



Weighted Ensemble Learning for Accident Severity Classification Using Social Media Data

Sanjib Kumar Raul¹ · Rashmi Ranjan Rout¹ · D. V. L. N. Somayajulu^{1,2}

Received: 1 August 2023 / Accepted: 4 April 2024

© The Author(s), under exclusive licence to Springer Nature Singapore Pte Ltd. 2024

Abstract

Due to the heavy increase in social media usage, like Twitter, there is a growing interest in the research community in developing automation tools like accident severity-based tweet classification models. These tools aid in automatically extracting severity information from the accident tweet content. Moreover, prediction models are essential for predicting the severity of an accident to increase the safety efficacy of the road traffic system. However, the difficulty lies in the collection of sufficient labeled data. We propose a model called weighted ensemble-based self-training with decision tree (WESTDT), a semi-supervised methodology with a dynamic data labeling strategy. The base classifier in this model is a homogeneous weighted ensemble of decision tree classifiers for better prediction of pseudo labels. We also propose a novel performance measure called risk factor to estimate the amount of risk present in the application using the prediction model. The proposed model outperformed the state-of-the-art model, namely reliable semi-supervised ensemble learning (RESSEL), and the baseline models, namely decision tree (DT) and self-training with decision tree (STDT), in terms of both the traditional and the proposed measures. The results indicate that the proposed framework outperforms all other models on all datasets in terms of precision, recall, and accuracy by a range of 5–18.3%, 3.9–9.3%, and 6.6%, respectively. These findings can be helpful for the development of efficient and sustainable systems for traffic management and safety. It is also crucial for assisting government authorities in devising prompt, proactive strategies to prevent traffic accidents and enhance road safety.

Keywords Self-training · Semi-supervised learning · Weighted ensemble learning · Decision tree · Accident severity classification

Introduction

Road accident injuries are the most common cause of injury, and the eighth most common cause of mortality reported globally [1, 2]. Around the world, traffic accidents involving pedestrians, buses, trucks, cars, motorbikes, and bicycles result in more than 3700 fatalities per day in the year 2020 [1, 3, 4]. One of the main challenges in developing smart cities and transportation networks is the avoidance of accidents. Human mistake is the main cause of collisions and road accidents. The cost of injuries and fatalities from road accidents is rising daily, having an adverse impact on society [5]. Accidents are happening due to various factors like driving speed, road conditions, collisions, wrong driving, vehicle conditions, and others.

The massive growth in vehicular traffic has led to an increase in accidents causing fatality and injuries [6]. Traffic authorities across the globe are investing huge economic resources and human efforts in handling traffic to minimize

This article is part of the topical collection “Dependable Cyber-Physical Systems and Cyber Security” guest edited by Deepak Puthal and Niranjana K. Ray.

✉ Sanjib Kumar Raul
sanjib.roul@student.nitw.ac.in

Rashmi Ranjan Rout
rashrr@nitw.ac.in

D. V. L. N. Somayajulu
soma@nitw.ac.in

¹ Department of Computer Science and Engineering, National Institute of Technology, Warangal, Telangana 506004, India

² Department of Computer Science and Engineering, Indian Institute of Information Technology Design and Manufacturing (IIITDM), Kurnool, Andhra Pradesh 518002, India

accidents for enabling smooth transportation. Traditional methods for analyzing causes of accidents utilize the information collected from cameras, sensors, loop detectors, etc [7]. However, the data collection methods in all these approaches are unreliable due to the untrusted networks and other factors of infrastructure [8]. Moreover, these methods are not cost-effective, making them infeasible to scale for larger applications. In addition, accident severity prediction approaches exist based on the conventional data from records in various traffic departments [9, 10]. Despite the reliability of data, these methods have limited scope due to restricted access, high cost, and also prone to human errors in recording the data [9, 11]. To mitigate these challenges in data collection, online social media like Twitter, given its large-scale usage and popularity, is a lucrative option for collecting information related to road accidents [12]. Hence, social media data from microblogging sites like Twitter is increasingly considered a readily available and huge source of information compared to physical sensors.

Twitter has emerged as a popular social media microblogging platform, with millions of people uploading vast amounts of information about various subjects [13]. The information shared in the tweets also includes topics like road accident severity; hence, tweets provide users with early information related to accidents and warnings in accident-prone zones [14, 15]. In earlier literature, techniques exist that develop various supervised models to detect the severity of accidents from tweets based on traditional machine learning (ML) and deep learning approaches. However, the works rely on supervised machine learning algorithms such as Decision Tree (DT), Naïve Bayes (NB), Multilayer Perceptron (MLP), etc. [16] and therefore have a constraint of requiring larger quantities of labeled data, making it infeasible for large-scale applications [17]. Given the scarcity of labeled tweets, there exists an opportunity for utilizing semi-supervised approaches in accident severity prediction from Twitter data. Since there can be different severity categories in the observed accident tweets information, we choose the types such as no-severity, minor, major, and fatal.

In this work, we developed a dynamic labeling framework called WESTDT, a semi-supervised learning solution for accident tweet severity classification. The proposed solution also acts as a classification solution as a supervised learning model that will be trained in the self-training phase by utilizing a combined set of pseudo-class labels and the original set of labeled accident severity tweets. The proposed approach utilizes the semi-supervised self-training learning model with the customized weighted ensemble of decision tree classifiers. The self-training model first utilizes the available labeled tweet information to train a group of decision tree classifiers. Then, the unlabelled tweet data is fed to the trained decision tree classifiers. Each trained DT classifier predicts a severity label along with the confidence value, and pseudo-labeled tweets that record a confidence threshold of at least 0.9 are included in the final pseudo-labeled data of that particular iteration. Now, final predicted value and final confidence value are obtained using the weighted combination of diverse classifiers. Using the pre-defined threshold value, the pseudo-severity labeled information with the highest confidence will be taken out from the unlabelled set and then added to the labeled set. This process will be repeated until no more unlabelled tweets or confident pseudo-labels are generated. The implications of the severity of accident tweet classification are early alerts to possible security risks. To enhance decision-making and establish management measures to reduce accidents, those accidents, specifically their severity, should be detected effectively.

In addition to the proposed framework, we also included an important measure called the risk factor (RF) which is derived based on the measure called mean absolute error (MAE) [18]. The risk factor is defined for each severity category (except no-severity), where the amount of risk associated with the application, if any high-severe information from a tweet, is classified into its low-level severity levels.

For the empirical evaluation of the proposed solution, we compared it against three models. The first is a traditional DT with fully labeled severity tweets. The second is trained based on a self-training strategy of semi-supervised learning

using a single decision-tree classifier. Thirdly, Sjoerd et al. [19] have recently introduced RESSEL in their work. The empirical evaluation is carried out on four datasets of different-sized tweets information. According to the analysis of variance (ANOVA) results, the proposed WESTDT model considerably outperformed the other three models on most datasets. The results of this study are expected to be helpful in the development or improvement of a successful system for traffic safety that falls under the umbrella of sustainable transportation systems. This is essential for aiding concerned government officials to avoid traffic accidents and improve road traffic safety effectively.

To our knowledge, the following contributions to the field of accident severity prediction study are made by this work.

1. We have formulated a weighted ensemble of self-training semi-supervised learning approaches for the dynamic labeling of the severity of road accident tweets.
2. A large-scale empirical analysis on the various sets of labeled tweets has been conducted to show the efficacy of the proposed approach.
3. An extensive comparative analysis has been conducted using various baseline classifiers to show the significant impact of using the proposed approach for accident severity tweet classification.
4. We have proposed an important performance measure called the risk factor to show the amount of risk i.e., still present in the end application, with the use of the prediction model.

The remainder of this paper is organized as follows. Several important pieces of literature on social media-based road severity accidents are discussed in “[Related Work](#)” section. The proposed framework is explained in “[The Weighted Ensemble-Based Self-Training with Decision Tree \(WESTDT\)](#)” section. The experimental setup and results are discussed in “[Experimental Setup](#)” section and in “[Results and Discussion](#)” section, respectively. Potential threats to validity of the proposed model are explained in “[Threats to Validity](#)” section. “[Conclusion and Future Work](#)” section concludes the work and shows some potential future directions from this work.

Related Work

This section reviews several important pieces of literature on social media-based road severity accidents. In “[Approaches for Accident Severity Classification Based on Conventional Data](#)” section, we present an accident-severity classification

based on conventional data. In “[Tweets-Based Approaches for Accident Severity Classification](#)” section, we present the detailed literature on the tweet-based classification approaches for the accident severity category. “[Ensemble Techniques in Related Domain](#)” section explained ensemble techniques in the related domain, and in “[Self-Training Semi-supervised Learning](#)” section, we discuss a few recent works that utilize the self-training semi-supervised learning approach for various problem contexts. Finally, in “[Text Vectorization Techniques](#)” section, we discuss the text vectorization techniques used in various models.

Approaches for Accident Severity Classification Based on Conventional Data

In [9], Sameen et al. developed a deep learning model for severity prediction of injuries based on recurrent neural network (RNN) from 1130 traffic accident records. The PLUS accident databases were used to get the data. Files related to accident frequency and severity, which are in Excel format, were used in this work. Assi et al. [11] developed an ML model for predicting crash injury severity as severe and non-severe using four ML models such as feed-forward neural networks (FNN), support vector machines (SVM), etc. The Great Britain crash database was used in this work. A deep forests-based approach for predicting the severity of traffic accidents is proposed in [10], using the UK road safety dataset. The categories considered in the severity of traffic accidents are: slight, serious, and fatal. Jhang et al. [20] developed a framework using a deep learning method with a convolutional neural network called Traffic Accident Severity Prediction- CNN (TASP-CNN). The authors of [20] implement TASP-CNN using 21,436 accident data records from Leeds City Council in the United Kingdom. Also, the different degrees of severity of accidents are categorized as slight, serious, and fatal.

Tweets-Based Approaches for Accident Severity Classification

The author of [21], developed a prediction model for traffic accidents utilizing deep learning techniques. In this work, they included sentiment and emotions from the tweets’ text for the detection and classification of traffic accident posts. The accident severity is classified as low, mid, and severe. In [17], the researchers discussed a framework for detecting traffic accident severity using Twitter data. The framework utilizes traditional and deep learning models like Naive Bayes (NB), support vector machines (SVM),

and bidirectional encoder representations from transformers (BERT). SVM is a supervised machine learning method mostly employed for classification tasks. It is favored for its high classification accuracy, capability to handle non-linear data, and lower susceptibility to overfitting than other classification approaches [22, 23]. Also, the NB model is used in classification tasks because of its simplicity in application and satisfactory results [24]. Google's BERT pre-trained model was fine-tuned to classify data by including annotated examples from a collection of tweets [25]. The severity of accidents is classified as severe, non-severe, and non-accident.

Ensemble Techniques in Related Domain

The authors of [26] have analyzed the comparative performance of various ensemble techniques on the traffic data for predicting road traffic congestion, using the data collected from Gauteng Province (GP) road traffic flow. The traffic congestion prediction using the combination of decision trees with ensemble techniques resulted in high performance. This work does not use weighted aggregation for the ensemble. In [27], the researchers employ ensemble techniques like extreme gradient boosting (XGBoost) and random forest (RF) for predicting the rural road crash frequency and developing a crash risk map. In addition, an analysis of the traffic crashes and influencing factors on rural roads are performed. In the bagging approach of the ensemble, RF does not use weighted aggregation for the ensemble. In [28], Jamal et al. have developed an ensembled model for predicting the severity of crash injury. The authors used the XGBoost approach, a gradient-boosted decision tree-based integrated ensemble machine learning model.

Self-Training Semi-supervised Learning

The self training mechanism of semi supervised learning is successful in many applications, such as software defect severity prediction [29], class balancing task [30], face recognition [31], traffic sign recognition [32], automatic speech recognition [33], image transformations [34], etc. Our work is highly motivated by the work of Sharma et al. [29]. In [29], Sharma et al. proposed a semi-supervised model using self-training to predict the software defect severity from the available defect information. Their model first builds a supervised learning model by utilizing the available labeled information. Later, most of the unlabeled defective modules will eventually be given pseudo-labels over a period of iterations. From the empirical analysis of the publicly

available AEEEM datasets, the self-training model achieved an improved performance on the combined set of labeled and pseudo-labeled defective instances. In [35], Liu et al. have discussed a self-training semi-supervised feature-weighted clustering method for classification and prediction confidence using UCI datasets. The authors suggested a decision tree node split approach that clusters instances based on weighted features. The authors of [19] proposed an ensemble model called RESSEL for making semi-supervised learning methods applicable without a need to understand the complex structure of problems. It creates a wrapper around the base classifier, and the unlabeled data is used to generate many diverse classifiers, which are used as an ensemble for final label prediction.

Text Vectorization Techniques

This section covers the techniques that are used to convert text data into feature vectors. Since most machine learning models are trained on numerical data, it is essential to vectorize the text data into numerical form.

A tweet is composed of a collection of words, where features are represented by words. *one-hot vector* is a feature encoding method, and each vector in this encoding is a representation of a word. Here, only the position of the word's occurrence is assigned a value of 1, while the remaining positions are assigned 0. For instance, if vocabulary contained four words, like (student, teacher, player, and coach), then 0100 is the one-hot vector representation for the word "teacher" [36]. A skip-gram and continuous bag of words (CBOW) word embedding model has been proposed by Mikolov et al. [37] to train millions of words to produce precise word representations. Word embeddings are capable of handling syntactic and semantic relationships as well as identifying language regularities and patterns. Furthermore, the authors of the work discussed, how the dimension of the word vector and the volume of training data affect training time and accuracy. Pennington et al. [38] suggested the Glove word embedding approach, which uses word analogies and similarity to perform different text processing tasks. In [39], Johnson et al. suggested a CNN text classification model that accepts input represented using one-hot encoding. The fields of biological sciences are also increasingly employing word embedding techniques. Ghosh et al. [40] presented a word2vec-based Dis2Vec model for categorizing illnesses like endemic, emergent, etc.

In [41], Raul et al. proposed a RVCNN-DBPSO model which uses the embedding technique of fastText to represent tweet vectors. FastText offers significant benefits compared to traditional word2vec embeddings [42]. We use the

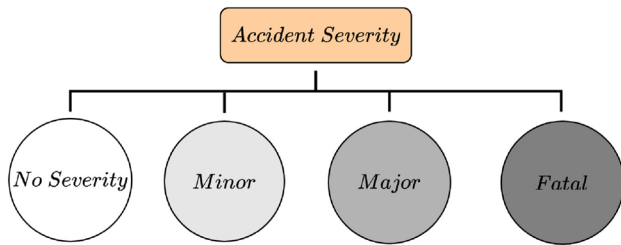


Fig. 1 Road accident severity classification

fastText technique in our suggested accident tweet severity categorization strategy due to its benefits. “[The Text Vectorization Phase](#)” section gives a detailed description of this technique.

From the literature review, it is noted that our proposed work exhibits the following differences from the existing works:

- Most of the models in the literature are based on supervised learning methods, which require a huge amount of labeled data, making them constrained and hence, not useful for large-scale applications.
- Within the semi-supervised self-training approaches, the base learners may not produce accurate predictions due to insufficient data. In this regard, the literature shows

that the ensemble learning models are efficient even if there is an availability of limited labeled data. Hence, in addition to proposing a self-training semi-supervised learning model, we also incorporate a weighted ensemble learning mechanism in the self-training phase to observe more accurate pseudo-labels.

- Most of the existing models in the literature focus on proposing prediction models or mitigating the constraints that limit the prediction performance in accident tweet severity classification, whereas in this work, in addition to proposing self-training semi-supervised learning, we also try to interpret the predictions in terms of risk factor, as this was not proposed in the literature.

The Weighted Ensemble-Based Self-Training with Decision Tree (WESTDT)

In this section, we present a detailed methodology for the dynamic labeling strategy using a decision tree (DT)-based self training semi supervised approach. We also refer to this approach as an accident tweet severity classification (ATSC) since, after labeling the unlabelled instances, we first train a model (for example, DT model) on the obtained pseudo-labeled tweet instances. Then,

Table 1 Table of notations

Symbol	Description
n	Total number of labeled tweets
t_i	i th tweet vector
y_i	i th severity label corresponding to tweet t_i
$\mathcal{L}$	The set of labeled tweets, $\mathcal{L} = \{(t_i, y_i)_{i=1}^n\}$
$\tilde{t}_i$	i th unlabeled tweet
u	Total number of unlabeled tweets
$\mathcal{M}$	Set of unlabeled tweets, $\mathcal{M} = \{\tilde{t}_i\}_1^u$
T	Number of samples of the training data, sampled with replacement
$\mathcal{L}_k$	k th sample drawn from the labeled dataset $\mathcal{L}$
$\mathcal{A}$	Decision Tree Classifier
$\mathcal{A} \leftarrow \mathcal{L}_k$	Decision Tree Classifier of k th sample data, $\mathcal{L}_k$
j	Any j th class label, $j \in \{\text{no-severity, fatal, major, minor}\}$
n_{kj}	Number of Accident tweet instances of severity type j trained on sample $\mathcal{L}_k$
y_t^j	The class label predicted by j th classifier for the tweet t
Y_t^k	The set of class labels predicted by classifiers, for tweet t , that are same as the class label predicted by k th classifier, y_t^k , for the same tweet
$Prop_k(t)$	Proportion of the classifiers from a total T classifiers that produces similar class-label
$Conf_k(t)$	Confidence value of label predicted using classifier k for tweet t
N	Total number of accident-tweet instances at leaf nodes of the decision tree
Γ	The threshold value of the confidence
τ	Iterative count, an hyperparameter
f_τ	Probability estimate for the pseudo-labeled accident tweet instances

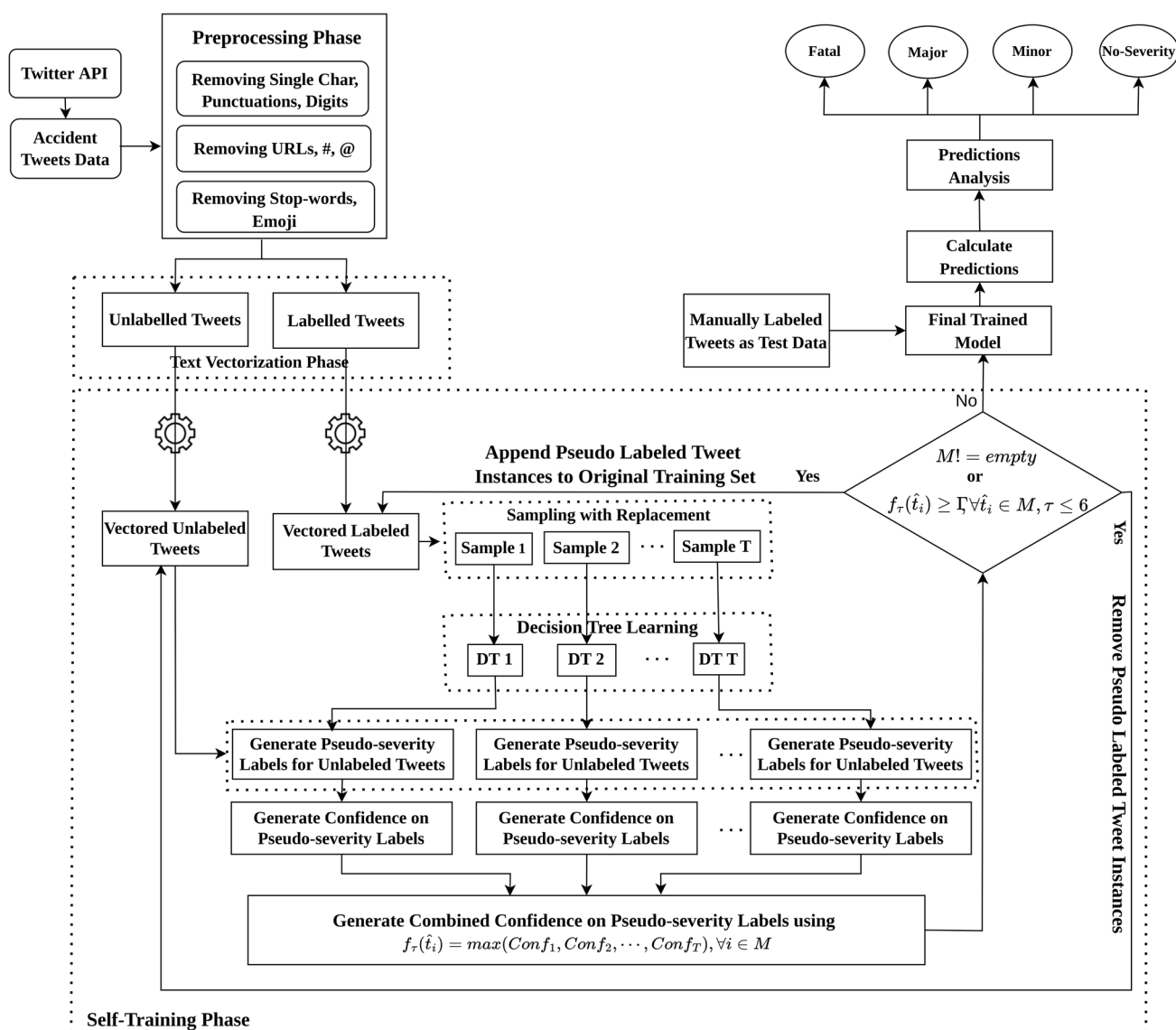


Fig. 2 Proposed self-training semi-supervised learning-based accident tweet severity classification

we perform predictions for the manually-labeled accident tweet instances using the trained accident tweet severity classification model. Table 1 contains symbols and notations used in this work.

As discussed in “Introduction” section, the proposed strategy is implemented for assigning the four severity labels such as *no-severity*, *minor*, *major*, and *fatal* to the accident tweet instances (it is depicted in Fig. 1). The proposed solution for the dynamic labeling strategy for the accident tweet severity instances using a DT-based self-training semi-supervised learning model is depicted in Fig. 2.

The process of labeling tweets using the DT-based self training semi supervised approach is an iterative approach and is successfully implemented at various tasks such as software defect severity prediction [29], class balancing task [30], face recognition [31], traffic sign recognition [32], automatic speech recognition [33], image transformations [34], etc. The proposed framework is divided into three phases: pre-processing, training, and testing. As discussed before, we first split the accident tweets data into two parts such as labeled and unlabeled. For labeling the tweets, we first crawled Twitter and collected around 30,000 tweets that were related to the accident tasks. Then, using a group of expert’s decisions, we label 2000 tweets into four severity categories such as *no-severity*, *minor*, *major*, and *fatal*.

The Pre-processing Phase

The tweet may include hashtags, mentions, white spaces, numerals, punctuation, emojis, and various special characters. By using regular expressions, such undesirable characters should be removed from tweets [43]. Initially, all the characters of the tweets are changed to lowercase. Afterward, hashtag hashes (#) and mentions (@) are eliminated. We remove Unicode characters like `\u002b` and `\x07` and non-English characters from the tweet [43]. The tweet can use slang and abbreviations. Therefore, it is important to understand its meaning. A word or phrase is abbreviated to make it shorter, but slang is understood as a sort of language comprising of phrases and words that are considered to be highly informative and are frequently limited to a particular context or a group of people. With such a set of 290 phrases and words, and their substitutes, a lookup table was manually generated [44]. The uniform resource locator (URL), emoticons, digits, and punctuation are eliminated. Stopwords, beneficial words that are regularly used, can be found in tweets [45]. However, they carry no valuable information for tweet categorization and hence, all the stopwords like—‘a’, ‘are’, ‘the’, ‘is’, etc are eliminated from tweets. In our implementation, we use the NLTK standard stopwords library [46]. NLTK stands for Natural Language Tool Kit, an open-source library for natural language processing tasks. NLTK has a predefined list of stopwords for around 22 languages, which helps remove the stopwords from the tweet preprocessing step [47]. In this work, we have used the NLTK of version 3.8.1. NLTK is employed for the pre-processing tasks. The parsing of structured tweets is performed using column specifications. Finally, the training

data is organized in two columns: first, the “Text” of the tweet, and second, the category “label” for the tweet severity classification.

The Text Vectorization Phase

In this work, we employ the pretrained embedding technique of fastText to vectorize the tweet data [48]. FastText extends the Word2Vec embedding by learning the n-gram representations for characters, with words being expressed as sums of n-gram vectors [49]. The fastText model is a quick and effective method for learning word representations for categorizing tweets [42]. In this paradigm, each word is represented by a group of characters called n-gram. A vector is used to represent each n-gram character and a word is defined by the sum of the n-grams. As an illustration, character n-grams can be used to represent the word “fatal” with $n = 3$ as (fa, fat, ata, tal, al) [42]. The reason for selecting the fastText word embedding approach is that it is more accurate than the word2vec and glove model techniques. FastText offers two key benefits compared to traditional word2vec embeddings. Word2vec is unable to give word embedding for words that are not in the vocabulary but can be resolved by fastText. By using a unique vector representation for each word, the Word2Vec method ignores the underlying structure of words. For instance, the terms bank and banking are used in many contexts, and we cannot identify the underlying commonality. Moreover, fastText provides word vectors learned on Wikipedia for 157 different languages. In our study, we used the wiki-news-300d-1 M fastText pre-trained word embedding to represent the tweets.

Algorithm 1 A weighted ensemble of decision tree-based self-training semisupervised learning for accident tweet severity classification

The iterative model uses the labeled accident tweets data to

INPUT: $\mathcal{L} = \{(t_i, y_i)_{i=1}^n\}$: Collection of labelled accident tweet severity data,
 $\mathcal{M} = \{(\tilde{t}_i)_{i=1}^u\}$: Collection of unlabelled tweets data,
 $\mathcal{A}$: Decision Tree Classifier,
OUTPUT: Trained Model, f .

```

1: function  $\mathcal{A}(\mathcal{L})$ 
2:    $\tau = 1$ 
3:   while ( $\mathcal{M} \neq \text{Empty}$  or  $\tau \leq 6$ ) do
4:     for  $k \in \{1, 2, 3, \dots, T\}$  do
5:       Draw a sample  $\mathcal{L}_k$  from  $\mathcal{L}$  with replacement
6:       Train the new classifier  $f_{k\tau} = \mathcal{A}(\mathcal{L}_k)$ 
7:       Find the predictions for each unlabelled tweets  $\mathcal{M}$  using  $f_{k\tau}$ 
8:       Estimate the value of  $n_{kj}$  as the number of accident tweet
       instances of a severity type  $j$  in  $N$  unlabelled tweets  $\mathcal{M}$ 
9:       Assign the confidence value for the pseudo-label of  $t_i$  from the
       classifier  $f_{k\tau}$  using  $\frac{n_{kj}}{N}$ 
10:    end for
11:    for each  $t_i \in \mathcal{M}$  do
12:      Find the proportion ( $Prop_k$ ) of the classifiers (from the total of
       $T$  classifiers) which produce a similar class label for each unlabelled tweet
13:      for  $k \in \{1, 2, 3, \dots, T\}$  do
14:        Find the final confidence as the weighted proportion for the
        prediction of a classifier  $k$  on tweet  $t$  is computed using:  $Conf_k(t) =$ 
 $\sum_{k=1}^T Prop_k(t) \times f_{k\tau}(t)$ 
15:      end for
16:      Determine the confidence for the pseudo-label of  $t_i$  using
       $\max(Conf_1, Conf_2, \dots, Conf_T)$ 
17:    end for
18:    Find out the subset  $\mathcal{M}_\tau = (\tilde{t}_i, \tilde{y}_i)$  which satisfy  $|f_\tau(\tilde{t}_i)| \geq \Gamma$ 
19:    Update  $\mathcal{M} = \mathcal{M} - \mathcal{M}_\tau$ 
20:    Update  $\mathcal{L} = \mathcal{L} \cup \mathcal{M}_\tau$ 
21:     $\tau += 1$ 
22:    for each  $f_\tau(\tilde{t}_i) \in f_\tau$  do
23:      if  $f_\tau(\tilde{t}_i) < \Gamma$  then
24:        Terminate the While loop
25:      end if
26:    end for
27:  end while
28:  Return the final classification model
29:  return  $f$ 
30: end function
31: Get the predictions for the unseen accident tweet instances using the
    trained model  $f$ 

```

The Training and Classification Phase

The algorithmic procedure for the accident tweet severity classification (ATSC) task is presented in Algorithm 1.

build an initial model during the training phase. Formally, let us have a labeled accident tweets data $\mathcal{L} = \{(t_i, y_i)_{i=1}^n\}$, where the set $\mathcal{L}$ contains n number of labeled accident tweet instances and, y_i is the corresponding severity class

Table 2 Description of the utilized labeled and unlabeled tweet datasets

Dataset	Sample size (training: 80% testing: 20%)	Unlabelled tweets	Total tweets
D1	500 (400, 100)	2500	3000
D2	1000 (800, 200)	5000	6000
D3	1500 (1200, 300)	7500	9000
D4	2000 (1600, 400)	10,000	12,000

Table 3 Distribution of labeled tweets in the datasets

Dataset	Fatal	Major	Minor	No-severity	Total tweets
D1	125	125	125	125	500
D2	250	250	250	250	1000
D3	375	375	375	375	1500
D4	500	500	500	500	2000

label for each accident tweet. Similarly, let us have an unlabelled accident tweets data, represented as $\mathcal{M} = \{\tilde{t}_i\}_{i=1}^u$, where the set $\mathcal{M}$ contains u number of unlabelled accident tweet instances.

Now, by utilizing the labeled accident tweets dataset, $\mathcal{L}$, sampled sets of labeled data, unlabelled accident tweets dataset, $\mathcal{M}$, and some threshold Γ , the procedure of self-training will iterate over the following two steps:

Step 1: Generating Pseudo-Labels

From the vectored labeled accident severity-tweets data, we sample the dataset into T sets with the replacement, and each sample has an equal number of tweets of a size equivalent to the original set of tweets. Now, we use these samples to train T different decision tree classifiers.

In the first iteration, we build an initial classification model (using the decision tree model, $\mathcal{A}$) by utilizing T sets of the labeled accident severity data, $\mathcal{L}$. The class labels for the unlabelled tweets will then be found using the trained T classifiers.

Now, from each trained classifier, we define the confidence for each predicted class label of the accident tweet instance based on the technique called absolute class frequencies [50]. The confidence value for the predicted class label is derived based on the majority of the severity class labels at the leaf node of the decision tree classifier [29, 50]. Formally, let n_{kj} represent the number of accident tweet instances of a severity type j over the total number of accident tweet instances N at the leaf nodes of the k th trained

Table 4 Crawled Twitter sample data

Processed tweets	Accident-category	Accident-severity
Today three persons killed bus road accident Chenani Bashy district Udhampur sympathies deceased injured persons may god bless early recovery injured may almighty bless peace deceased government requested help victims	Bus	Death
Hindu Puranik family gets heavily injured car accident muslim Sammer Bagsiraj comes rescue	Car	Major
Had a whole trip for Disneyland planned but now I am minor injured from a car accident crash and broke with a paid off trip I have no way of going on	Car	Minor
Saturday Night Live alum Pete Davidson was recently involved in a car accident in Beverly Hills police confirmed to E News but no one was injured	Car	No-severity

decision tree model. The absolute class frequencies (also called the *confidence value*) for each predicted value of tweet instance is derived as:

$$f_{k\tau}(\tilde{t}) = \frac{n_{kj}}{N} \quad (1)$$

Equation 1 is referred to as the probability estimate for the pseudo-labeled accident tweet instance $\tilde{t}$. As mentioned above, we eliminate the low confident (derived based on the absolute class frequency) tweet instances based on the predefined threshold value, Γ . Now, in the following procedure, we calculate the final confidence on the predicted class label.

Since each trained classifier may produce different class labels for the unlabelled tweet, we followed the weighted absolute class frequencies to find the final confidence on the predicted class label. Formally, for each tweet in a set $\mathcal{M}$, we first find the proportion (denoted as $Prop_k$) of the classifiers (from a total of T classifiers) that produce a similar class label. The final weighted confidence for the prediction of a classifier k on tweet t is given as follows:

$$Conf_k(t) = \sum_{k=1}^T Prop_k(t) \times f_{k\tau}(t) \quad (2)$$

Where, the proportion (denoted as $Prop_k$) of the classifiers that produce a similar class label as that of classifier k for a given tweet t , is computed as follows:

$$Prop_k(t) = \frac{1 + |Y_t^k|}{T}, \quad \text{where,} \quad (3)$$

$$Y_t^k = \{y_t^j | y_t^j = y_t^k, i \neq k \quad \forall j \in 1, 2, \dots, T\}$$

Here, Y_t^k is the set of class labels predicted by classifiers that produce a similar class label as that of classifier k for tweet t . Now, among the obtained weighted confidence values for each severity type, we consider the class label with the highest weighted confidence to the predicted tweet. This is given as:

$$f_{\tau}(\tilde{t}) = \max(Conf_1, Conf_2, \dots, Conf_T) \quad (4)$$

The τ is a hyperparameter, and hence, its value is decided based on the cross-validation. On the original set of labeled instances, we find the hyperparameter values of the self-training model and the best performance at the value 6.

Using these confidence values, we eliminate the low-confident pseudo-labels using the threshold value Γ . In this work, the confidence threshold value Γ is considered 0.9. This makes the proposed WESTDT algorithm include only those pseudo-labeled tweets in set $\mathcal{M}_{\tau}$ for which the confidence value, computed using Eq. 4, is greater than or equal to the threshold of 0.9. Formally, from the set of unlabelled accident tweets data $\mathcal{M}$, we first group the subset of accident

tweets $\mathcal{M}_{\tau} = (\tilde{t}_i, \tilde{y}_i), |\mathcal{M}_{\tau}| < u$, in the present iteration using the criteria that satisfy $f_{\tau}(\tilde{t}_i) \geq \Gamma$.

Step 2: Refining the Classifier

In the previous step, we have derived a set $\mathcal{M}_{\tau}$ that contains the high-confident pseudo-labeled accident tweet instances in its current iterate. Now, in this step, we append the pseudo-labeled accident tweet instances set $\mathcal{M}_{\tau}$ to the original set of labeled accident tweet instances $\mathcal{L}$. Analogously, we remove the information of the tweets $\mathcal{M}_{\tau}$ from the unlabelled tweet set $\mathcal{M}$.

We repeat step 1 multiple times till either of the sets such as the unlabelled tweets set, $\mathcal{M}$, or the pseudo-labeled tweets set $\mathcal{M}_{\tau}$ is empty.

Experimental Setup

The experiments were conducted on a Lenovo ThinkStation p720 with an Intel Xeon processor running at 2.20 GHz with 40 cores. The operating system was 64-bit Ubuntu 20.04.1, and the machine had 64 GB of RAM. The works were implemented on the spyder-4.2.1 Integrated Development Environment (IDE) with python-3.8 programming language. In the following subsections, we provide the details of the utilized datasets (in “[Utilised Datasets](#)” section), the methods used for comparative analysis (in “[Methods Used for Comparative Analysis](#)” section), the performance measures (in “[Performance Measures](#)” section), and the significance test (in “[Statistical Significance Test](#)” section).

Utilised Datasets

A web crawler is used to retrieve tweets about accidents. It uses the freely available *tweepy* python library. To extract the tweets from the Twitter server, Twitter API calls were used using Twitter developer account credentials. We utilize the representational state transfer application programming interface (REST API) [51] in this work since it allows for creating search queries to get tweets that include the required accident keywords. *Since, until, lang* and other parameters are utilized during the crawling process. Moreover, it’s

Table 5 Accident Severity Confusion Matrix

Actual	Predicted				
	Classes	Fatal	Major	Minor	No-severity
Fatal		$TP_{1,1}$	$FN_{1,2}$	$FN_{1,3}$	$FN_{1,4}$
Major		$FP_{2,1}$	$TP_{2,2}$	$FN_{2,3}$	$FN_{2,4}$
Minor		$FP_{3,1}$	$FP_{3,2}$	$TP_{3,3}$	$FN_{3,4}$
No-severity		$FP_{4,1}$	$FP_{4,2}$	$FP_{4,3}$	$TP_{4,4}$

possible to create effective searches by combining keywords using the logical operations “OR” and “AND” [52]. The retrieved tweets are stored in a comma-separated values (CSV) file, which includes the tweet ID, user name, date, and text for each tweet. For all the experiments, we collected data on road accidents from Twitter. By utilizing various hashtags and keywords, we were able to gather over 30K tweets. Among them, 5K tweets form four varying-sized datasets, as described in Table 2. Table 3 describes the distribution of the severity classes in each dataset’s labeled tweets. It is noted from Table 3 that the class labels in each dataset are equally distributed. In the context of automated accident tweet evaluation, we define each label as no-severity, minor, major, and fatal.

We also consider the “no-severity” class because some tweets may contain accident information with no harm conditions attached to them. A few examples of tweets in Table 4 illustrate the severity information from the tweet content. The type of accident label task can take values from the predefined classes. Each tweet was given one of four severity labels experienced by the three vehicles, such as the bike, car, and bus involved.

Annotation Process

One native English speaker familiar with the context of tweets annotated the dataset. To ensure the dataset’s quality, 500 randomly chosen tweets from the dataset were annotated by a second annotator who is also a native English speaker and familiar with the accident tweet environment. The inter-annotator agreement between two annotators was calculated using Cohen’s k coefficient, which is considered a statistical measure that is more trustworthy than a straightforward percentage agreement calculation [53].

Raw tweets are incredibly unorganized and often duplicate information. For effective processing, unwanted data from raw tweets like symbols and numbers must be deleted. Tweet categorization’s performance is impacted by noise and unstructured tweets. As a result, the tweets must be pre-processed. The steps for pre-processing are presented in “The Pre-processing Phase” section. The occurrence of words in the manually labeled data set is used to build the index for each word in the dictionary. Every tweet undergoes a transformation into a set of word indices, and these word index sequences are then padded to the same length. The resulting sequences are used to train the classifier in both the proposed and other models.

Methods Used for Comparative Analysis

Since we are utilizing the weighted ensemble of decision tree-based self-training semi-supervised learning for this task, to observe whether the performance of the proposed

model has changed or not, we are comparing the proposed approach with two models: one is a supervised learning model that is trained on an original set of labeled accident tweets data, and the other is the self-training semi-supervised learning model that utilizes a single decision tree classifier as the confidence estimator. In the first model, we feed the decision tree classifier the original labeled data set. This is because we use the decision tree classifier in the self-training phase to obtain pseudo-labels for the unlabeled tweets. Nevertheless, we can use any supervised learning model for the comparative analysis.

The self-training with one decision tree is also an iterative model, where, in the first iteration, a decision tree classifier will be trained using the original set of labeled data, and the pseudo-labels and their corresponding confidences will be generated using the trained DT model for the set of unlabelled data. In [50], Tanha et al. provided the working procedure for the self-training semi-supervised learning approach using the decision tree learning model. We have implemented the same model on the accident tweet severity datasets for the comparative analysis. We named this model as STDT (Self-Training with Decision Tree). Similarly to the proposed approach, a pre-defined threshold value is used to select the most likely pseudo-labels into the original training set while removing information about such tweet instances from the unlabelled tweets set. This iterative process will be continued until either the unlabeled set is empty or no more confident labels are available.

In addition to the above, in this work, we also perform the comparative analysis with the recent work called the RESSEL model [19]. This is, in contrast to our approach, which first builds a self-training semi-supervised learning model using each diverse classifier. That is, for each labeled data that is sampled with replacement, different self-training semi-supervised learning models were created (that label the unlabelled instances) using the different baseline classifiers. The trained models are then combined to obtain the best pseudo-labels for the unlabelled instances. In their work, they have used the out-of-sample bootstrap method at various labeled percentages (5%, 10%, and 20%), to validate their RESSEL approach. In this work, we choose the same methodology with 20% labeled data to obtain the pseudo-labels for the unlabelled instances, as they reported better predictions using 20% of the labeled instances.

Performance Measures

As aforementioned, this work categorizes the accident tweets into four categories, such as *Fatal*, *Minor*, *Major*, and *No-Severity*. Because the predictions are multi-class and each severity category follows the class label

hierarchy, this becomes an ordinal classification. The classes in ordinal classification are assigned the integer values in increasing order of their severity. Hence, the set of classes {No-severity, *Minor*, *Major*, *Fatal*} are represented by corresponding elements of the set {1, 2, 3, 4}. The multi-class ordinal confusion matrix is provided in Table 5.

We use traditional performance measures such as precision, recall, and accuracy to capture the predictions of the proposed classification model. The self-training model's performance can be estimated using the measure of accuracy since the data are balanced before training the model. In addition, we also proposed a measure called the risk factor to calculate the amount of risk that is present in the end application as a result of false-negative instances. A detailed explanation of these performance measures is given in the following sections.

Accuracy

Accuracy is calculated as the correctly predicted severity class labels from the total tested tweet instances [54]. Equation 5 calculates the accuracy from the confusion matrix 5.

$$\text{Accuracy} = \frac{\text{TP}_{1,1} + \text{TP}_{2,2} + \text{TP}_{3,3} + \text{TP}_{4,4}}{n} \times 100 \quad (5)$$

Here, n represents the total number of tweets in a test set.

Precision

In terms of the total number of forecasts for each severity class, the precision counts the number of accurate predictions [54]. The precision for each severity category is given as:

$$\text{Precision(Fatal)} = \frac{\text{TP}_{1,1}}{\text{TP}_{1,1} + \text{FP}_{2,1} + \text{FP}_{3,1} + \text{FP}_{4,1}} \quad (6)$$

$$\text{Precision(Major)} = \frac{\text{TP}_{2,2}}{\text{FN}_{1,2} + \text{TP}_{2,2} + \text{FP}_{3,2} + \text{FP}_{4,2}} \quad (7)$$

$$\text{Precision(Minor)} = \frac{\text{TP}_{3,3}}{\text{FN}_{1,3} + \text{FN}_{2,3} + \text{TP}_{3,3} + \text{FP}_{4,3}} \quad (8)$$

$$\text{Precision(No-Severity)} = \frac{\text{TP}_{4,4}}{\text{FN}_{1,4} + \text{FN}_{2,4} + \text{FN}_{3,4} + \text{TP}_{4,4}} \quad (9)$$

Table 6 Prediction Performances on various sizes of tweet datasets using precision and recall

S. no.	Dataset	Precision				Recall				Accuracy			
		DT		RESSEL		STDT		WESTDT		DT		RESSEL	
		Fatal	Minor	Major	No-severity	Fatal	Minor	Major	No-severity	Fatal	Minor	Major	No-severity
1	D1	0.669	0.738	0.745	0.755	0.612	0.767	0.757	0.754	0.659	0.710	0.779	0.784
2	D2	0.675	0.741	0.772	0.763	0.699	0.731	0.768	0.782	0.679	0.740	0.764	0.796
3	D3	0.684	0.752	0.768	0.770	0.655	0.762	0.781	0.760	0.695	0.762	0.793	0.798
4	D4	0.695	0.768	0.774	0.781	0.699	0.758	0.769	0.790	0.702	0.769	0.801	0.807
S. no.	Dataset	Precision				Recall				Accuracy			
		DT		RESSEL		STDT		WESTDT		DT		RESSEL	
		Fatal	Minor	Major	No-severity	Fatal	Minor	Major	No-severity	Fatal	Minor	Major	No-severity
1	D1	0.691	0.734	0.682	0.709	0.711	0.752	0.712	0.722	0.737	0.764	0.781	0.726
2	D2	0.702	0.733	0.714	0.711	0.730	0.721	0.749	0.721	0.748	0.738	0.758	0.739
3	D3	0.719	0.742	0.728	0.718	0.742	0.735	0.729	0.719	0.759	0.750	0.768	0.741
4	D4	0.732	0.753	0.739	0.724	0.751	0.759	0.739	0.734	0.763	0.765	0.793	0.749

Table 7 Prediction performances on various sizes of datasets using Accuracy

S. no.	Data set	Accuracy			
		DT	STDT	RESSEL	WESTDT
1	D1	0.785	0.798	0.799	0.817
2	D2	0.794	0.799	0.801	0.821
3	D3	0.809	0.820	0.798	0.838
4	D4	0.801	0.821	0.813	0.854

The best accuracy values are highlighted in bold

Recall

Contrary to the measure precision, the recall measures the portion of correct predictions from the actual observations of each severity class [54]. The recall for each severity category is given as:

$$\text{Recall (Fatal)} = \frac{TP_{1,1}}{TP_{1,1} + FN_{1,2} + FN_{1,3} + FN_{1,4}} \quad (10)$$

$$\text{Recall (Major)} = \frac{TP_{2,2}}{FP_{2,1} + TP_{2,2} + FN_{2,3} + FN_{2,4}} \quad (11)$$

$$\text{Recall (Minor)} = \frac{TP_{3,3}}{FP_{3,1} + FP_{3,2} + TP_{3,3} + FN_{3,4}} \quad (12)$$

$$\text{Recall (No-Severity)} = \frac{TP_{4,4}}{FP_{4,1} + FP_{4,2} + FP_{4,3} + TP_{4,4}} \quad (13)$$

Risk Factor

The risk factor measure informs users about the level of risk associated with the final application as a result of false-negative occurrences. The risk factor for any class $r \in \{1, 2, 3, 4\}$ is calculated as follows:

$$RF(r) = \frac{1}{N_r} \times \sum_{s=1}^4 [FN_{r,s} \times (s - r) \times B_{r,s}] \quad (14)$$

Where,

$$B_{r,s} = \begin{cases} 1, & r < s \\ 0, & \text{Otherwise} \end{cases}$$

Here, the set of classes {No-severity, *Minor*, *Major*, *Fatal*} are represented by corresponding elements of the set $\{1, 2, 3, 4\}$. From Table 5, $FN_{r,s}$ represents the false negative instances. It means the number of tweets with actual class r was misclassified into the low level of severity s . $RF(r)$ represents the risk factor for the numerical representation of the severity class r , and N_r is the number of tweets in class r .

The risk factor is derived from the mean absolute error (MAE) [18], where the MAE is the magnitude of difference between the prediction of a tweet and the true value of that tweet. Note that the MAE is calculated for both the false negatives and the false positives. The risk factor, however, is only computed for false negatives (the inequality $r < s$ ensures the false negative information from the confusion matrix), as we are interested in assessing the level of risk associated with the implementation of the prediction model. Since the severity represents the level of urgency, this multi-class classification problem will also be treated as the ordinal classification. Hence, by utilizing the numerical ordinal value, we calculated the risk factor value for each severity class except for the “no-severity” class. Here, we assume the information that is present in the tweet class “no-severity” does not exhibit the severity of the accident. Also, to indicate the impact of the misclassification, we have assigned the weights (that is $s - r$) to each FNs where the highest weight indicates the highly severe information (for example, fatal) in a tweet is misclassified as a no-severity tweet. Similarly, the lowest weight will be observed in three cases, such as when a fatal tweet is misclassified into major, a major severity tweet is misclassified into minor, or a minor severity class is misclassified into no-severity. For any prediction model,

Table 8 Anova test for the Precision of the fatal (accident-severity) WESTDT, RESSEL, STDT, and DT models utilizing all four datasets

Source of variation	Sum of square	df	Mean square	f-value	p-value	f-critical
Between the groups	0.011803	3	0.003934	6.605709	0.00694	3.490295
Within the groups	0.007147	12	0.000596			
Total	0.01895	15				

Table 9 Anova test for the Precision of major (accident-severity) WESTDT, RESSEL, STDT, and DT models utilizing all four datasets

Source of variation	Sum of square	df	Mean square	f-value	p-value	f-critical
Between the groups	0.003725	3	0.001242	3.514478	0.04911	3.490295
Within the groups	0.004239	12	0.000353			
Total	0.007964	15				

Table 10 Anova test for the Precision of minor (accident-severity) WESTDT, RESSEL, STDT, and DT models utilizing all four datasets

Source of variation	Sum of square	<i>df</i>	Mean square	f-value	<i>p</i> -value	f-critical
Between the groups	0.002593	3	0.000864	5.882587	0.010413	3.490295
Within groups	0.001763	12	0.000147			
Total	0.004356	15				

Table 11 Anova test for the Precision of no-severity (accident-severity) WESTDT, RESSEL, STDT, and DT models utilizing all four datasets

Source of variation	Sum of square	<i>df</i>	Mean square	f-value	<i>p</i> -value	f-critical
Between the groups	0.001972	3	0.000657	4.307398	0.027972	3.490295
Within groups	0.001832	12	0.000153			
Total	0.003804	15				

Table 12 Anova test for the Recall of fatal (accident-severity) WESTDT, RESSEL, STDT, and DT models utilizing all four datasets

Source of variation	Sum of square	<i>df</i>	Mean square	f-value	<i>p</i> -value	f-critical
Between the groups	0.005842	3	0.001947	7.868255	0.00362	3.490295
Within groups	0.00297	12	0.000247			
Total	0.008811	15				

Table 13 Anova test for the Recall of major (accident-severity) WESTDT, RESSEL, STDT, and DT models utilizing all four datasets

Source of variation	Sum of square	<i>df</i>	Mean square	f-value	<i>p</i> -value	f-critical
Between the groups	0.003073	3	0.001024	5.283903	0.014886	3.490295
Within groups	0.002327	12	0.000194			
Total	0.0054	15				

Table 14 Anova test for the Recall of minor (accident-severity) WESTDT, RESSEL, STDT, and DT models utilizing all four datasets

Source of variation	Sum of square	<i>df</i>	Mean square	f-value	<i>p</i> -value	f-critical
Between the groups	0.006327	3	0.002109	4.987486	0.017908	3.490295
Within groups	0.005075	12	0.000423			
Total	0.011402	15				

Table 15 Anova test for the Recall of no-severity (accident-severity) WESTDT, RESSEL, STDT, and DT models utilizing all four datasets

Source of variation	Sum of square	<i>df</i>	Mean square	f-value	<i>p</i> -value	f-critical
Between the groups	0.001377	3	0.000459	9.737075	0.001548	3.490295
Within groups	0.000566	12	4.71E-05			
Total	0.001943	15				

Table 16 Anova test for the Accuracy of the WESTDT, RESSEL, STDT, and DT models utilizing all four datasets

Source of variation	Sum of square	<i>df</i>	Mean square	f-value	<i>p</i> -value	f-critical
Between the groups	0.002882	3	0.000961	6.378976	0.007859	3.490295
Within groups	0.001808	12	0.000151			
Total	0.00469	15				

minimizing the risk factor is the primary criteria as it may affect the end application.

Detailed empirical results in terms of risk factors are provided in “[Results in Terms of Risk Factor](#)” section.

Results and Discussion

In this section, we present the results obtained from the

Table 17 Anova test for the Risk Factor for Fatal class of the WESTDT, RESSEL, STD, and DT models utilizing all four datasets

Source of variation	Sum of square	df	Mean square	f-value	p-value	f-critical
Between the groups	0.00078	3	0.00026	4.373583	0.026746	3.490295
Within groups	0.000713	12	5.94E−05			
Total	0.001493	15				

Table 18 Anova test for the Risk Factor for Major class of WESTDT, RESSEL, STD, and DT models utilizing all four datasets

Source of variation	Sum of square	df	Mean square	f-value	p-value	f-critical
Between the groups	0.000527	3	0.000176	5.556519	0.012618	3.490295
Within groups	0.000379	12	3.16E−05			
Total	0.000906	15				

Table 19 Anova test for the Risk Factor for Minor class of WESTDT, RESSEL, STD, and DT models utilizing all four datasets

Source of variation	Sum of square	df	Mean square	f-value	p-value	f-critical
Between the groups	0.000121	3	4.02E−05	4.509781	0.022926	3.490295
Within groups	0.000179	12	1.49E−05			
Total	0.0003	15				

experiments conducted as part of our work. Here, we developed three models as discussed in “[Methods Used for Comparative Analysis](#)” section—a basic supervised classifier of Decision Tree (DT) and two semi-supervised models—Self-training with Decision Tree (STD), RESSEL model, and the proposed WESTDT model. All of the models are implemented on the four datasets described in Table 2.

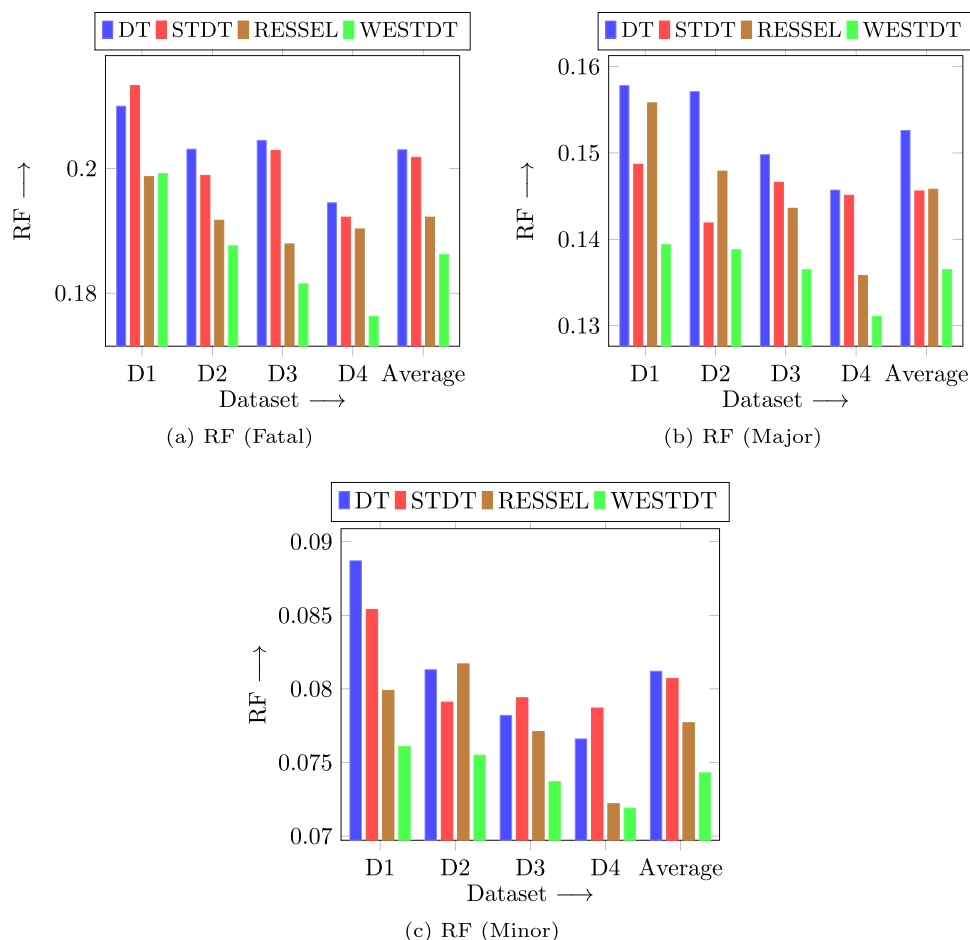
Results in Terms of Traditional Measures

The WESTDT model begins with a limited quantity of labeled accident severity data and iteratively integrates unlabeled data to enhance its performance. The hyperparameter tuning in the proposed WESTDT follows a two-stage method, involving the identification of optimal values for the decision tree’s hyperparameters, as well as the self-training parameters, including the threshold and the number of iterations. The threshold has an impact on the quality of the pseudo-labeled data that is added to the training set. Moreover, the self-training iterations play a crucial role as a hyperparameter, determining how often new pseudo-labeled data is included in the training set. We employed a repeated 10-fold cross-validation technique and accuracy as the metrics to determine the optimal parameter values. In the 10-fold cross-validation, we iterated the self-training model multiple times using various random splits and then calculated the average performance across multiple runs to provide an accurate estimate of the model’s performance. We initially determined the hyperparameters of the decision tree on the original set of labeled cases. Next, we determine the optimal hyperparameter values for the self-training model. The experimental assessments indicate that increasing the

threshold value can enhance the overall performance of the model. This is because it incorporates highly confident intervals from the original training set in each iteration. In this study, we assessed all the datasets using the proposed model, with the acceptability threshold established at 0.9. Considering the number of iterations, we achieved optimal performance while using a value of 6. Therefore, we assessed and demonstrated the effectiveness of the WESTDT model by specifying the number of iterations as 6.

Table 6 shows that the proposed method of WESTDT performs better on each severity class across all four datasets in terms of accuracy and recall measures. The proposed model exhibits the best improvement in accuracy by 18.3% over the RESSEL model on dataset D1 for the “Fatal” class. Also, the proposed model demonstrates the maximum improvement of recall for the same class is 7.6% when compared to model DT, implemented on dataset D1. Similarly, for the class “Major”, the highest improvement in precision is 9.7%, over the model STD when implemented on dataset D1. In the same class, our proposed model records the highest increment in the recall by 6.4% when compared to the model DT on dataset D1. For the class of “Minor”, the proposed model exhibits the highest increment in precision by 5.3%, over the model DT, implemented on D1. In the same case, the model records the highest improvement in the recall by 9.3%, over the model DT, on dataset D1. For the class of “No-severity”, the highest improvement in precision is 5% when compared to the model RESSEL implemented on dataset D3. Moreover, for the same class, the largest increment in the recall is observed as 3.9% when compared to the model DT implemented on dataset D2. Hence, it is noted that the proposed model of WESTDT exhibits good improvements in

Fig. 3 Risk factor values of three severity classes such as **a** fatal, **b** major, and **c** minor for the models such as DT, STDT, RESSEL, and WESTDT, respectively, on the datasets D1, D2, D3, and D4



both precision and recall among all classes when compared to all other models on all datasets.

Table 7 displays the accuracy for each model when applied to all four datasets. Table 7 shows that, when compared to all other models, the proposed model had the best accuracy across all datasets. Moreover, the largest improvement in accuracy of the proposed WESTDT model is 6.6% when compared to the model DT on dataset D4.

Statistical Significance Test

The self-training learning mechanism is attributed to providing high-confident class labels to the large set of unlabelled tweets. We implement a supervised learning model (a decision tree model) before and after executing the proposed self-training learning strategy to determine the variation in the achieved performances. The original set of labeled Twitter data and the final pseudo-labeled tweet data are used to train a decision tree learning model.

We employ the statistical test known as Analysis of Variance (ANOVA) to determine the statistical significance of the proposed WESTDT model over the other models implemented in our work [41, 55, 56]. ANOVA is a statistical

method that allows for the comparison of means across many groups, indicating if there are any significant differences between these means. It helps in validating the results of the proposed WESTDT model. For this purpose, we define a null and alternate hypothesis as follows:

H_0 : The performance of the DT model that is trained on the original set of tweets data is equal to the performance of the DT model that is trained on the pseudo-labeled tweets data predicted from the proposed weighted ensemble-based self-training model.

H_1 : The performance of the DT model that is trained on the original set of tweets data differs from the performance of the DT model that is trained on the pseudo-labeled tweets data predicted from the proposed weighted ensemble-based self-training model.

In essence, the null hypothesis (H_0) signifies the absence of any significant variation among the average performances of the proposed and other models under comparison. Whereas the alternate hypothesis (H_1) signifies the statistical difference between the proposed model and the other models considered. The final result of the test is to either reject or accept the null hypothesis (H_0). In the ANOVA test, the null hypothesis is rejected if the α -value is less than the threshold

value of 0.05 [55]. We have performed the ANOVA test using the Microsoft Excel 2020 tool. The test is applied to all four datasets.

Tables 8, 9, 10 and 11, respectively, present the results of the ANOVA test on the precision measure obtained for the classes fatal, major, minor and no-severity. In the same way, Tables 12, 13, 14 and 15 present the results of the ANOVA test corresponding to the recall measure recorded by the classes of fatal, major, minor and no-severity. Table 16 shows the results of the ANOVA test on the accuracy measure obtained from the experiments. Finally, Tables 17, 18 and 19 display the results of the ANOVA test corresponding to the risk factor measure obtained for the classes fatal, major, and minor.

From the Tables 8, 9, 10, 11, 12, 13, 14 and 15, it is noted that the total models built for each class are sixteen as each of the four classifiers are built on all four datasets. Similarly, in Table 16, it is noted that the total number of models built is sixteen. Similar to the previous measure, Tables 17, 18 and 19 reveal that a total of sixteen models were created for each class of risk factor because each of the four classifiers was created using all four datasets. The degree of freedom (df) is calculated in a two-step procedure: In the first step, it is computed as one less than the total models built. It is derived by subtracting the total number of evaluation metrics for all datasets from the total number of models in the second step.

The condition for rejecting the null hypothesis is given as, $f\text{-value} > f\text{-critical}$. It is observed from all the tables of ANOVA results (i.e., from Tables 8, 9, 10, 11, 12, 13, 14 and 15 and Tables 17, 18 and 19) that $f\text{-value} > f\text{-critical}$. It can be inferred that the models are not the same, as the null hypothesis is rejected with $p\text{-value} < 0.05$ in all the cases. Moreover, in the case of the ANOVA test on the accuracy, it is observed that $f\text{-value} > f\text{-critical}$ and $p\text{-value} < 0.05$, making the null hypothesis to be rejected. Hence, all of the results obtained are statistically significant, and we conclude that the proposed WESTDT exhibits significant improvement in performance over the other models considered.

Results in Terms of Risk Factor

Figure 3a–c depicts the risk factor result for three severity classes: fatal, major, and minor using the prediction models such as DT, STDT, RESSEL, and WESTDT, respectively, on the datasets such as D1, D2, D3, D4. The figure also presents the average results on the four datasets by the four models.

Figure 3a–c shows that the WESTDT model outperformed the other models, such as DT, STDT, and RESSEL, on the datasets such as D1, D2, D3, and D4 for three severity classes: fatal, major, and minor. The average risk

factor for the Fatal class (the Fig. 3a) for the model DT is high (it is recorded as 0.2030) when compared with the models such as STDT (it is recorded as 0.2018), RESSEL (it is recorded as 0.1922) and WESTDT (it is recorded as 0.1862).

In terms of the severity class Major (the Fig. 3b), the WESTDT's average risk factor values are less (it is recorded as 0.1365) when compared with the models such as DT (it is recorded as 0.1526), STDT (it is recorded as 0.1456), and RESSEL (it is recorded as 0.1458). Similarly, in terms of the severity class Minor (the Fig. 3c), the WESTDT's average risk factor values are less (it is recorded as 0.0743) when compared with the models such as DT (it is recorded as 0.0812), STDT (it is recorded as 0.0807), and the RESSEL model (it is recorded as 0.0777). Among these, the performances of the RESSEL model are close to our proposed WESTDT model in terms of the three severity labels such as fatal, major, and minor. These results indicate an advantage of utilizing the ensemble of the supervised models (in this case, decision trees) in the process of obtaining the most suitable pseudo-labels. However, future prediction models should concentrate on minimizing the risk factor values.

Discussion

This work is intended to observe the class labels for unlabelled instances. We have utilized the weighted ensemble of decision tree-based self-training semi-supervised learning classifiers for this. Technically, the proposed approach also acts as the classification system as supervised learning (weighted ensemble of decision tree model) is used in the self-training phase. It is observed from Tables 7 and 6 that, as the size of the unlabelled set increases, the precision, recall, and accuracy values also gets increase. For example, on dataset D1, the proposed WESTDT is recorded as 0.817, whereas on dataset D4, the accuracy is recorded as 0.854. Therefore, there is an increment of 0.037 accuracy points between these two datasets.

Similarly, the performances in terms of risk factors also show the benefit of using the proposed WESTDT model. It is observed from Fig. 3 that, in terms of Fatal class, on the dataset D1, the risk factor value is observed as 0.1992. The risk factor value is also reduced as the unlabelled data points increase. That is, on dataset D4, the risk factor value is observed as 0.1763. Hence, there is a reduction of 0.0229 risk factor values when utilizing more unlabelled data points. A similar result can also be observed in the Major and Minor cases. Hence, as the unlabelled set increases, we can observe the likelihood of an increment in the performance of the proposed WESTDT model.

Threats to Validity

Since the performance of the machine learning models depends on various factors such as the dimensionality, data size, training time, complexity, cost, etc [57]. In this regard, we discuss potential risks that may impair the performance of the proposed WESTDT model in this section.

Construct Validity

In the construction of the WESTDT model, we have chosen the size of the ensemble as 10, as per the study by Opitz and Maclin [58], which is an appropriate sample number (or the size of the ensemble) to generate the classifiers. Nevertheless, the size of the ensemble may affect the final prediction performance. If the size of the labeled tweets data and the unlabelled data are sufficiently large, then the performance of the WESTDT model will vary at different ensemble sizes.

In addition to the above, the size of the ensemble can be increased using various techniques other than the size of the ensemble, such as the multi-inducer system, parameter optimization, feature subspace sampling, manipulating the training data (that is, including large sets of tweets data), and changing the output representation [59]. A recent study by Sharma et al. [60] also suggests using large sets of diverse ensemble classifiers to obtain better performances. However, the computational complexity increases as the number of diverse classifiers in the ensemble grows [59]. Therefore, a better trade-off between the computational complexity and the diverse classifiers must be achieved for better predictive performances.

Internal Validity

In constructing the proposed WESTDT model, in the text-vectorization phase, we utilized a pretrained word embedding model called fastText to vectorize the tweet information. Nevertheless, the proposed model can be implemented using other approaches like Word2Vec, BERT embedding, BoW, TF-IDF, etc. Since our intention is to develop a framework that assigns class labels for the large portion of unlabelled tweets, we prioritized using fastText implementation according to the literature [48, 49].

In the semi-supervised learning phase, we utilized a decision tree model to observe the pseudo-labels. This is because, in [50], Tanha et al. suggested the advantage of using decision tree implementation in observing the pseudo-labels. However, we can use different supervised learning models to observe the change in prediction performance by assigning pseudo-labels to unlabeled tweet instances.

In addition to the above, we have generated a uniform labeled (or balanced) accident tweet dataset in this work.

However, we may experience an imbalance between the severity levels in real-world scenarios. Hence, the proposed WESTDT model's performance may also vary to include techniques for addressing the class imbalance problem.

External Validity

In this work, the proposed model has been evaluated only on the accident severity tweets. Nevertheless, the model can be extended by validating the other tasks such as accident cause (speed drive, drunken drive, bad road, wrong drive, vehicle broke-down, collision, road-block, fire, etc.) and type (bus, car, bike, airplane, train, etc.) prediction models. The generalisability is achieved by validating the proposed model on the above datasets.

Conclusion and Future Work

The present work has executed a dynamic labeling framework called Weighted Ensemble-based Self-Training with Decision Tree (WESTDT) semi-supervised learning solution for the accident tweets. The suggested self-training-based system employs an iterative methodology in which the ensemble of decision trees is used to train the base classifier. At the end of all iterations, we get pseudo-severity labels for unlabeled accident tweets. Four datasets are used to verify the models: first, a traditional supervised classifier of DT with fully labeled severity tweets. Second, based on a self-training strategy of semi-supervised learning using an STDT classifier; the third, the proposed WESTDT semi-supervised learning; and the RESSEL model. Empirical results demonstrate that the proposed WESTDT framework improves precision and recall by 5–18.3% and 3.9–9.3%, respectively, when compared to all other models on all the datasets. Moreover, the proposed model achieves the highest improvement in accuracy by 6.6% when compared with all models on all datasets. All the results obtained are proved to be statistically significant using the ANOVA test. In addition to proposing the WESTDT model, we also proposed a novel performance measure called the risk factor to estimate the amount of risk that is present in the application with the use of the prediction model. In terms of the proposed risk factor measure, the proposed WESTDT shows a reduction in the values when compared with the other models, such as DT, STDT, and the RESSEL. The results indicate the effectiveness of the proposed WESTDT model over the other models, such as DT, STDT, and RESSEL, in terms of the proposed and traditional measures. Hence, it assures the possibility of achieving better traffic safety transportation systems through the proposed WESTDT model.

Future research into this problem will incorporate the diversity-based ensemble models within the self-training approach for detecting accident severity using tweet data. Another possible future direction is to filter out fake accident tweets for robust performance of accident severity classification models.

Author Contributions Sanjib Kumar Raul: Conception, Design, Proposed Model, Model Analysis, Drafting, Data collection, Experimentation, Review, and Approve. Rashmi Ranjan Rout: Conception, Design, Model Analysis, Drafting, Review, and Approve. D.V.L.N. Somayajulu: Conception, Design, Model Analysis, Drafting, Review, and Approve.

Funding Not applicable.

Data Availability On reasonable request from the corresponding author.

Code Availability On reasonable request from the corresponding author.

Declarations

Conflict of interest The authors affirm that they have no known Conflict of interest.

Ethical Approval The submitted work by the authors is entirely their own, and it has not been published or considered for publication elsewhere.

Informed Consent Not applicable.

References

- World health organization report on road traffic injuries. <https://www.who.int/news-room/commentaries/detail/it-s-time-to-get-serious-in-addressing-the-leading-killer-of-our-youth>; 2018.
- World health organization global status report on road safety. <https://www.who.int/publications/i/item/9789241565684>; 2018.
- Katanalp BY, Eren E. The novel approaches to classify cyclist accident injury-severity: hybrid fuzzy decision mechanisms. *Accident Anal Prevent*. 2020;144:105590.
- Ma Z, Mei G, Cuomo S. An analytic framework using deep learning for prediction of traffic accident injury severity based on contributing factors. *Accident Anal Prevent*. 2021;160:106322.
- Jeong H, Jang Y, Bowman PJ, Masoud N. Classification of motor vehicle crash injury severity: a hybrid approach for imbalanced data. *Accident Anal Prevent*. 2018;120:250–61.
- Yang Z, Zhang W, Feng J. Predicting multiple types of traffic accident severity with explanations: a multi-task deep learning framework. *Saf Sci*. 2022;146: 105522.
- Liu L, Guevara A, Sanchez-Galan JE. Identification and classification of road traffic incidents in panama city through the analysis of a social media stream and machine learning. *Intell Syst Appl*. 2022;16: 200158.
- Ali F, Ali A, Imran M, Naqvi RA, Siddiqi MH, Kwak K-S. Traffic accident detection and condition analysis based on social networking data. *Accident Anal Prevent*. 2021;151: 105973.
- Sameen M, Pradhan B. Severity prediction of traffic accidents with recurrent neural networks. *Appl Sci*. 2017;7(6):476.
- Gan J, Li L, Zhang D, Yi Z, Xiang Q. An alternative method for traffic accident severity prediction: using deep forests algorithm. *J Adv Transp*. 2020;1–13:2020.
- Assi K, Rahman SM, Mansoor U, Ratrou N. Predicting crash injury severity with machine learning algorithm synergized with clustering technique: a promising protocol. *Int J Environ Res Public Health*. 2020;17(15):5497.
- Gutierrez-Orsorio C, González FA, Pedraza CA. Deep learning ensemble model for the prediction of traffic accidents using social media data. *Computers*. 2022;11(9):126.
- Liu H, Kumar S, Morstatter F. Twitter data analytics. Springer briefs in computer science. London: Springer; 2014.
- Wang C, Nulty P, Lillis D. Transformer-based multi-task learning for disaster tweet categorisation. Preprint [arXiv:2110.08010](https://arxiv.org/abs/2110.08010); 2021.
- Abbas AM. Social network analysis using deep learning: applications and schemes. *Soc Netw Anal Min*. 2021;11(1):106.
- Taamneh M, Alkheder S, Taamneh S. Data-mining techniques for traffic accident modeling and prediction in the United Arab Emirates. *J Transp Saf Sec*. 2017;9(2):146–66.
- Salam S, Islam MS, Ahmed F, Khan L, Kim D, Allo N, Nwariaku O. Exploring the roles of social media data to identify the locations and severity of road traffic accidents. In: 2021 IEEE 4th international conference on artificial intelligence and knowledge engineering (AIKE). IEEE; 2021. p. 62–71.
- Willmott CJ, Matsuura K. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Clim Res*. 2005;30(1):79–82.
- de Vries S, Thierens D. A reliable ensemble based approach to semi-supervised learning. *Knowl Based Syst*. 2021;215: 106738.
- Zheng M, Li T, Zhu R, Chen J, Ma Z, Tang M, Cui Z, Wang Z. Traffic accident's severity prediction: a deep-learning approach-based CNN network. *IEEE Access*. 2019;7:39897–910.
- Azhar A, Rubab S, Khan MM, Bangash YA, Alshehri MD, Illahi F, Bashir AK. Detection and prediction of traffic accidents using deep learning techniques. *Clust Comput*. 2022;1–17:2022.
- Vapnik V. The nature of statistical learning theory. London: Springer; 1999.
- Han L, Luo S, Jianmin Yu, Pan L, Chen S. Rule extraction from support vector machines using ensemble learning approach: an application for diagnosis of diabetes. *IEEE J Biomed Health Inform*. 2014;19(2):728–34.
- Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data mining, inference, and prediction. London: Springer; 2009.
- Devlin J, Chang M-W, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. Preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805); 2018.
- Bokaba T, Doorsamy W, Paul BS. A comparative study of ensemble models for predicting road traffic congestion. *Appl Sci*. 2022;12(3):1337.
- Iranmanesh M, Seyedabrishami S, Moridpour S. Identifying high crash risk segments in rural roads using ensemble decision tree-based models. *Sci Rep*. 2022;12(1):20024.
- Jamal A, Zahid M, Rahman MT, Al-Ahmadi HM, Almoshaogeh M, Farooq D, Ahmad M. Injury severity prediction of traffic crashes with ensemble machine learning techniques: a comparative study. *Int J Injury Control Saf Promot*. 2021;28(4):408–27.
- Umamaheswara SB, Sadam R. Towards developing and analysing metric-based software defect severity prediction model. e-prints, [arXiv–2210](https://arxiv.org/abs/2210.08010); 2022.

30. Wei C, Sohn K, Mellina C, Yuille A, Yang F. Crest: a class-rebalancing self-training framework for imbalanced semi-supervised learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2021. p. 10857–66.
31. Roli F, Marcialis GL. Semi-supervised PCA-based face recognition using self-training. In: Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR). Springer; 2006. p. 560–8.
32. Nartey OT, Yang G, Asare SK, Wu J, Frempong LN. Robust semi-supervised traffic sign recognition via self-training and weakly-supervised learning. *Sensors*. 2020;20(9):2684.
33. Zhang Y, Park DS, Han W, Qin J, Gulati A, Shor J, Jansen A, Xu Y, Huang Y, Wang S, et al. Bigssl: exploring the frontier of large-scale semi-supervised learning for automatic speech recognition. *IEEE J Sel Top Signal Process*. 2022;2022:1.
34. Wang X, Kihara D, Luo J, Qi G-J. Enaet: a self-trained framework for semi-supervised and supervised learning with ensemble transformations. *IEEE Trans Image Process*. 2020;30:1639–47.
35. Liu Z, Wen T, Sun W, Zhang Q. Semi-supervised self-training feature weighted clustering decision tree and random forest. *IEEE Access*. 2020;8:128337–48.
36. Madisetty S, Desarkar MS. A neural network-based ensemble approach for spam detection in twitter. *IEEE Trans Comput Soc Syst*. 2018;5(4):973–84.
37. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. Preprint [arXiv:1301.3781](https://arxiv.org/abs/1301.3781); 2013.
38. Pennington J, Socher R, Manning C. Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP); 2014. p. 1532–43.
39. Johnson R, Zhang T. Effective use of word order for text categorization with convolutional neural networks. Preprint [arXiv:1412.1058](https://arxiv.org/abs/1412.1058); 2014.
40. Ghosh S, Chakraborty P, Cohn E, Brownstein JS, Ramakrishnan N. Characterizing diseases from unstructured text: a vocabulary driven word2vec approach. In: Proceedings of the 25th ACM international on conference on information and knowledge management. ACM; 2016. p. 1129–38.
41. Raul SK, Rout RR, Somayajulu DVLN. Topic classification using regularized variable-size CNN and dynamic BPSO in online social network. *Arab J Sci Eng*. 2023;2023:1–23.
42. Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching word vectors with subword information. *Trans Assoc Comput Ling*. 2017;5:135–46.
43. Symeonidis S, Effrosynidis D, Arampatzis A. A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis. *Expert Syst Appl*. 2018;110:298–310.
44. Kouloumpis E, Wilson T, Moore J. Twitter sentiment analysis: the good the bad and the omg! In: Proceedings of the international AAAI conference on web and social media, vol. 5; 2011. p. 538–41.
45. Chanda S, Pal S. The effect of stopword removal on information retrieval for code-mixed data obtained via social media. *SN Comput Sci*. 2023;4(5):494.
46. Loper E, Bird S. Nltk: the natural language toolkit. Preprint [arXiv:cs/0205028](https://arxiv.org/abs/cs/0205028); 2002.
47. Hardeniya N, Perkins J, Chopra D, Joshi N, Mathur I. Natural language processing: Python and NLTK. London: Packt Publishing Ltd; 2016.
48. Mikolov T, Grave E, Bojanowski P, Puhersch C, Joulin A. Advances in pre-training distributed word representations. Preprint [arXiv:1712.09405](https://arxiv.org/abs/1712.09405); 2017.
49. Grave E, Bojanowski P, Gupta P, Joulin A, Mikolov T. Learning word vectors for 157 languages. Preprint [arXiv:1802.06893](https://arxiv.org/abs/1802.06893); 2018.
50. Tanha J, Van Someren M, Afsarmanesh H. Semi-supervised self-training for decision tree classifiers. *Int J Mach Learn Cybern*. 2017;8(1):355–70.
51. Masse M. REST API design rulebook: designing consistent RESTful web service interfaces. London: O'Reilly Media Inc; 2011.
52. Alomari E, Mehmood R. Analysis of tweets in Arabic language for detection of road traffic conditions. In: Smart societies, infrastructure, technologies and applications: first international conference, SCITA 2017, Jeddah, Saudi Arabia, November 27–29, 2017, proceedings 1. Springer; 2018. p. 98–110.
53. McHugh ML. Interrater reliability: the kappa statistic. *Biochem Med*. 2012;22(3):276–82.
54. Sarkar S, Pramanik A, Maiti J, Reniers G. Predicting and analyzing injury severity: a machine learning-based approach using class-imbalanced proactive and reactive data. *Saf Sci*. 2020;125:104616.
55. Demšar J. Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res*. 2006;7:1–30.
56. Panda SK, Pande SK, Das S. Task partitioning scheduling algorithms for heterogeneous multi-cloud environment. *Arab J Sci Eng*. 2018;43(2):913–33.
57. Bishop CM, Nasrabadi NM. Pattern recognition and machine learning, vol. 4. London: Springer; 2006.
58. Opitz D, Maclin R. Popular ensemble methods: an empirical study. *J Artif Intell Res*. 1999;11:169–98.
59. Rokach L. Pattern classification using ensemble learning. *Ser Mach Percept Artif Intell*. 2010;75:1.
60. Sharma U, Sadam R. How far does the predictive decision impact the software project? The cost, service time, and failure analysis from a cross-project defect prediction model. *J Syst Softw*. 2023;195: 111522.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.