", and "musical improvisation". By creating specialized embeddings for concepts and weighting them more heavily, the model can better capture their significance.

3.2.4. Attention Layer

The attention layer is used to produce the feature representation for the previous embeddings. As shown in Figure 1, we fed the character-level, word-level, and concept-level embedding into the representation layer. In this layer, we first use the TextCNN model to represent the contexts of short texts on the three layers and then use the attention mechanism to obtain the feature representation by considering the significant features. Therefore, we can obtain different context feature representations at all phases.

3.2.5. TextCNN

TextCNN is a robust deep-learning model commonly used for classifying short text sequences, and it has become a standard benchmark owing to its strong performance. The core concept of this model involves aligning the size of the convolutional kernel in the CNN with the dimensionality of the word embeddings. For example, if each word vector is 300-dimensional, the convolutional kernel can be set to a size of (4, 300). This configuration means that during each convolution operation with a stride of 1, the kernel processes four consecutive word vectors simultaneously. By employing multiple convolutional kernels of varying sizes, CNN can effectively extract features from text sequences of different lengths [38].

3.2.6. BiLSTM

LSTM networks extend traditional RNNs to address issues of vanishing gradients, achieving this by incorporating the input, forget, and output gates and a cell state that allows the network to retain information over long sequences. Consequently, bidirectional LSTM (BiLSTM) is applied to capture information from both past and future contexts in a sequence, containing two separate LSTM layers: one processes the input sequence in the forward direction, and the other processes it backward [39].

Although the forward and backward LSTMs operate independently and have separate parameters, they share the same input word embeddings $\{w_1,\ w_2, ...w_n\}$ where $\boldsymbol{n}$ is the length of the input sentence. This structure enables the model to effectively summarize information from both directions, enhancing its ability to understand contextual nuances in the text.

Contextual information is important for short text classification because it captures the semantic relationships between words and characters. Existing methods use the sequence model based on CNN, RNN [40], LSTM [41], and BERT [42] to capture the context information, which has already gotten a lot of attention due to its effectiveness. However, in the handling of short texts, these RNN-based methods can only process one word in the short text at a time and process the next word after the previous one has been processed and do not consider the high-dimensional and sparse issues due to the limited text length. Consequently, we introduce TextCNN and BiLSTM [38] instead of the RNN and CNN models to capture the contexts of short texts at the character, word, and concept levels. This is because TextCNN effectively captures local dependencies through n-grams, while BiLSTM understands context from both directions in a sequence, which is crucial in short texts [39].

After TextCNN and BiLSTM are applied, and the context of each character, word, and concept is established, an attention layer is proposed to measure the weight of context-relevant characters, words, and concepts, where the weight is assigned dynamically by the attention mechanism. The larger the weight, the greater the importance of the word or the concept; that is, these words and concepts are more important to the classification of the short text. Multi-stage concept feature representations can be obtained by combining all representations of different stages at the concept level. Finally, these features are merged to produce the text representation.

3.2.7. Softmax Layer

In the proposed DLCTC model, the softmax layer plays a crucial role by converting the final feature representations into class probabilities for short text classification. After processing through the embedding and representation layers, the model outputs logits for each class, which the softmax function normalizes into probabilities that sum to 1. This normalization allows the model to determine the likelihood of a short text being classified as "traffic-related" or "traffic-related" by providing a probabilistic interpretation of the model's confidence. For instance, if the softmax layer outputs a higher probability for "traffic-related", the text is classified accordingly. This probabilistic output not only facilitates decision-making but also helps evaluate the model's performance and manage challenges such as data imbalance.

3.3. Experimental Environment and Hyperparameter

The experiments were conducted using Google Colab and TensorFlow with its high-level API, Keras, as the deep learning framework. Table 3 outlines the specific experimental parameters.

Table 3. Experimental parameters.

Hyperparameter	Value
Optimizer	Adam
Learning rate	0.001
LSTM units	256
Batch size	64
Dropout (1st layer)	0.4
Dropout (2nd layer)	0.3
Max pool size	2
Epochs	20
Validation split	0.2
Kernel sizes (conv1d)	3, 4, 5
Conv1d filters	256
BiLSTM layer	Bidirectional
Class weight	Balanced

3.4. Evaluation Indicators

Accuracy is a widely used evaluation metric for assessing the performance of machine learning models, particularly in classification tasks. It measures the overall proportion of correctly classified samples. Accuracy is beneficial as it provides a high-level understanding of a model's performance, but it may not tell the whole story, especially for datasets with imbalanced class distributions.

In addition to accuracy, other important performance metrics include precision, recall (also known as sensitivity), and the F1 score. Precision indicates how reliable the model's positive predictions are. A high precision means the model rarely misclassifies negative instances as positive. Recall, on the other hand, measures the proportion of true positive predictions among all the actual positive instances in the dataset. The F1 score combines precision and recall into a single metric, providing a balanced measure of the model's performance.

4. Results

We performed experiments on three datasets, each divided into training and testing sets with a 70/30 ratio as discussed in Section 3.

Comparison with Other Deep Learning Models

We evaluated the performance of the following four models using the collected dataset:

- Logistic Regression (LR) [40]: A linear classification model that predicts the probability of a class by modeling the relationship between input features and the target variable.
- Support Vector Machine (SVM) [41]: a supervised learning algorithm that finds the optimal hyperplane to separate classes by maximizing the margin between data points.

- Long Short-Term Memory (LSTM) networks [42]: a type of recurrent neural network capable of learning long-term dependencies in sequence data through specialized gating mechanisms.
- Convolutional Neural Networks (CNN) [43]: deep learning models that capture local patterns in data through convolutional layers, often used for image and text classification.

We also evaluated the performance of the following three combination models from the literature on the collected dataset:

- Doc2Vec + SVM [27]: A method where documents are converted into fixed-length vector representations using Doc2Vec and then classified using a Support Vector Machine.
- RoBERTa [3]: A robustly optimized BERT pretraining approach that enhances the BERT model by training with more data and computational power for better language understanding.
- BiLSTM + ELMo + Attention mechanism [44]: A model combining bidirectional LSTMs with ELMo contextualized embeddings and an attention mechanism to capture complex linguistic patterns for improved performance.

The DLCTC model demonstrated a clear advantage over other models on the collected X dataset, achieving an F1 score of 87.64% and an accuracy of 85.14%, as outlined in Table 4. This enhanced performance can be attributed to the model's unique integration of character, word, and concept embeddings, combined with BiLSTM and attention mechanisms. Such architecture effectively captures detailed linguistic features and context, significantly boosting classification accuracy. The use of Probase for concept embeddings adds another layer of understanding, allowing the model to disambiguate meanings in tweets that are often informal and ambiguous. The model's ability to handle nuanced language makes it exceptionally well-suited for analyzing short social media texts.

Table 4. Comparison of the results of four commonly used base models and three hybrid models on the collected datasets.

Model	**Collected X Datasets**			
	Precision (%)	**Recall (%)**	**F1 (%)**	**Accuracy (%)**
LR	79.02	95.41	62.23	75.45
SVM	79.24	95.47	62.26	75.5
LSTM	76.53	95.01	56.92	71.04
CNN	79.02	95.9	61.53	75.11
Doc2Vec+SVM	59.74	77.90	67.62	62.94
RoBERTa	84.32	95.21	75.34	80.88
BILSTM + ELMo + Attention mechanism	78.12	94.30	84.11	79.23
Our model (DLCTC)	79.7890	99.02	87.64	85.14

In contrast, traditional machine learning models like LR and SVM achieved F1 scores of around 62% and accuracies of approximately 75%. While these models provided reasonable performance, their simpler designs were not capable of grasping complex language patterns or understanding the context of social media content as effectively. Deep learning models, such as LSTM and CNN, reached F1 scores of 56.9% and 61.53%. Although these models can learn sequential and spatial features, they cannot handle informal language nuances effectively and do not benefit from the character, word, and concept-level embeddings that DLCTC utilizes.

A significant factor in the DLCTC model's success is the extensive preprocessing it employs. Techniques like text normalization, Byte Pair Encoding (BPE) for handling abbreviations and slang, and converting emojis into text all contribute to ensuring that the input data are both clean and semantically rich. This meticulous approach enables the model to analyze and interpret the tweets more effectively, leading to higher performance metrics.

When comparing RoBERTa with DLCTC and BiLSTM + ELMo + Attention, the distinctions become evident. RoBERTa leverages transformer architecture, using self-attention across tokens to capture bidirectional context and long-range dependencies [45]. This allows it to perform very well on many NLP tasks; however, unlike DLCTC, it does not explicitly use character or concept-level embeddings, which are vital for understanding informal social media language. While RoBERTa may reach high levels of precision and recall, its approach is more generalized compared to the multi-level embedding strategy of DLCTC, which is tailored for the nuances of X data.

The BiLSTM + ELMo + Attention model, while capable of capturing word-level context and emphasizing critical sequence parts through attention, still falls short in comparison to DLCTC due to its lack of character and concept embeddings. ELMo's deep contextual word representations are powerful for text understanding, but the absence of character embeddings limits the model's ability to manage variations in spelling and abbreviation common in tweets. Furthermore, without concept embeddings like those in DLCTC, the model has a limited capacity for semantic disambiguation. Thus, although both RoBERTa and BiLSTM + ELMo + Attention models show strong performance on various NLP tasks, DLCTC's design, which integrates multi-level embeddings and advanced preprocessing, offers a distinct advantage for processing complex and informal social media text.

5. Discussion

The results of our study demonstrate that the DLCTC model outperforms traditional machine learning models and standard deep learning architectures on the collected X dataset. Achieving an F1 score of 87.64% and an accuracy of 85.14%, the DLCTC model effectively addresses the challenges inherent in classifying short, informal texts like tweets.

The strengths of the DLCTC model are evident in its ability to integrate multi-level embeddings (character, word, and concept embeddings) with advanced neural architectures like BiLSTM and TextCNN, which allows for superior handling of informal and ambiguous social media texts. This integration enables the model to capture linguistic patterns at various levels, improving classification performance on short texts. Additionally, the attention mechanism incorporated in the model ensures that the most relevant features are weighted more heavily, further enhancing the model's ability to correctly classify traffic-related messages.

In evaluating the model's performance, training and validation loss were monitored as indicators of how well the model fit the training data and its ability to generalize to new data. Train loss represents the error between the model's predictions and actual labels on the training dataset, showing how well the model learns from data it has seen. Validation loss, on the other hand, measures this error on a separate, unseen validation dataset, which helps assess the model's performance on data outside the training set.

In this study, the final training results showed an accuracy of 85.14% and a train loss of 0.2071, indicating that the model was able to learn and fit the training data well and the validation loss of 0.2011. The close alignment between training and validation loss suggests a good generalization ability, with no significant overfitting. The slightly lower validation accuracy indicates a minor performance drop when encountering new data, which is typical but well within acceptable bounds for robust model performance. Furthermore, the precision of 79.90% and recall of 99.02% suggest the model is highly effective at capturing relevant instances (high recall), though slightly less precise.

Despite these strengths, there are some limitations. For example, while the model performs well on the collected X dataset, the computational complexity introduced by multi-level embeddings and the attention mechanism also increases training time, which may pose a challenge for real-time applications. Further optimization of the model could be explored to address these concerns.

5.1. Learned Word Representations

One of the key factors contributing to the performance of the DLCTC model is its ability to learn rich word representations through the integration of character, word, and concept embeddings. By employing FastText for character embeddings, the model captures morphological and orthographic features, enabling it to understand variations in word forms and handle out-of-vocabulary words. The word embeddings provide semantic understanding based on the context in which words appear, while the concept embeddings derived from the Probase knowledge base allow the model to disambiguate words with multiple meanings by associating them with relevant concepts.

For instance, the word “crash” can refer to a computer malfunction or a traffic accident. Through concept embeddings, the model associates “crash” with the concept of “traffic incident” when it appears alongside words like “I-95 South” and “Exit 24”, enhancing its ability to accurately classify the tweet as traffic-related.

5.2. Impact of Preprocessing Techniques

The extensive preprocessing steps were instrumental in preparing the data for effective modeling. Converting text to lowercase, removing URLs, mentions, and special characters, and handling abbreviations and slang using Byte Pair Encoding (BPE) ensured that the textual data were normalized and free from noise. The lemmatization process reduced words to their base forms, reducing dimensionality and improving the model’s ability to generalize.

Handling informal language, which is prevalent in tweets, was crucial. By converting emojis into their textual descriptions and expanding abbreviations, the model preserved semantic information that could be critical for accurate classification. These preprocessing steps mitigated the challenges posed by the informal and succinct nature of X data.

5.3. Ablation Study

To understand the contribution of each component in the DLCTC model, we conducted an ablation study by modifying parts of the model and observing the impact on performance.

5.3.1. Without Character Embeddings

Removing the character embeddings reduced the F1 score by approximately 5%, with precision dropping to 0.65, recall to 0.78, and accuracy to 81%. This reduction highlights the importance of character-level information, which aids the model in capturing morphological nuances and handling out-of-vocabulary words, a crucial feature when dealing with informal and creatively spelled language typical in social media contexts. Character embeddings thus play a vital role in refining the model’s understanding of word variations, contributing to a balanced recall and precision.

5.3.2. Without Concept Embeddings

Excluding concept embeddings led to a significant 7% drop in the F1 score, underscoring the importance of semantic disambiguation provided by these embeddings. Concept embeddings allow the model to utilize contextual knowledge from the Probase knowledge base, which helps differentiate between words with multiple meanings (e.g., “jam” as a food item versus “traffic jam”). This component is particularly valuable in accurately interpreting traffic-related text by embedding contextual meaning, as shown by the high recall and precision retained in the original model.

5.3.3. Without Attention Mechanism

Removing the attention layer resulted in a 4% decrease in the F1 score, with precision at 0.74, recall at 0.99, and accuracy at 81%. The absence of attention affects the model’s capacity to focus on the most relevant parts of each text, which is essential for distinguishing critical features. The model still achieves high recall without attention, indicating that it can detect most positive cases, but the slight drop in precision suggests an increased rate of false positives when attention is omitted.

5.3.4. Using an LSTM Instead of a BiLSTM

This led to a 3% decrease in F1 score. BiLSTM's ability to process context from both past and future tokens enhances the model's comprehension of short text structure and meaning. This is particularly beneficial in traffic-related text, where word order and context often determine the exact nature of an incident. With only a unidirectional LSTM, the model loses some context, resulting in a slight performance decline in understanding complex word relationships.

5.3.5. Performance of Individual Embedding Types

To further understand each embedding type's contribution, we evaluated the model using only one embedding type at a time:

Character Embedding Only: Precision was 0.65, recall was 0.78, the F1 score was 0.7732, and accuracy was 81%. While character embeddings alone provide substantial recall, they lack the semantic depth needed for high precision.

Word Embedding Only: Precision improved to 0.7811, recall to 0.95, F1 score to 0.84, and accuracy to 81%. Word embeddings alone enable the model to capture semantic information, achieving a more balanced precision and recall compared to character embeddings alone.

5.4. Comparison with Other Models

Traditional models like Logistic Regression and SVM achieved F1 scores of around 62%, highlighting their limitations in handling tweet data. Deep learning models like standard LSTM and CNN performed slightly better but still fell short compared to the DLCTC model. Their inability to fully capture the nuances of informal language and contextual relationships in short texts limited their effectiveness. However, the DLCTC model introduces greater complexity compared to LSTM and CNN, with more trainable parameters and a longer training time per epoch. This added complexity allows for a richer understanding of linguistic features, which is crucial for handling the informal and context-dependent nature of social media text.

The DLCTC model's architecture, which combines multiple levels of embeddings with advanced neural network components, demonstrates the advantage of a more sophisticated approach. By integrating character, word, and concept embeddings, along with Probase for effective term disambiguation, DLCTC captures linguistic patterns at various levels. Combined with the attention mechanism to focus on the most informative features, this approach enhances classification accuracy and robustness.

6. Conclusions

This study introduced the Deep Learning Classification of Traffic-Related Tweets (DLCTC) model, specifically designed to tackle the challenges of classifying short, informal texts from X data. By integrating character, word, and concept embeddings with advanced neural architectures like Bidirectional Long Short-Term Memory (BiLSTM) networks and attention mechanisms, the DLCTC model achieved a superior F1 score of 87.64% and an accuracy of 85.14% on the collected X dataset. This performance significantly surpasses traditional machine learning models such as Logistic Regression and Support Vector Machines, as well as standard deep learning models like LSTM and CNN, which struggled to capture the complex linguistic patterns and contextual nuances inherent in tweets.

The success of the DLCTC model can be attributed to its multi-level embedding approach, which captures morphological nuances, semantic meanings, and contextual relationships within the tweets. The incorporation of the Probase knowledge base for concept embeddings was particularly effective in disambiguating words with multiple meanings based on context, enabling the model to differentiate between terms like "crash" as a traffic incident versus a computer malfunction. The meticulous preprocessing steps—including text normalization, handling of abbreviations and slang through Byte Pair Encoding, lemmatization, and conversion of emojis into textual descriptions—ensured that the data fed into the model was both clean and semantically rich.

An ablation study underscored the importance of each component within the DLCTC model. The removal of character embeddings, concept embeddings, or the attention mechanism resulted in notable decreases in performance, highlighting their essential roles in the model's effectiveness. While the DLCTC model demonstrates significant improvements, certain limitations remain, such as handling extremely noisy data and computational complexity, which may hinder its deployment in real-time applications. Future work could focus on enhancing robustness to noisy data through more sophisticated normalization techniques, optimizing the model for speed, and extending support to multiple languages to increase its utility in global contexts.

In conclusion, the DLCTC model represents a significant advancement in short-text classification within social media contexts. By effectively addressing the challenges posed by the informal and ambiguous language of tweets, it holds promise for applications not only in traffic monitoring but also in areas like sentiment analysis, emergency response, and public opinion tracking. The methodologies and findings from this study contribute to the broader field of natural language processing, particularly in handling noisy and short text data, and pave the way for future research and development in this domain.

Author Contributions: Conceptualization, W.Y.M.; Methodology, W.Y.M., A.A. and F.M.; Software, W.Y.M.; Validation, W.Y.M.; Formal analysis, W.Y.M.; Data curation, W.Y.M.; Writing—original draft, W.Y.M.; Writing—review & editing, A.A.; Visualization, W.Y.M.; Supervision, A.A. and F.M.; Project administration, A.A. and F.M.; Funding acquisition, A.A. and F.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work is funded by the University of Derby.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: The authors express their gratitude to the Data Science Research Centre (DSRC) at the University of Derby for their support in research the development. Additionally, the authors extend their thanks to the anonymous reviewers for their valuable contributions to this work.

Conflicts of Interest: There are no conflicts of interest.

References

1. Xu, J.; Cai, Y.; Wu, X.; Lei, X.; Huang, Q.; Leung, H.F.; Li, Q. Incorporating context-relevant concepts into convolutional neural networks for short text classification. *Neurocomputing* **2020**, *386*, 42–53. [CrossRef]
2. Liu, Y.; Li, P.; Hu, X. Combining context-relevant features with multi-stage attention network for short text classification. *Comput. Speech Lang.* **2022**, *71*, 101268. [CrossRef]
3. Babbar, S.; Bedi, J. Real-time traffic, accident, and potholes detection by deep learning techniques: A modern approach for traffic management. *Neural Comput. Appl.* **2023**, *35*, 19465–19479. [CrossRef]
4. Taha, K.; Yoo, P.D.; Yeun, C.; Taha, A. Text Classification: A Review, Empirical, and Experimental Evaluation. *arXiv* **2024**, arXiv:2401.12982.
5. Cui, W.; Shang, M. KAGN:knowledge-powered attention and graph convolutional networks for social media rumor detection. *J. Big Data* **2023**, *10*, 45. [CrossRef]
6. Chen, J.; Hu, Y.; Liu, J.; Xiao, Y.; Jiang, H. Deep Short Text Classification with Knowledge Powered Attention. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 6252–6259. [CrossRef]
7. Wu, W.; Li, H.; Wang, H.; Zhu, K. Probase: A probabilistic taxonomy for text understanding. In Proceedings of the SIGMOD '12: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, Scottsdale, AZ, USA, 20–24 May 2012; ACM Digital Library: New York, NY, USA, 2012; pp. 481–492.
8. Fontes, T.; Murcos, F.; Carneiro, E.; Ribeiro, J.; Rossetti, R.J.F. Leveraging Social Media as a Source of Mobility Intelligence: An NLP-Based Approach. *IEEE Open J. Intell. Transp. Syst.* **2023**, *4*, 663–681. [CrossRef]

9. Nirbhaya, M.A.W.; Suadaa, L.H. Traffic Incident Detection in Jakarta on Twitter Texts Using a Multi-Label Classification Approach. In Proceedings of the 2023 10th International Conference on Computer, Control, Informatics and Its Applications: Exploring the Power of Data: Leveraging Information to Drive Digital Innovation, IC3INA 2023, Bandung, Indonesia, 4–5 October 2023; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2023; pp. 290–295. [CrossRef]
10. Dabiri, S.; Heaslip, K. Developing a Twitter-based traffic event detection model using deep learning architectures. *Expert Syst. Appl.* **2019**, *118*, 425–439. [CrossRef]
11. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [CrossRef]
12. Sepp, H.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780.
13. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems (NeurIPS): Mountain View, CA, USA, 2013.
14. Pennington, J.; Socher, R.; Manning, C.D. GloVe: Global Vectors for Word Representation. 1997. Available online: https://aclanthology.org/D14-1162.pdf (accessed on 29 September 2024).
15. Vaswani, A. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
16. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
17. Dong, L.; Yang, N.; Wang, W.; Wei, F.; Liu, X.; Wang, Y.; Gao, J.; Zhou, M.; Hon, H.W. Unified Language Model Pre-training for Natural Language Understanding and Generation. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems (NeurIPS): Vancouver, BC, Canada, 2019; Available online: https://github.com/microsoft/unilm (accessed on 29 September 2024).
18. Chen, H.; Zhang, Z.; Huang, S.; Hu, J.; Ni, W.; Liu, J. TextCNN-based ensemble learning model for Japanese Text Multi-classification. *Comput. Electr. Eng.* **2023**, *109*, 108751. [CrossRef]
19. Kim, Y. Convolutional Neural Networks for Sentence Classification. 2014. Available online: http://nlp.stanford.edu/sentiment/ (accessed on 29 September 2024).
20. Biswas, S. Stock Price Prediction using Bidirectional LSTM with Attention. In Proceedings of the 2022 1st International Conference on AI in Cybersecurity, ICAIC 2022, Victoria, TX, USA, 24–26 May 2022; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2022. [CrossRef]
21. Tang, H.; Mi, Y.; Xue, F.; Cao, Y. An Integration Model Based on Graph Convolutional Network for Text Classification. *IEEE Access* **2020**, *8*, 148865–148876. [CrossRef]
22. Yousaf, A.; Umer, M.; Sadiq, S.; Ullah, S.; Mirjalili, S.; Rupapara, V.; Nappi, M. Emotion Recognition by Textual Tweets Classification Using Voting Classifier (LR-SGD). *IEEE Access* **2021**, *9*, 6286–6295. [CrossRef]
23. Claveau, V. Neural text generation for query expansion in information retrieval. In Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, Melbourne, Australia, 14–17 December 2021; Association for Computing Machinery: New York, NY, USA, 2021; pp. 202–209. [CrossRef]
24. Wu, L.; Zhang, M. Deep Graph-Based Character-Level Chinese Dependency Parsing. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2021**, *29*, 1329–1339. [CrossRef]
25. Yang, X.; Bekoulis, G.; Deligiannis, N. Traffic event detection as a slot filling problem. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106202. [CrossRef]
26. Chen, Q.; Wang, W.; Huang, K.; De, S.; Coenen, F. Multi-modal generative adversarial networks for traffic event detection in smart cities. *Expert. Syst. Appl.* **2021**, *177*, 114939. [CrossRef]
27. Suat-Rojas, N.; Gutierrez-Osorio, C.; Pedraza, C. Extraction and Analysis of Social Networks Data to Detect Traffic Accidents. *Information* **2022**, *13*, 26. [CrossRef]
28. Raksachat, T.; Chawuthai, R. A Classification Model for Road Traffic Incidents on Twitter Data. In Proceedings of the ITC-CSCC 2022—37th International Technical Conference on Circuits/Systems, Computers and Communications, Phuket, Thailand, 5–8 July 2022; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2022; pp. 442–445. [CrossRef]
29. Chen, Z.; Li, P.; Hu, X. Knowledge and separating soft verbalizer based prompt-tuning for multi-label short text classification. *Appl. Intell.* **2024**, *54*, 8020–8040. [CrossRef]
30. Liu, S.; Liu, Q. A Deep Learning Short Text Classification Model Integrating Part of Speech Features. In Proceedings of the 2024 4th International Conference on Neural Networks, Information and Communication Engineering, NNICE 2024, Guangzhou, China, 19–21 January 2024; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2024; pp. 550–555. [CrossRef]
31. Hua, J.; Sun, D.; Hu, Y.; Wang, J.; Feng, S.; Wang, Z. Heterogeneous Graph-Convolution-Network-Based Short-Text Classification. *Appl. Sci.* **2024**, *14*, 2279. [CrossRef]
32. Sun, G.; Cheng, Y.; Zhang, Z.; Tong, X.; Chai, T. Text classification with improved word embedding and adaptive segmentation. *Expert. Syst. Appl.* **2024**, *238*, 121852. [CrossRef]
33. Kim, Y.; Song, S.; Lee, H.; Choi, D.; Lim, J.; Bok, K.; Yoo, J. Regional Traffic Event Detection Using Data Crowdsourcing. *Appl. Sci.* **2023**, *13*, 9422. [CrossRef]
34. Azhar, A.; Rubab, S.; Khan, M.M.; Bangash, Y.A.; Alshehri, M.D.; Illahi, F.; Bashir, A.K. Detection and prediction of traffic accidents using deep learning techniques. *Clust. Comput.* **2023**, *26*, 477–493. [CrossRef]

35. Mathur, A.; Foody, G.M. Multiclass and binary SVM classification: Implications for training and classification users. *IEEE Geosci. Remote Sens. Lett.* **2008**, *5*, 241–245. [CrossRef]
36. Karim, F.; Majumdar, S.; Darabi, H.; Chen, S. LSTM Fully Convolutional Networks for Time Series Classification. *IEEE Access* **2017**, *6*, 1662–1669. [CrossRef]
37. Raksachat, T.; Chawuthai, R. Improving a text classifier using text augmentation: Road traffic content from Twitter. In Proceedings of the 2023 20th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON 2023, Nakhon Phanom, Thailand, 9–12 May 2023; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2023. [CrossRef]
38. Dong, X.; Hu, R.; Li, Y.; Liu, M.; Xiao, Y. Text sentiment polarity classification based on TextCNN-SVM combination model. In Proceedings of the 2021 IEEE International Conference on Artificial Intelligence and Computer Applications, ICAICA 2021, Dalian, China, 28–30 June 2021; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2021; pp. 325–328. [CrossRef]
39. Méndez, M.; Merayo, M.G.; Núñez, M. Long-term traffic flow forecasting using a hybrid CNN-BiLSTM model. *Eng. Appl. Artif. Intell.* **2023**, *121*, 106041. [CrossRef]
40. Liu, G.; Guo, J. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing* **2019**, *337*, 325–338. [CrossRef]
41. Zhang, X.; Wang, Z. Stability and robust stabilization of uncertain switched fractional order systems. *ISA Trans.* **2020**, *103*, 1–9. [CrossRef]
42. Sun, C.; Qiu, X.; Xu, Y.; Huang, X. How to Fine-Tune BERT for Text Classification? *arXiv* **2019**, arXiv:1905.05583.
43. Kolekar, S.S.; Khanuja, H.K. Tweet Classification with Convolutional Neural Network. In Proceedings of the Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 16–18 August 2018.
44. Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S.R. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. *arXiv* **2018**, arXiv:1804.07461.
45. Xie, J.; Chen, B.; Gu, X.; Liang, F.; Xu, X. Self-Attention-Based BiLSTM Model for Short Text Fine-Grained Sentiment Classification. *IEEE Access* **2019**, *7*, 180558–180570. [CrossRef]