Journal of Rail Transport Planning & Management 20 (2021) 100265

Contents lists available at ScienceDirect

Journal of Rail Transport Planning & Management

journal homepage: www.elsevier.com/locate/jrtpm

Multi-class railway complaints categorization using Neural Networks: RailNeural

Meenu Gupta [a], Anubhav Singh [b], Rachna Jain [e], Anmol Saxena [c], Shakeel Ahmed [d,*]

[a] Chandigarh University, Punjab, India
[b] Indraprastha Institute of Information Technology Delhi, India
[c] Bharati Vidyapeeth's College of Engineering, New Delhi, India
[d] College of Computer Sciences and Information Technology, King Faisal University, Alhassa, Saudi Arabia
[e] Bhagwan Parshuram Institute of Technology, New Delhi, India

ARTICLE INFO

Keywords:
LSTM
Convolutional neural network
Multi-classification
Text classification
CRIS
Twitter
Bidirectional LSTM
COMS
Attention
RailMadad

ABSTRACT

Indian railways are one of the largest rail networks in the world, and millions of passengers travel daily through it, due to which there are also a vast number of complaints in front of Indian Railways coming every minute through various mediums like COMS (Complaint Management System) app, RailMadad, SMS etc. Given the top-down approach which is followed for the uncategorised complaints making official's work time-consuming. Therefore, faster complaint redressal becomes a critical factor for the passenger's satisfaction. Previous research has focused on traditional machine learning algorithms and Twitter dataset available publicly to tackle this problem. In this work, we have explored deep learning techniques on an official dataset of the COMS app from CRIS (Centre for Railways Information Systems) and proposed RailNeural: an Attention Based Bi-Directional Long Short-Term Memory (LSTM) model which analyses user's complaint input sequences, capturing the underlying character level feature and then classifies them into their respective departments of field units ensuring prompt and accurate redressal of complaints. Our model outperforms several baseline models achieving an accuracy of 93.25 per cent and an F1-Score of 0.93.

1. Introduction

The Indian railways are known for their mileage and rapid increase of speed, even though railways increase their regional and national economic growth. The railway industry attracts many customers not only with their excellent services and even with their extensive framed infrastructure and equipment. Railways provide a high quality of services, increasing attention from media and masses (Indian Railways: Problems, 1963; Benjamin, 2021; Chatterjee, 2019). Wide varieties avail the Indian railways of customers either they may belong to a low-income family (travel through general class) or wealthy family (travel through Air Conditioned class). Indian railways always seek their customers towards their betterment by providing enhanced services. To maintain decorum, Indian railways offer a platform for their customers to give feedback/suggestions/complaints, which can help railways improve their services in the context of the public, and the public can travel stress-free. The platform provided for feedback and suggestion is standard for all

* Corresponding author.
E-mail addresses: meenu.e9406@cumail.in (M. Gupta), anubhav20057@iiitd.ac.in (A. Singh), rachnajain@bpitindia.com (R. Jain), anmsaxena123@gmail.com (A. Saxena), shakeel@kfu.edu.sa (S. Ahmed).

https://doi.org/10.1016/j.jrtpm.2021.100265
Received 12 March 2021; Received in revised form 21 July 2021; Accepted 23 July 2021
Available online 24 September 2021

types of trains such as passenger, local, Rajdhani, Mail, Express, Shatabdi, superfast or any other types of train (Chippagiri, 2021). Customers' complaints may be related to security, food services and quality provided on the train (Stauss and Seidel, 2019), amenities or any other kind of suggestion/complaint are entertained by railways by a different platform such as RailMadad nowadays. The COMS (Complaint Management System) app was used by Indian Railways, backed by the centralised redressal monitoring system.

Customers can drop their complaints and suggestions related to station facilities or problems faced during the travelling time in the CMOS platform.

Due to the increasing number of passengers and mileage, the number of complaints has also increased, making the railway's management department pay attention to passengers' complaints and satisfaction. The management of railways always looks to improve service satisfaction for the railway industry's development and become more popular among people (Garding and Bruns, 2015). For providing better services, Indian railways not only provided platforms such as COMS and RailMadad for putting complaints even different mediums such as telephone, internet, and mail complaints are also there for interaction. Through this medium, railways receive many complaints from their passengers every year, recorded in the text infrastructure (Garding and Bruns, 2015; Jones et al., 2005; Bala et al., 2018). A big question arises in front of railways to automatically classify the urgent problem complaint from this large amount of complaint data received in text form.

Given the top-down approach in the previous system, complaints were sent to branch officers (divisional heads), who then forwarded the complaints to the dedicated workforce deployed to direct the complaints to concerned field units for redressal. This added considerably to the redressal time of grievances, and the multi-layered approach bred delays and wrong alerts. Whereas, in the current system of Complaint Redressal RailMadad, these are sent to field units. So, with the help of the proposed method, complaints may be directly transferred to the service provider of any particular department. Thus reducing the time in the whole complaint redressal process.

Two different methods (i.e., text features extraction and classifier design) can automatically classify the received data. The text feature extraction is a vector space model mainly used to express and describe the text. Word segmentation (or word frequency), a statistical algorithm, can be used to extract the features of each dimension in the vector space model. The discrimination ability of components can be examined with the help of word frequency, information gain, TF-IDF (Namburu et al., 2005; Kim and Gil, 2019), matrix, mutual information etc. Further, features can be extracted through words because the dimension of this model directly obtained using the word as features is too large to solve. In the above-said method, TF-IDF features are also widely used for text feature extraction. To better passengers and fast service assistance, we classify the complaints into its subcategories by applying various algorithms and Bi-directional LSTM (Poria et al., 2018) neural network models for text multi-classification problems. The department who is responsible will immediately get notified, and the complaint may get resolved. The handling of complaints has been outsourced to a trained team that works 24×7 in shifts. Therefore categorization is necessary, and we are optimising the grievance-redressal system. We follow every complaint closely and try to liquidate all the cases daily.

This paper is further classified in different subsections, such as other researcher's views and implemented technologies related to the theme of the paper are discussed in section 2. The collected data set and its pre-processing with the technique implemented is discussed in section 3. Further, the proposed model using LSTM has been covered in section 4. Next, experimental results analysis is discussed in section 5. Finally, this paper is concluded in section 6.

2. Related work

In (Hadifar et al.), Hadifar et al. have explored the domain-specific consumer service NLP model deployment techniques. They have used the pre-training strategies by collecting the multilingual social media data and then provided a comparative analysis of various pre-training and fine-tuning approaches. The proposed models have been then applied to five different end tasks for use in a non-English environment. The work has been focused on moderately sized in-domain datasets.

Akhtar et al. (Akhtar and Sufyan Beg, 2021) have proposed a complaint dataset extracted from Twitter in the context of many tweets in the handles of the Indian Ministry of Railways having fields such as suggestions, grievances, feedbacks and complaints. Collected tweets are modelled as a social network graph with the use of tweet reply features. The author has then used the SVM classifiers with linear kernels for classification purposes. Linguistic features were also considered to capture the semantic meaning of the tweets, which are analysed by the topic labelling task.

Lifeng et al. (Naive Bayesian Automatic, 2019) has used the Naive Bayesian techniques for Railway Service Complaint Text classification in Chinese Railways. The author has identified various types of characteristics in such complaints. Eigenvalue extraction has been used with TextRank and Word2Vec feature extraction techniques and classified with the Naive Bayesian Classifier. The dataset faces the challenges of the repetitive nature of keywords which can be eliminated further to improve the semantic capture capability of the model.

Teng et al. (2018) have analysed the passenger's complaints in the Chinese urban rail transit. The metro organisations utilize the artificial technique to categorise grievances information and make a straightforward outline that is wasteful and improved. In light of that, the author intends to set up a complaint classification system to address such grievances. Moreover, a few examination techniques are incorporated to manage the complete grumbling information, including text mining, geostatistical investigation and complaints redressal system. With multidimensional investigation, thus focussing the trouble spots in traveller complaint redressal system, at last, gives choice help to operational supervisors.

Goyal et al. (2020) have discussed the research perspective of complaints redressal system in Indian railways in social media platforms such as Twitter in high velocity and variety of real-time data context making it hard to process complaints through manual processing. So, to better equip the authorities in handling these complaints. The author has proposed a model using the bigrams and

unigrams features testing with the Naive Bayes and Decision Tree Classifiers. Due to the characters' limitation of 140, Twitter has increased the chances of using slang and abbreviations, which is not seen frequently in official sites such as RailMadad.

Further in (Tong et al., 2018), the authors proposed a novel method as a complaint text classification model based on characters. The main reason for proposing this model is behind complaints. Firstly, authors employ a Negative Element Removal (NER) model to remove the complaints which have a negative expression in their text. Secondly, they worked on a Character-Based Convolutional Network (CBCN) to reduce the effects of grammatical error. As a result, they concluded that their proposed model could achieve state-of-art results on both Chinese and English complaint text compared to another model. Next, in (Chippagiri, 2021), they collected the message dataset from the phones of individual persons to train the system. They implemented the natural language text processing (TF-IDF) technique with the combined implementation of machine learning (i.e., Multinomial Naïve Bayes'). The main focus of this study was to find whether the messages coming from the phone are either standard messages or spam messages.

In (Deng et al., 2019), authors discussed the different machine learning techniques such as the Nearest Neighbor (NN) method, Naive Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT), and Neural Networks (NN) in their study for feature extraction of text. They discussed the mixed data collected from Video, text, images, and audio for their analysis. The main aim of this study was to find the best feature selection methods for text classification. Further in (Stein et al., 2019), they worked on a hierarchical text classification model based on Word Embedding (WE). They used modern machine learning algorithms for automatic document classification. In this study, they investigate the application of the model for a specific problem generated through analysis and extermination. They trained a classification model using a machine learning algorithm on publicly available data and evaluated them with appropriate Hierarchical Text Context (HTC) accuracy measures with 89% accuracy. Further, the Hierarchical Label Set Expansion (HLSE) technique is used on data labels to analyse the impact of different WE models to incorporate grammatical and syntactical features (Gargiulo et al., 2018, 2019).

In (Li et al., 2020), they analyse the capability of the deep learning model on CNN based model for data collected from Twitter classification about Hurricanes Sandy, Harvey, and Irma disaster management. They further classified the collected dataset in different categories such as Caution and Advice, Casualties and Damage, Information Sources, Infrastructure and Resources, and Donation and Aid. As a result, they showed that the proposed model has higher accuracy than the traditional approach for the considered dataset.

Further, in (Liang et al., 2021), they worked on Building Quality Complaints (BQC) for classification in the context of security and health-related resolution to the respective agencies. They applied a CNN-based approach with a deep learning method to automatically classify the short text in BQC. In(Liu and Guo, 2019), they discussed challenges in text data such as high dimensional, sparsity, and complex semantics of the natural language. They proposed a bi-directional LSTM (BiLSTM) method to solve these problems. As a result, they concluded that the proposed model outperforms other state-of-art text classification methods with high accuracy.

Further in (Du et al., 2021), the author discussed the effectiveness of NLP and RNN in text classification. The main challenge, i.e., quantification in the connection between context words in a sentence if further discussed in their work. The proposed RNN model-based self-attention mechanism enhances text performanTheelivering long sentences (in the whole document). They concluded that their proposed model outperforms text classification. In (Nowak et al., 2017), they implemented text classification on a collected dataset from Spambase Data Set; Farm categorized and Amazon book reviews. In this, the authors compared the first two datasets with a feed-forward neural network and the other two datasets with the bag-of-words algorithm. They applied BiLSTM and Gradient recurrent units for their implementation.

Next, in (Nowak et al., 2017; HaCohen-Kerner et al., 2019), they worked on the automatic classification of complaint letters categorized according to their organisation. They applied classification methods based on their organisation insurance, rental cars, hospitals, mobile companies etc. In their findings, they concluded that hospitals are the primary domain itself that has significant issues.

After analysing all the reviewers' views and their implemented technology, we found that every year or Day passengers used the Railways facility. They even faced many difficulties in travelling such as variation of time in departing from one station to another station every day. Indian people met many problems in not solving complaints by Railways on time even though Indian Railways facilitates an Open platform for their passenger's complaint box to provide better services. The previously designed cannot categorise text data received in the form of complaints and sent to their respective department. In this paper, we try to resolve these issues with our proposed model using the LSTM method on Natural Language Processing, further discussed in sections 4, 5, and 6 of this paper.

Text	Complaint	complaint_id	clean_text
Sr. Citizen discount fare in tatkal sewa from ...	Reservation/Enquiry_Office Issues	11	sr citizen discount fare tatkal sewa jammu sta...
RATS ARE PRESENT IN COACH B1 OF TRAIN NUMBER 1...	Maintenance / Cleanliness	7	rat present coach train number ranakpur expres...
12003 coach e1 toilet no 3 leaking vinay pnr 2...	Maintenance / Cleanliness	7	coach toilet leak vinay
Train no . 12321 pnr no 6122312493 complain _ ...	Catering and Vending Services	3	train complain stationtry car meal prize rs.mi...
Train no . 12322 dt.24/08/2013 pnr no 87155750...	Catering and Vending Services	3	train dt complain stationtry car meal prize rs...

Fig. 1. Final Obtained DataFrame representation.

3. Approach used

3.1. The dataset used and pre-processing

Dataset (i.e., real-time data) was collected from the COMS app of Indian Railways in the form of text provided by CRIS (Centre for Railways Information Systems). The data was recorded between April 2018 to June 2019, consisting of nearly 180 K complaints. Initially, the collected data was not cleaned, and it had very high redundancies, which were removed through required text pre-processing techniques. Firstly, few keywords such as '#SEPARATOR#','$', double quotations, and backslashes are used to split the data and convert it into a CSV file, as shown using the NumPy libraries and Pandas library. In Fig. 1 where data was converted into columns such as Complaint, Complaint Id and Complaint. When the data was converted into a CSV file, then 32 different classes were generated. To convert the data into a more informative one, many similar classes have been merged into common departments as different classes of complaints were handled by the same department. Therefore, we finally got it converted into 14 classes by merging such Complaint types (i.e., Misc. Cause, malfunctioning of electrical Equipment, Maintenance/Cleanliness of coaches, Punctuality etc.) as shown in Table 1. Each class contains about 13.3 K complaints, each resulting in equal distribution of instances. It becomes challenging for an organisation to have a structure to mine actionable insights from the text being generated finally, as shown in Fig. 1, the final data frame of the clean dataset received.

Fig. 2 shows the different pre-processing steps applied to the collected data to clean the dataset right from the raw text into a cleaned one. As shown in Fig. 2, the first step in pre-processing the dataset is removing punctuation (i.e., commas and quotes) and white spaces. Removing punctuation and white spaces will help in reducing the size of the training dataset. The next step of pre-processing was removing stop words; these words do not contribute much to the overall context of the sentence that it is communicating. Also, those words were replaced, which does not contribute to the deeper meaning of the concept; for example, the station name was replaced with "railway station". Further, tokenisation is performed for pre-processing the dataset. This part divides the text into a sequence of words or sentences. In this work, the text blob library (Hazarika et al., 2020) (Diyasa et al., 2021) converts complaints into the blob and then converted them into a series of words. After tokenisation, the stemming pre-processing technique is used to remove words like "ing", "ly", "s", etc., by a simple rule-based approach. For performing stemming in this work, the *PorterStemmer algorithm is used* from the NLTK library. Finally, lemmatisation is performed on pre-processing (Vel and SakthiVel, 2021). In this, it converts the words into root words rather than just stripping suffices. Lemmatisation is more effective than the stem, which uses vocabulary and does a morphological analysis to obtain the root word. WordNet Lemmatize (Fellbaum, 2015) is used for the lemmatisation pre-processing step. The last step of pre-processing is to convert tweets text into lowercase to remove multiple copies of the exact words, which might be considered as different words. Finally, the data is clean.

3.2. Data exploration

In data exploration, we focussed on gaining insights into whole data, generating further new information about other processes. At first, we extracted the most frequent words as shown in Table 2, showing the top 9 words in the data with their respective frequencies. The word frequency is also visualisedif, once moren the form of Word Cloud, as shown in Fig. 3.

Further, the average word count feature is used to calculate the average word length of each complaint. This potentially helps in improving the proposed model by adding the length of all word count and applying mean of the complaints. Table 3 shows the average word count by category.

The bigrams are the next feature of data exploration, which is used to generate the pairs of words that frequently counter in the text document such as train number, dear sir, ac_work., as shown in Fig. 4. Identifying such types of pairs in text document are helpful in classification.

Table 1
32 different categories converted into 14 classes after cleaning the data set.

Complaint Category	Compliant Id
Bedroll Complaints	0
Booking of Luggage/Parcels/Goods	1
Bribery and corruption	2
Catering and Vending Services	3
Emergency Assistance	4
Feedback/Suggestions	5
Improper behaviour of non-railway/railway staff	6
Maintenance/Cleanliness	7
Malfunctioning of Electrical Equipment	8
Non-availability of Water Sub	9
Punctuality of Train	10
Reservation/Enquiry Office Issues	11
Thefts/Pilferages	12
Unauthorized passengers in coaches	13

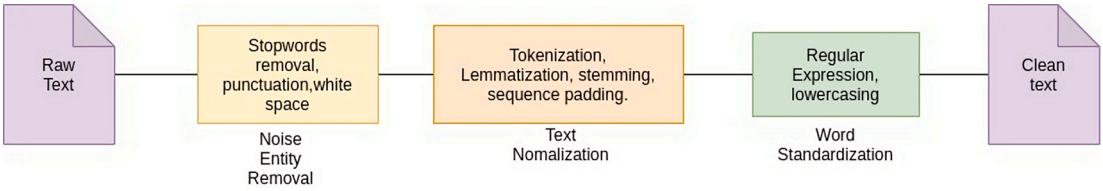

Fig. 2. Pre-processing steps.

Table 2
Most frequent words with their frequencies.

Word	train	coach	PNR	travel	sir	seat	water	work	station
Frequency	215,323	95,779	94,744	41,407	41,319	40,274	39,540	36,183	36,133

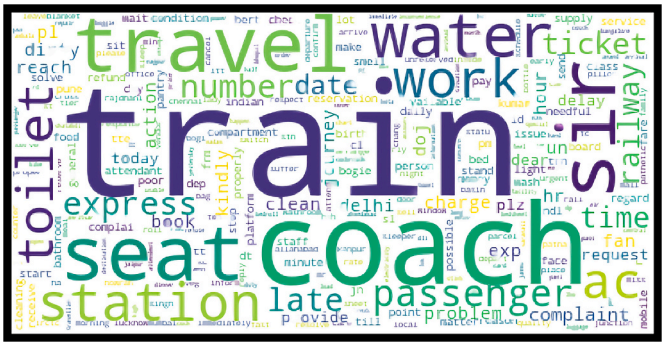

Fig. 3. Word frequencies analysis through WordCloud

Table 3
Average word count with respect to each category.

Complaint Category	Average Complaint Length
Bedroll Complaints	34.67
Booking of Luggage/Parcels/Goods	44.88
Bribery and corruption	55.41
Catering and Vending Services	40.42
Emergency Assistance	33.88
Feedback/Suggestions	42.78
Improper behaviour of non-railway/railway staff	55.26
Maintenance/Cleanliness	28.85
Malfunctioning of Electrical Equipment	25.29
Non-availability of Water Sub	23.94
Punctuality of Train	35.74
Reservation/Enquiry Office Issues	43.01
Thefts/Pilferages	51.10
Unauthorized passengers in coaches	40.21

3.3. Feature engineering

Feature engineering is the technique of changing raw information into highlights that better convey the fundamental issue to the prescient models. Highlight building transforms the contributions to things the calculation can comprehend. To create features from the text, it is necessary to turn the words into numbers. The features can be created from the processed text rather than raw text (i.e., noisy data). To convert words into numbers, it needs to create a matrix with total word count (using count vectorizer); otherwise, Term Frequency-Inverse Document Frequency (TF-IDF) can be used.

We have used two feature engineering techniques: Tf-Idf Matrix and FastText (Yao et al., 2020) (Srinivasa-Desikan, 2018) Embedding trained on our dataset.

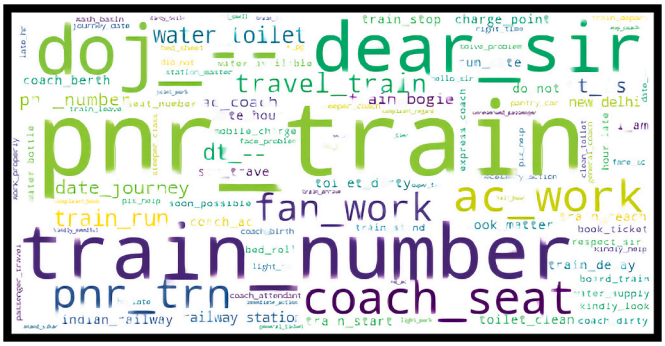

Fig. 4. Bigrams.

3.3.1. *Term Frequency-Inverse Document Frequency (tf-idf) vectors*

On account of the **term frequency** tf (t, d), the least difficult decision is to utilize the *raw count* of a term in a corpus, i.e., the occasions that term t happens in record d shown in Eq. (1). $f_{t,d}$ depicts the raw count whereas the simplest of scheme is tf (t,d) = $f_{t,d}$.

$$f_{t,d} \Big/ \sum_{t' \in d} f_{t'd} \tag{1}$$

The inverse document frequency, shown in Eq. (2), is a proportion of how much data the world gives, i.e., if it's normal or uncommon overall reports. It is the scaled logarithmically opposite division of the archives that contain the word (acquired by separating the all outnumber of reports by the number of records having the term, and afterwards taking the logarithm of that quotient):

$$idf(t, D) = loglog \frac{N}{|\{d \in D : t \in d\}|} \tag{2}$$

where.

N: all number of reports in the corpus N = |D|

It is consequently normal to modify the denominator to 1+ | {d ε D: t ε d}|. Furthermore, |{d ε D: t ε d}| is the quantity of reports when the term t shows up (i.e., tf (t,d) ≠ 0). In the event that the term isn't in the corpus, this will prompt a division-by-zero.

Then tf-idf is calculated as.

Tfidf (t,d,D) = tf (t,d). idf (t,D).

After this, crude content information will be changed to include vectors and new highlights will be made utilizing the current dataset. TF-IDF Vectors as features: Term frequency is essentially the proportion of the inclusion of a word present in a sentence to the length of the sentence, and inverse document frequency (IDF) is used to find a word appearing in all documents, which is not of much use.

3.3.2. *FastText embedding*

FastText was introduced by Facebook in 2016 for efficient word learning representation by utilizing the n-grams of characters helpful in considering the rare words as they are sliced into n-grams. We have used FastText embedding, which is trained on our dataset from scratch using the Gensim library on 30 epochs with a size of 300 dimensions on our own vocabulary. This FastText embedding layer is the input layer for the model, which creates the word embedding layer, which is then fed into the Bi-LSTM layer.

Finally, for pre-processing and tokenisation of the input sequences, we have used:

- Keras Tokenizer Text To sequence: The Keras profound learning library gives some essential instruments to help in getting ready content information. Content information must be encoded as numbers to be utilized as info or yield for AI and profound learning models, where the Coefficient of each token is considered to be in binary terms considering the count of words as in tf-idf.

3.4. *Bi-directional long-short term memory (LSTM)*

The enactment yields from neurons proliferate in the two headings (from contributions to yields and from yields to contributions to) Recurrent Neural Networks model, dissimilar to Feed-forward neural systems in which actuation yields are engendered uniquely one way. This makes circles in the neural system design, which goes about as a 'memory state' of the neurons and permits the neurons a memorable capacity that has been realized up until now. Even if the memory state in RNNs gives a favourable position over customary neural systems, an issue called Vanishing Gradient is related to them. In this issue, while learning with countless layers, it turns out to be extremely difficult for the system to learn and tune the parameters of the prior layers. To address this issue, another kind of RNNs (i.e., LSTMs) model has been developed, which is discussed in section 4.

Eq. (3) and Eq, (8) describes the underlying working mechanism of LSTM mathematically:

$$ltm_t = remember_t * ltm_{t-1} + save_t * ltm_t^{'} \quad (3)$$

$$wm_t = focus_t * tanhtanh\ ltm_t \quad (4)$$

where each memory/attention sub-mechanism is just a mini-brain of its own:

$$remember_t = \sigma(W_r x_t + U_r wm_{t-1}) \quad (5)$$

$$save_t = \sigma(W_s x_t + U_s wm_{t-1}) \quad (6)$$

$$focus_t = \sigma\left(W_f x_t + U_f wm_{t-1}\right) \quad (7)$$

$$ltm_t^{'} = tanh(W_l x_t + U_l wm_{t-1}) \quad (8)$$

- The long-term memory, ltm_t, is usually called the **cell state**, denoted ct. φ is an activation function, regularly picked to be tanh. Here, sigmoid activation since we need numbers somewhere in the range of 0 and 1.
- The working memory, *we*, is normally called the **hidden state**, indicating ht. This is similar to the shrouded state in vanilla RNNs.
- The recollect vector, *remember*, is generally called the **forget gate** (in spite of the way that a 1 in the overlooks door despite everything intends to keep the memory and a 0 despite everything intends to overlook it), indicated *ft*.
- The save vector, $save_t$, is normally called the **input gate** (as it decides the amount of the contribution to allow into the phone state), indicating it.
- The focus vector, $focus_t$, is normally called the **output gate**, indicated ot.

LSTM is aimed to outperform these blunder backflow issues. Long short-term memory is a novel repetitive system engineering related to a suitable slope-based learning calculation. It can find out the way to establish time intervals greater than 1000 stages even if there should arise an occurrence of boisterous, compact groupings of data, having no loss of brief timeframe slack dimension. This is accomplished by optimal, gradient-based computation for an engineering implementing steady (consequently, neither shatter nor disappearing) error steps through the inside situation of typical units (given the inclination calculation is shortened at certain design explicit focuses; this doesn't influence long haul mistake stream, however).

We have then employed the Bi-directional capabilities of the LSTM where two separate LSTM modules acquire the data from two directions of the sentence which is then concatenated to produce a final output layer at each time step. These two LSTM networks are independent of each other but input the common word embedding to produce the final output embedding layer.

3.5. Attention layer

We have then deployed the Attention Layer on the output layer of Bi-Directional LSTM. The attention mechanism allows the Bi-LSTM to capture the semantics of the word applying attention to certain words which are contributing to the final classification. Based on this attention layer aggregates those features to produce a sentence vector using context vectors of their own. These context vectors are a high-level representation for distinguishing the importance of input text which is calculated by the weighted mean of the hidden state which was obtained through the Bi-LSTM layer which is then passed through the softmax function.

4. Model formulation and training

We have set the Limit of the data set to the top 30,000 words. Set the max number of words in each complaint at 60 after analysis explained in Section 3.2. Every row was truncated and input sequences were padded so that they are all in the same length for

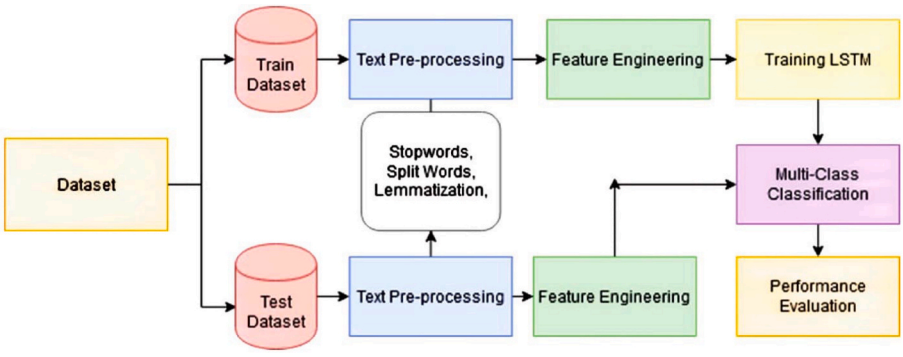

Fig. 5. Proposed model for multi-text classification of railways complaint categorization.

modelling. Then we converted categorical labels to numbers as the output of the model is in numbers. Vector of size 100 is used as an embedded layer that depicts each word.

SpatialDropout1D was used to perform variational dropout in our model before having 100 memory units layer of LSTM. The output layer must create 14 output values, one for each class. SoftMax was used as an Activation function for multiclass classification. Because it is a multiclass classification problem, the loss function is of categorical cross-entropy. The architecture for the model has been shown in Fig. 5 below:

4.1. Softmax activation function

The softmax function squashes the yields of every unit to be somewhere in the range of 0 and 1, much the same as a sigmoid capacity (Milde and Biemann, 2020). In any case, it additionally isolates each yield with the end goal that the all-out entirety of the yields is equivalent to 1. The yield of the softmax work is comparable to a straight-out likelihood conveyance, it reveals to you the likelihood that any of the classes are valid.

Scientifically the softmax work appears in Eq. (9), a vector defined by z of the contributions to the yield layer (in the event that you have 10 yield units, at that point there are 10 components in z). Furthermore, once more, j records the yield units, so j = 1, 2, …, K.

$$\sigma(Z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \tag{9}$$

4.2. Categorical cross-entropy

Categorical cross-entropy is just a single class of material for every datum point. At the end of the day, a model can have a place with one class in particular, it is a loss function that is utilized for single mark arrangement. Eq (10) depicts a cost function having multiclass loss in logarithmic terms having N size.

$$J = -\frac{1}{N} \sum_{i=1}^{N} y_i * log(y_i) \tag{10}$$

where ŷ is the predicted value, and J is the loss function.

4.3. Hardware and training

We have performed the hyper tuning of the parameters to determine the optimal setting of these. We have retrieved the hyper-parameters based upon the highest validation accuracy obtained through Grid -search experiments of our proposed model. Following were the parameters which are hyper tuned: bi-lstm size:{100,200,300}, dropout probability:{0.2,0.3,0.5}, batch_size:{16,32,64,128} and learning rate: {1e-2,1e-3,1e-4}.

The optimal parameters were: bi_lstm_size:200, dropout probability: 0.2, batch_size: 32, and learning rate: 1e-3.

We have used the 1xTesla K80 for training our model, which consists of 2496 CUDA cores and 12 GB GDDR5 VRAM. To avoid overfitting, the model is trained on 20 epochs that have been used. For the purpose of training, we have used 135,918 samples and 16,781 for testing purposes. Adam optimizer is used with the dropout of 0.2 in the final layer. Also, we have used Early Stopping and ReduceLROnPlateauas callbacks to avoid overfitting and get the optimum result while training. We used Adam optimizer as it adds the finest properties of the Adam and RMSProp algorithms to give an optimal algorithm that can hold sparse gradients on noisy problems.

5. Experimental evaluation

The supervised text classification technique has many problems in many areas. The main aim of this paper is to research which supervised AI strategies are most appropriate to explain it. As the new grievance comes, the model would dole out it to one of the 14 objection classes. The classifier makes the supposition that each new protest is relegated to one and only one classification. This is a multiclass content arrangement issue or more appropriately, a multiclass text classification problem. All of the classes are perfectly balanced, which is something we will almost never find in the wild. The text processing would be able to look at the most frequent words/bigrams. The processed text will also be what we use to create the features. The text will be tokenized, lowercase and lemmatized. It will have punctuation, numbers and stop words removed. The contractions will also be expanded out. We have sub-divided and categorized the text into using complaint_id: complaint, clean_text. We have utilized matplotlib. pyplot, each pyplot work rolls out some improvement to a figure: e.g., makes a figure, makes a plotting territory in a figure, plots a few lines in a plotting zone, brightens the plot with labels, and that is an assortment of order style works that make Matplotlib work like MATLAB and to know which complaint has got the maximum frequency and which one has the least frequency.

5.1. Evaluation criteria

Metrics are defined as a mechanism that helps us compare and measure the difference between the desired output and predicted output. The efficacy of the model is hence defined through different evaluation techniques. A model with better prediction returns a

higher metric score and vice-versa. Our task is mainly that of binary classification, so we shall be using standard metric scores such as Accuracy and F1 Score, these are the standard measures for evaluation of text prediction since our classes are balanced accuracy is appropriate to measure considering the True Positive and False Positives, whereas precision and recall are considered in F1-Score providing a balance evaluation for the model, these are defined as follows:

5.1.1. Accuracy

It is the proportion of the number of right predictions to the total number of input tests shown in Eq. (11).

$$Accuracy = \frac{True\ Positives + False\ Positives}{Total\ number\ of\ samples} \tag{11}$$

5.1.2. Precision

It is the quantity of right positive outcomes separated by the number of positive outcomes anticipated by the classifier mentioned in Eq. (12).

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{12}$$

5.1.3. Recall

It is the quantity of right positive outcomes isolated by the quantity of every single applicable example (all examples that ought to have been recognized as positive), discussed in Eq. (13).

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{13}$$

5.1.4. F1 score

The range for F1 Score is [0, 1], it reveals to you how exactly your classifier is (what number of occasions it arranges accurately), just as how powerful it will be (it doesn't miss a critical number of cases), it is the Harmonic Mean between Precision and recall. The more noteworthy the F1 Score, the better the exhibition of our model. High exactness yet lower review gives you a very precise, yet it at that point misses countless cases that are hard to characterize. Scientifically, it very well may be communicated in Eq. (14).

$$F1 = 2^{*}\frac{1}{\frac{1}{precision} + \frac{1}{recall}} \tag{14}$$

F1 Score attempts to discover the harmony between Precision and recall. We have only used the accuracy and F1 for comparison as they have been widely used for comparison in the majority of the methods. The recall and Precision have been calculated to indicate the performance of our model.

6. Results & analysis

For analysis of results, firstly, we have used the evaluation metrics discussed in section 6.1, Accuracy, Precision, Recall, and F1-score, as shown in Table 4. For, comparison we have used basically three other variations of the model such as Convolutional Neural Network (CNN) (Luan and Lin, 2019; Ce and Tie, 2020), Bidirectional LSTM (Liu and Guo, 2019; Bai, 2018; Jiang and Jin, 2017), a combination of CNN and LSTM and used Attention Mechanism (Liu and Guo, 2019; Sun and Lu, 2020; Kang, 2020)with FastText embeddings. Fig. 6 shows the accuracy curve of various models in respect of testing and training as well, where the x-axis and y-axis represent the accuracy and number of epochs, respectively.

It can be clearly seen from Table 4, representing testing accuracy, that our LSTM model obtains an accuracy of 89.5% while CNN + LSTM is performing better after LSTM. The Recall value of CNN + LSTM and Bi-directional is better than LSTM, while LSTM outperforms other models in Precision, having a value of 89 per cent. It can be clearly observed that with the use of Attention Mechanism on the input sequence, there is a jump in all metrics, as the attention mechanism helps the system to focus on specific words that are contributing to the final prediction and shows an accuracy of 90.81 and F1-Score of 0.92. Since in the above model we have Tf-Idf for the features, now to provide a character level embedding, we have created FastText Embedding on our dataset, which was provided as input to the Bi-LSTM Attention Model providing the highest performance with an accuracy of 93.25 and an F1-Score of 0.93. Fig. 7

Table 4
Comparative analysis of results of models.

Model Name	Accuracy (percentage)	Precision	Recall	F1-score
CNN	88.56	0.87	0.87	0.87
Bidirectional LSTM	88.3	0.88	0.89	0.88
CNN + LSTM	88.8	0.88	0.89	0.88
LSTM	89.5	0.89	0.88	0.88
Bi-LSTM -Attention	90.81	0.93	0.91	0.92
Bi-LSTM-Attention-FastText	93.25	0.93	0.94	0.93

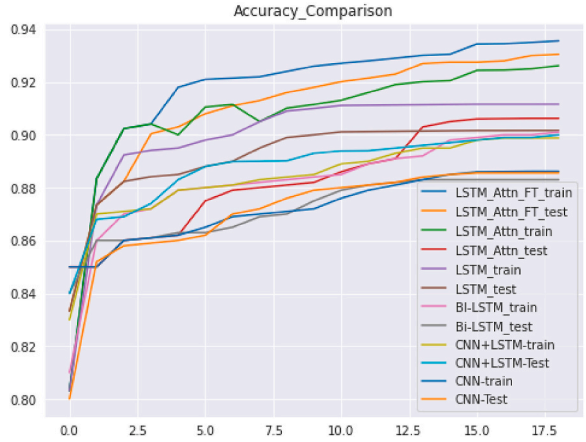

Fig. 6. Testing and training accuracy comparison curves.

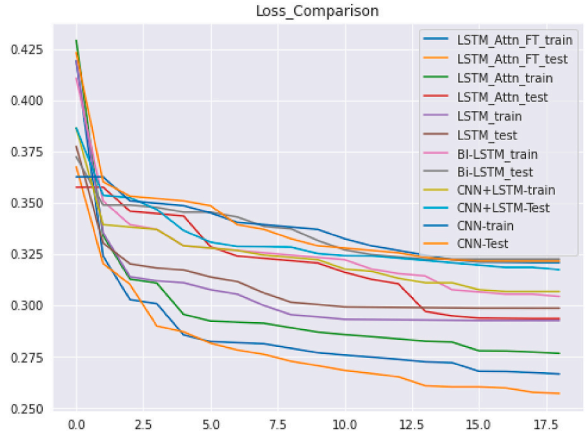

Fig. 7. Testing and training loss comparison curves.

shows Loss Curves in testing as well as training of all the compared models, where the x-axis and y-axis represent testing loss and number of epochs, respectively. Whereas Table 5 collects the Testing loss comparison from Fig. 7, and it reveals that LSTM has a Loss value of 0.31, which is the minimum among all, while the CNN model has the next lowest loss value that the model performs better than other neural network methods. Whereas, when compared with loss, LSTM and CNN models perform comparably the same, whereas the Bi-LSTM model and CNN-model have a higher loss. It is also observed that Bi-LSTM -Attention Model with or without FastText has achieved the loss of 0.26 and 0.29, respectively, which are one of the among all. This is due to the fact that attention mechanisms have been proven effective in highlighting the attention-worthy words during encoding which results in faster loss convergence.

In a nutshell, it is seen that the recall of Bi-LSTM-Attention with FastText embedding higher than the LSTM model also F1 Score is equal, whereas the LSTM model gives better results inaccuracy. Clearly, the LSTM model is achieving one of the lowest losses among other models. It can be perceived from the above graphs and Comparison table that Bi-LSTM with the introduction of FastText embedding and Attention mechanism is performing overall better than other compared model methods in all the evaluation metrics compared upon.

The confusion matrix of the Bi-LSTM Attention model having FastText embedding is shown in Fig. 8, where predictions of each class are mapped to the ground truth of that class since it is our best model. The vertical axis shows the True Label frequency, whereas the Horizontal axis shows the Predicted Labels. It was observed complaints related to 'Maintenance/Cleanliness' are sometimes misclassified into 'Malfunctioning of Electrical' Class. This could be due to similarity among the complaints of both the classes, as it has been observed that malfunctioning of electrical components at times related to maintenance category but has a significant difference as the latter one is a much broader category which includes various other services in the railways. Since the data is classified on the basis of Railway Departments making the class of electrical components to be created as a separate class.

But, overall, our proposed model performs well in distinguishing other classes. Fig. 9 shows the classification report of every class by the proposed model, where each class's prediction is evaluated on the basis of Precision, Recall and F1-Score. Here support represents the number of instances of a particular class present in the corpus.

Table 5
Comparative analysis of testing loss.

Model Name	Loss
CNN	0.31
Bidirectional LSTM	0.32
CNN + LSTM	0.33
LSTM	0.31
Bi-LSTM -Attention	0.29
Bi-LSTM-Attention-FastText	0.26

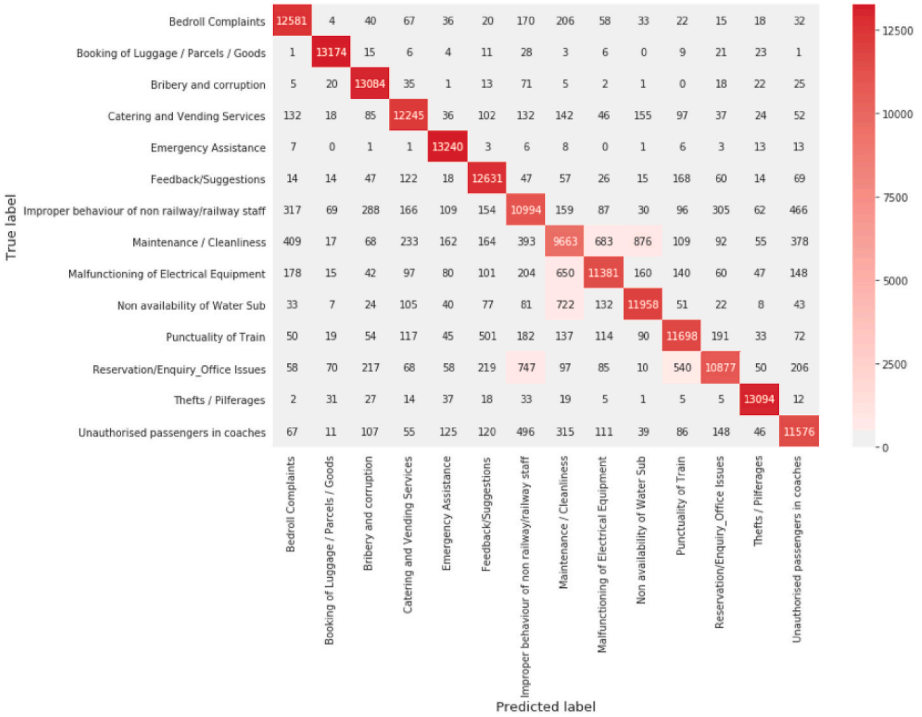

Fig. 8. Confusion matrix of proposed model.

	precision	recall	f1-score	support
Bedroll Complaints	0.91	0.95	0.93	13302
Booking of Luggage / Parcels / Goods	0.98	0.99	0.98	13302
Bribery and corruption	0.93	0.98	0.96	13302
Catering and Vending Services	0.92	0.92	0.92	13303
Emergency Assistance	0.95	1.00	0.97	13302
Feedback/Suggestions	0.89	0.95	0.92	13302
Improper behaviour of non railway/railway staff	0.81	0.83	0.82	13302
Maintenance / Cleanliness	0.79	0.73	0.76	13302
Malfunctioning of Electrical Equipment	0.89	0.86	0.87	13303
Non availability of Water Sub	0.89	0.90	0.90	13303
Punctuality of Train	0.90	0.88	0.89	13303
Reservation/Enquiry_Office Issues	0.92	0.82	0.86	13302
Thefts / Pilferages	0.97	0.98	0.98	13303
Unauthorised passengers in coaches	0.88	0.87	0.88	13302

Fig. 9. Classification metrics of proposed system.

7. Conclusion & future scope

The aim of this paper is to improve the pace of the complaint redressal system for railway passengers through the use of deep learning techniques. The model is trained on the complaint dataset of the COMS app, having 14 classes of complaint categories. Also, We have proposed an Attention-based Bi-Directional LSTM model trained on FastText Embedding for complaints text classification. Additionally, an analysis of the proposed system has been provided. This system acts as a feedback system that helps in the

improvement of railways services and providing structure to the complaint Redressal System of Railways. The other application of the proposed system could be in addressal of complaints received from mobile text messages, pushing the boundaries of the model's uses to non-smartphone users too. The model, on the whole, will classify the complaints into the respective department in Indian Railways. The proposed system can be used better assist the whole process of complaint redressal where the model may be deployed in parallel with the RailMadad system, which will then integrate itself with the National Train Enquiry System (NTES), which will directly send the user's complaint into the respective department or the service provider of the field unit responsible for providing the relief. We have identified further work in the model as prioritization, as an important task is to find out which complaint needs more attention. Once analysed, the actionable tweets are classified on the basis of priority (i.e., high, medium and low). Tweets that seek medical assistance, cleanliness or police help are kept in the critical or high-priority category. Others, such as those about broken windows, catering issues and missing parcels, would be categorized as medium and low priority. The handling of complaints can be outsourced to a trained team that works 24×7 in shifts. Thus, providing priority to every complaint can add itself into the complaint management system further for the betterment of it. A feedback mechanism can be established after each grievance redressal which will be a cloud-based form that will be collated with the RailMadad app for further analysis and improvement. Also, the system can be integrated with micro-blogging sites such as Twitter, where with the deployment of Twitter API, we can directly process such actionable tweets into the RailMadad system; this will require one-time permanent integration with Indian railways Twitter handles, thus providing real-time complaint handling.

Funding statement

The authors have not received any specific funding for this study. This pursuit is a part of their scholarly endeavours.

Declaration of competing interest

The authors declare that they have no conflicts of interest to report regarding the present study.

Acknowledgement

We want to thank CRIS (Centre for Railways Information Systems) for their support.

Appendix. Pseudocode of Proposed System

Algorithm 1: RailNeural

Data: 1. Embedding matrix $I_1 \in \mathbb{R}^{n \times k_1}$ where I_1 is obtained from Gensim FastText Embedding generation algorithm with weights $W_g \in \mathbb{R}^{|S| \times e}$

Data: 2. n is the number of input sequences, $k_1 \rightarrow$ sequence padding length, $S \rightarrow$ vocab size, e is embedding dimension.

Result: Complaint's label

$$L = \{ l_i \in \{0...13\} \mid i \in [1, n] \}$$

```
begin
    for epoch ←1 to N do
        Apply spatial dropout on I_1.
        Perform Bi-LSTM encoding on I_c matrix to transform into X matrix.
        Connect X with Attention mechanism to get A matrix.
        Flattening the encoded representation to add extra channel.
        Feed A to Sigmoid layer for labelling as final activation.
    end
    Update parameters using categorical cross entropy Adam optimizer.
end
```

References

Akhtar, N., Sufyan Beg, M.M., 2021. Railway Complaint Tweets Identification. , Data Management, Analytics and Innovation, pp. 195–207.

Bai, X., 2018. Text classification based on LSTM and attention. In: 2018 Thirteenth International Conference on Digital Information Management (ICDIM).

Bala, M.M., et al., 2018. Dynamic Behavior Analysis of Railway Passengers. , Innovative Applications of Big Data in the Railway Industry, pp. 157–182.

Benjamin, N., 2021. Problems of Indian Railways up to c.1900, the Railways in Colonial South Asia, pp. 307–331.

Ce, P., Tie, B., 2020. An analysis method for interpretability of CNN text classification model. Future Internet 12, 228.

Chatterjee, A.K., 2019. The Great Indian Railways: A Cultural Biography. Bloomsbury Publishing.

Chippagiri, S., 2021. Glory of Indian railways. J. Epidemiol. Community Health 75, 484.

Deng, X., et al., 2019. Feature selection for text classification: a review. Multimed. Tool. Appl. 78, 3797–3816.

Diyasa, I.G.S.M., et al., 2021. Twitter sentiment analysis as an evaluation and service base on Python textblob. IOP Conf. Ser. Mater. Sci. Eng. 1125, 012034.

Du, J., et al., 2021. Novel efficient RNN and LSTM-like architectures: recurrent and gated broad learning systems and their applications for text classification. IEEE Trans Cybern 51, 1586–1597.

Fellbaum, C. WordNet, 2015. Oxford Handbooks Online.

Garding, S., Bruns, A., 2015. Moving towards successful complaint management. SpringerBriefs in Business 13–26.

Gargiulo, F., et al., 2018. Deep convolution neural network for extreme multi-label text classification. In: Proceedings of the 11th International Joint Conference on Biomedical Engineering Systems and Technologies.

Gargiulo, F., et al., 2019. Deep neural network for hierarchical extreme multi-label text classification. Appl. Soft Comput. 79, 125–138.

Goyal, M., et al., 2020. Smart Government E-Services for Indian Railways Using Twitter. Micro-Electronics and Telecommunication Engineering, pp. 721–731.

HaCohen-Kerner, Y., et al., 2019. Automatic Classification of Complaint Letters According to Service Provider Categories, vol. 56. Information Processing & Management, p. 102102.

Hadifar, S. et al. Interruption of Signaling Pathways in Lung Epithelial Cell byMycobacterium Tuberculosis. .

Hazarika, D., et al., 2020. Sentiment analysis on twitter by using TextBlob for natural language processing. In: Proceedings of the International Conference on Research in Management & Technovation 2020.

Indian railways: problems and prospects. A study in the management and working of Indian railways. Int. Aff. 39, 1963, 314, 314.

Jiang, W., Jin, Z., 2017. Integrating bidirectional LSTM with inception for text classification. In: 2017 4th IAPR Asian Conference on Pattern Recognition (ACPR).
Jones, G.A., et al., 2005. Creating Knowledge, Strengthening Nations: the Changing Role of Higher Education. University of Toronto Press.
Kang, B., 2020. A convolutional neural network with word-level attention for text classification. In: Proceedings of the 12th International Conference on Computer Modeling and Simulation.
Kim, S.-W., Gil, J.-M., 2019. Research paper classification systems based on TF-IDF and LDA schemes. Human-centric Computing and Information Sciences 9.
Li, Z., et al., 2020. Social Sensing and Big Data Computing for Disaster Management. Routledge.
Liang, Q., et al., 2021. Artificial Intelligence in China: Proceedings of the International Conference on Artificial Intelligence in China. Springer.
Liu, G., Guo, J., 2019. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. Neurocomputing 337, 325–338.
Luan, Y., Lin, S., 2019. Research on text classification based on CNN and LSTM. In: 2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA).
Milde, B., Biemann, C., 2020. Improving Unsupervised Sparsespeech Acoustic Models with Categorical Reparameterization. Interspeech 2020.
Naive bayesian automatic classification of railway service complaint text based on eigenvalue extraction. Tehnicki vjesnik - Technical Gazette 26, 2019.
Namburu, S.M., et al., 2005. Experiments on supervised learning algorithms for text categorization. In: 2005 IEEE Aerospace Conference.
Nowak, J., et al., 2017. LSTM Recurrent Neural Networks for Short Text and Sentiment Classification. Artificial Intelligence and Soft Computing, pp. 553–562.
Poria, S., et al., 2018. Multimodal Sentiment Analysis. Springer.
Srinivasa-Desikan, B., 2018. Natural Language Processing and Computational Linguistics: A Practical Guide to Text Analysis with Python, Gensim, spaCy, and Keras. Packt Publishing Ltd.
Stauss, B., Seidel, W., 2019. Organizational aspects of complaint management. Management for Professionals 391–429.
Stein, R.A., et al., 2019. An analysis of hierarchical text classification using word embeddings. Inf. Sci. 471, 216–232.
Sun, X., Lu, W., 2020. Understanding attention for text classification. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.
Teng, J., et al., 2018. The Method of Analyzing Metro Complaint Data and its Application. CICTP, p. 2017.
Tong, X., et al., 2018. A complaint text classification model based on character-level convolutional network. In: 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS).
Vel, S.S., Sakthi Vel, S., 2021. Pre-processing techniques of text mining using computational linguistics and Python libraries. In: 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS).
Yao, T., et al., 2020. Text classification model based on fastText. In: 2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS).