IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Impact of SMOTE on Imbalanced Text Features for Toxic Comments Classification using RVVC Model

VAIBHAV RUPAPARA[1,†], FURQAN RUSTAM[2,†], HINA FATIMA SHAHZAD[2], ARIF MEHMOOD[3], IMRAN ASHRAF[4,*], AND GYU SANG CHOI.[4,*]

[1]School of Computing and Information Sciences, Florida International University, USA
[2]Department of Computer Science, Khwaja Fareed University of Engineering and Information Technology, Rahim Yar Khan, Punjab 64200, Pakistan
[3] Department of Computer Science & Information Technology, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan
[4]Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38544, South Korea

Corresponding author: Imran Ashraf and Gyu Sang Choi, (Email:ashrafimran@live.com,castchoi@ynu.ac.kr)
† Primary co-authors. These authors contributed equally to this work.

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2019R1A2C1006159, and in part by the Ministry of Science and ICT (MSIT), South Korea, through the Information Technology Research Center (ITRC) support program supervised by the Institute for Information and communications Technology Promotion (IITP) under Grant IITP-2021-2016-0-00313.

ABSTRACT Social media platforms and microblogging websites have gained accelerated popularity during the past few years. These platforms are used for expressing views and opinions about products, personalities, and events. Often during discussions and debates, fights take place on social media platforms which involves using rude, disrespectful, and hateful comments called toxic comments. The identification of toxic comments has been regarded as an essential element for social media platforms. This study introduces an ensemble approach, called regression vector voting classifier (RVVC), to identify the toxic comments on social media platforms. The ensemble merges the logistic regression and support vector classifier under soft voting criteria. Several experiments are performed on the imbalanced and balanced dataset to analyze the performance of the proposed approach. For data balance, the synthetic minority oversampling technique (SMOTE) is used on the imbalanced dataset. Furthermore, two feature extraction approaches are utilized to investigate their suitability such as term frequency-inverse document frequency (TF-IDF) and bag-of-words (BoW). The performance of the proposed approach is compared with several machine learning classifiers using accuracy, precision, recall, and F1-score. Results suggest that RVVC outperforms all other individual models when TF-IDF features are used with SMOTE balanced dataset and achieves an accuracy of 0.97.

INDEX TERMS Toxic comments classification; ensemble classifier; synthetic minority oversampling technique, TF-IDF; BoW; text classification; data re-sampling

I. INTRODUCTION

SOCIAL media platforms and microblogging websites have gained accelerated popularity for social communication between individuals and groups. Through these platforms, people share their thoughts, ideas, opinions and express their feelings using comments and feedback [1]. The number of internet users has been increasing gradually each year, from 2.4 billion in 2014 to 3.4 billion, 4 billion, and 4.4 billion in 2016, 2017, and June 2019, respectively [2]. As of May 2020, the number of internet users is increased to 4,648 billion [3]. Social media platforms provide a common ground for these users to share opinions and discuss ideas. However, problems arise when debates take a dirty side and fights take place on social media platforms which involves using rude, disrespectful, and hateful comments called toxic comments. Text in online comments contain many hazards such as fake news, cyberbullying, online harassment and toxicity [4]. Unfortunately, these toxic comments have become a serious issue that affects the reputation of social platforms and cause different psychological problems for users, such as depression, frustration, and even suicidal thoughts [1]. Toxic comment classification is very important to overcome the above-mentioned issues and maintain stability in online debates [5]. Toxic comments can be considered as a personal

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3083638, IEEE Access

attack, online harassment, and bullying behaviors. Over the past few years, several cases of police arrests happened where police arrested many individuals due to the abusive or negative content on personal pages [6], [7].

So a framework that can detect toxic comments and prevent publishing is of significant importance. As a result, several approaches have been introduced for the automatic detection of toxic comments using machine learning algorithms. For example, the study [8] combines machine learning and crowd-sourcing to classify the comments that are considered a personal attack. Support vector machines were also used by [9] for Cyberbullies detection. The cyberbullies are also detected in [10] using deep learning models. Despite the proposed approaches, there is a need to model more approaches to provide high accuracy for toxic comments. This study introduces an ensemble approach for toxic comments detection in imbalanced datasets and makes the following contributions

- This study proposes a novel approach, called regression vector voting classifier (RVVC), for toxic comment classification. RVVC is an ensemble classifier that combines the logistic regression and support vector classifier through soft voting criteria.
- For evaluation, term frequency-inverse document frequency (TF-IDF) and bag-of-words (BoW) are utilized as feature extraction with imbalanced and imbalanced datasets. Synthetic minority oversampling technique (SMOTE) and random under-sampling technique are used for balancing the datasets.
- Several state-of-the-art models are used along with machine learning models including support vector machine (SVM), random forest (RF), gradient boosting machine (GBM), logistic regression (LR), and k-nearest neighbor (K-NN) for performance appraisal. Additionally, a recurrent neural network is implemented for toxic sentiment classification.

The rest of the paper is organized as follows. Section II discusses research papers from the literature which are closely related to the current study. Section III gives an overview of the machine learning algorithms adopted for the current research, as well as, the description of the dataset used for the experiment. The proposed approach is also presented in the same Section. Results are discussed in Section IV while the conclusion is given in Section V.

II. LITERATURE REVIEW

Toxic comments on social media platforms have been a source of a great stir between individuals and groups. A toxic comment is not only verbal violence but includes the comment that is rude, disrespectful, negative online behavior, or other similar attitudes that make someone leave a discussion. Therefore, the toxic comments identification on social platforms is an important task that can help to maintain its interruption and hatred-free operations. Consequently, a large variety of toxic comment approaches have been proposed. Three characteristics concerning toxic classification are evaluated: classification, feature dimension reduction, and feature importance.

The authors use a deep learning-based toxic comments classification approach in [11] for the imbalanced toxic dataset. The performance evaluation is carried out on Kaggle Wikipedia's talk page edits dataset which contains 159,571 records of toxic comments. The proposed approach makes a multi-class classification including toxic, threat, severe toxic, obscene, insult, and identity hate. Convolutional neural network (CNN), bidirectional long short- term memory (LSTM), bidirectional gated recurrent unit (GRU), and the ensemble of the three models are used for classification. Results indicate that the ensemble approach gives the highest classification with an F1 score of 0.828 for toxic/non-toxic and 0.872 for toxicity types. The study [12] proposed a method to classify the online toxic comments using logistic regression and neural network models. Online toxic comments classification dataset is taken from Kaggle and logistic regression (LR), CNN, LSTM, and CNN+LSTM (2 layers of LSTM and 4 layers of CNN) are used. All models perform good but CNN+LSTM achieves 0.982 accuracy which is the highest among all the classifiers. In the same vein, the study [13] perform classification for online toxic comments using support vector machine (SVM), naive Bayes (NB), K-nearest neighbor (KNN), linear discriminant analysis (LDA), and CNN. The classification is conducted on Kaggle Wikiperida comments for toxic and non-toxic comments. CNN model achieves accuracy higher than 90% accuracy while the machine learning classifier obtains accuracy between 65% to 85%.

Due to the reported high accuracy of deep learning approaches, several researchers focus on using deep CNN and LSTM architectures for classification. For example, deep neural network architectures are used for toxic comments classification in [14]. The study uses NB, LSTM, and RNN to identify toxic comments. For this purpose, a toxic comment classification challenge dataset comprising 159,000 comments is used. LSTM performs best with 67% true positive rate which is 20% higher than the NB model. On the other hand, LSTM achieved a 73% F1 score, 81% precision score, and 66% recall. Similarly, hybrid deep learning approaches are adopted in [15] for the same task. For this purpose, the Jigsaw toxic comments classification dataset is used. The hybrid deep learning achieved 98% accuracy and 80% F1 score. Another study [16] created their dataset taking comments from Facebook pages and labeled them with six categories: toxic, severe toxic, obscene, threat, insult, identity hate. Different machine learning and deep learning algorithms are applied for Bangla toxic comments classification. SVM, Gaussian NB, Multinomial NB, Multi-Label k Nearest Neighbor (MLKNN), and Backpropagation for Multi-Label Neighbor (BP-MLL) are used to classify comments. BP-MLL outperforms both machine learning and deep learning algorithms used for experiments.

The study [17] proposed a methodology for the classi-

fication of toxic comments and depth error analysis. The study uses two datasets including the Wikipedia talk pages and a Twitter dataset, containing six classes of toxic comments. The study uses CNN, LSTM, bidirectional LSTM, bidirectional GRU, bidirectional GRU attention, and LR for the classification. For feature extraction, GloVe is applied for word embedding and fastText for sub-word embedding. Deep learning models are trained and tested with both GloVe and fastText tools, while the LR is used with char-n-grams and word-n-grams. The ensemble classifier achieves a 79.1% F1 score on the Wikipedia dataset and 79.3% on the Twitter dataset. In the study, [18] deep neural network architectures are used to perform toxic comment classification. CNN, bidirectional LSTM, and bidirectional GRU are used for classification where the bidirectional GRU performs the best. Another study [19] proposed a methodology to establish lexical baselines for classification by applying supervised classification methods. A 78% accuracy is achieved for the three-class task of classifying hate, offensive, and Ok using n-Gram and linear SVM. DNN based twitter hate speech detection is proposed in [20]. This study created its dataset from 6 publicly available datasets. For classification of hate speech on Twitter, DNN, and a combined model of CNN and GRU are used. The combined model of GRU+CNN is optimized with a dropout and pooling layer (1D max pooling, softmax poling, and global max pooling). The proposed model achieves an accuracy of 91.4% and 92.1% on two different datasets.

Another study [21] developed a model for automatically identifying the comments from social media as toxic or non-toxic. TF-IDF has used the feature selection and a multi-headed model using logistic regression is used for the classification. Toxic comments are further categorized into severe toxic, obscene, threat, insult, and identity-hate. Results are pretty good, however, the research provides the result for training accuracy only. In the same way, the study [22] classify toxic comments by using multi-class and multi-label word embedding techniques. For feature selection BOW, and TF-IDF are used and the GloVe, Google news dataset, SNAp, FastText, and Dranziera word embedding techniques are used for multi-label and multi-class words. The highest accuracy and AUC are achieved as 0.83 and 0.89 for per label accuracy using TF-IDF features.

Features analysis is an important part of toxic comment classification and several tend to perform analysis on the influence of various feature selection methods on classification accuracy. For example, Twitter hate-speech text classification is done using CNN in [23]. The study uses a Twitter dataset that contains four categories including racism, sexism, both (racism and sexism), and non-hate-speech. In this study, CNN is used with four techniques including random vectors, word2vec, character n-grams, and word2vec+character n-grams. Test results with 10-fold cross-validation show a 78.3% F score with word2vec embeddings. The role of feature selection on the text classification is analyzed in [24], where two machine learning algorithms including Naive Bayes (NB) and KNN. Experiments are performed with several feature extraction approaches such as information gain, mutual information, etc. and results indicate the better performance of NB with most of the used features. Dimensionality reduction plays an important role in enhancing the classification accuracy of machine learning classifiers. The impact of feature dimensions is analyzed through Chi-square, GSS coefficient, odds ratio, NGL coefficient, information gain, relevancy score, and multi-set of features with NB and KNN in [25]. Similarly, the use of multi-viewpoints cosine-based similarity visual assessment tendency is made in [26] to handle the scalability issues in data clustering from social media.

Abusive language is detected by [27] using machine learning approaches. This study is based on online comments classification which is collected from Yahoo! Finance, and News. Three datasets including the primary dataset, temporal dataset, and WWW2015 dataset are used for this purpose. Features are divided into four classes: n-grams, linguistic, syntactic, and distributional semantics for the experiments. Experiments indicate that when features are combined the classification accuracy is high. Emotional states are used to classify hate speech from social media comments in the approach proposed in [28]. The emotional states of joy, anger, sad, surprise, fear, trust, disgust and anticipation are used for this purpose. Annotated hate speech dataset is used to detect hate speech with the lexicon-based approach. The proposed approach achieves an accuracy of 80.56%.

A project called 'Perspective' is launched by Google and jigsaw which uses machine learning techniques to automatically detect online abusive, insulted, and harassment comments. Perspective is a toxic detector API on Google that filters the comments on news websites to identify abuse and harassment. An attack strategy is proposed in [29] to deceive Perspective by modifying the toxic phrase to significantly lower the toxic score assigned by the Perspective. Research indicates that the use of white spaces, redundant characters, and full stops can substantially cheat the Perspective to lower the toxic score.

Despite the high accuracy for toxic sentiment classification, the above-cited research works have several limitations. For example, the studies often use imbalanced datasets, and consequently, the reported accuracy is higher than the F1 score. F1 score is preferred on imbalanced datasets which is very low in the discussed research works. The machine learning algorithms can be overfitted for the majority class on highly imbalanced datasets. Previous studies do not focus on balancing the datasets and their results may be biased due to the model's overfitting. Predominantly, the proposed approaches follow deep learning models that are data-intensive and the accuracy is affected when used with smaller datasets. Hence, this study leverages the machine learning algorithms to perform toxic comment detection and overcomes the mentioned issues.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3083638, IEEE Access

III. MATERIALS AND METHODS

This study uses different techniques, methods, and tools for the classification of toxic and non-toxic comments. Also, various preprocessing steps, data re-sampling methods, features extraction techniques, and supervised machine learning models are adopted for the said task.

A. DATA DESCRIPTION

This study aims at the automatic classification of toxic and non-toxic comments from social media platforms. Various machine learning models are utilized for this purpose to evaluate their strength for the said task. For evaluation, the selected models are trained and tested with binary class datasets. Traditionally, toxic comments are grouped under several classes such as hate, toxic, threat, severe toxic, obscene, insult and non-toxic, etc. We follow a different approach by grouping the comments under two classes, toxic and non-toxic. The original dataset which is taken from Kaggle [30], is a multi-label dataset and contains labels such as toxic, severe_toxic, obscene, threat, insult, and identity_hate. The non-toxic comments belong to one class, while from the other comments only those comments are selected that have toxic labels. It means that the comments that label severe_toxic, obscene, threat, insult, and identity_hate are not selected. For example, Table 1 shows that 'comment 2' is only toxic and 'comment 3' is non-toxic. For our experiment, both 'comment 2' and 'comment 1' are selected under toxic and no-toxic classes, but 'comment 1' and 'comment 4' are not selected.

TABLE 1: Example of various classes in the original dataset.

Comment_text	Comment 1	Comment 2	Comment 3	Comment 4
identity_hate	0.0	0.0	0.0	0.0
insult	1.0	0.0	0.0	0.0
obscene	1.0	0.0	0.0	0.0
severe_toxic	0.0	0.0	0.0	0.0
threat	0.0	0.0	0.0	0.0
toxic	0.0	1.0	0.0	0.0
toxicity	0.0	0.0	0.0	1.0

We extract only toxic and non-toxic comments from the dataset and Table 2 shows the ratio of toxic and non-toxic comments in the dataset used for experiments. The ratio of toxic and non-toxic comments in the dataset is not equal which shows the imbalanced data problem. The performance of the classifiers could be affected due to an imbalanced dataset.

TABLE 2: Number of records for toxic and non-toxic comments.

Category	No. of comments	Experimental Data
Non-Toxic	143346	70000
Toxic	15294	15294
Total	158640	85294

The dataset contains 158,640 comments in total with toxic comments having the lowest ratio in the dataset, i.e., 15,294 while non-toxic comments are 143,346. It makes a huge difference and makes the dataset highly imbalanced. Due to the large size of the dataset, only 70,000 non-toxic comments are randomly selected for the experiments.

B. PREPROCESSING STEPS

Pre-processing techniques are applied to clean the data which helps to improve the learning efficiency of machine learning models [31]. For this purpose, the following steps are executed in the given sequence.

Tokenization: is a process of dividing a text into smaller units called 'tokens'. A token can be a number, word, or any type of symbol that contains all the important information about the data without conceding its security.

Punctuation removal: involves removing the punctuation from comments using natural language processing techniques. Punctuations are the symbols that are utilized in sentences/comments to make the sentence clear and readable for humans. However, it creates problems in the learning process of machine learning algorithms and needs to be removed to improve their learning process. Some common punctuation marks are mostly used such that colon, question marks, comma, semicolon, full-stop/period, etc.... ?:,;.[]() [32].

Number removal: is also a part of preprocessing which helps to improve the performance of the machine learning algorithms. Numbers are unnecessary and do not contribute to the learning of text analysis approaches. Removing the numbers increases the efficiency of models and decreases the complexity of the data.

Stemming: is an important part of preprocessing because it increases the performance by clarifying affixes from sentences/comments and converting the comments into the original form. Stemming is the process of transforming a word into its root form. For example, different words have the same meaning such as: 'plays', 'playing', 'played' are modified forms of 'play'. Stemming is implemented using the Porter stemmer algorithms [33].

Spelling correction: is the process of correcting the misspelled words. In this phase, the spelling checker is used to check the misspelled words and replace them with the correct word. Python library 'pyspellchecker' provides the necessary features to check the misspelled words and is used for the experiments [34].

Stopwords removal: Stopwords are those English words that do not add any meaning to a sentence. So these can be removed by stopwords removal without affecting the meaning of a sentence. The removal of stop-words increases the model's performances and decreases the complexity of input features [35].

C. FEATURE ENGINEERING

Feature engineering aims at discovering useful data features or constructing features from original features to train machine learning algorithms effectively [36]. The study [37] concludes that feature engineering can improve the efficiency of machine learning algorithms. 'Garbage out' is a corporate

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3083638, IEEE Access

proverb used in machine learning which implies that senseless data used as the input, yields meaningless output. In contrast, more information-driven data will yield favorable results. Hence, feature engineering can derive useful features from raw data which helps to improve the reliability and accurateness of learning algorithms. In the proposed methodology, two feature engineering methods are used including the bag of words and term frequency-inverse document frequency.

D. BAG-OF-WORDS

The bag of words (BoW) technique is used to extract features from the text data. The boW is easy to implement and understand besides being the simplest method to extract features from the text data. The boW is very suitable and useful for language modeling and text classification. The 'CountVectorizer' library is used to implement BoW. CountVectorizer calculates the occurrence of words and constructs a spare database matrix of words [38]. The boW is a pool of words or features, where every feature is categorized as a label that signifies the occurrences of the categorized feature.

E. TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY

TF stands for term frequency and IDF stands for inverse document frequency of the word. The TF-IDF is a statistical analysis that is used to determine how many relevant words are in a list or corpus. The value increases with the number of times a word is shown in the text but is normalized by the word occurrence in the document [39].

- Term Frequency (TF): is the frequency of a term given in the text of a document. Because each document is dissimilar in size, it is likely that in long documents a word will occur more often than the shorter ones. To normalize, the term frequency is also divided by the length of the text.

$$F(t) = \frac{\text{No. of times } t \text{ appears in a document}}{\text{Total no. of terms in the document}} \quad (1)$$

- Inverse Document Frequency (IDF): is a rating of how infrequent the term is in a given document. IDF indicates the importance of a word on account of its rareness. The rare words have a higher IDF score.

$$IDF(t) = log_e \frac{\text{Total no. of documents}}{\text{No. of documents with term } t \text{ in it}} \quad (2)$$

TF-IDF is then calculated using both TF and IDF using

$$TF - IDF = TF_{t,d} * log \frac{N}{D_f}, \quad (3)$$

where the $TF_{t,d}$ is frequency of term t in document d.

F. DATA RE-SAMPLING TECHNIQUES

Data re-sampling techniques are used to solve imbalanced dataset problems. The imbalanced dataset contains an unequal ratio of the target classes and can cause problems in classification tasks because models can over-fit on the majority class [40]. To solve this problem different data re-sampling techniques have been presented. In this study, two types of re-sampling techniques are used including under-sampling and over-sampling.

1) Random Under-Sampling

Under-sampling reduce the size of the dataset by deleting example of the majority class. For the under-sampling, a random under-sampling approach is used in the current study. In the random under-sampling, the major class examples are rejected at random and deleted to balance the distribution of the target classes. Simply we can say that under-sampling aims to balance class distribution by randomly deleting majority class examples. The random under-sampling technique is one of the widely used re-sampling approaches and selected due to its reported performance [41]–[44].

2) Synthetic Minority Over-Sampling Technique

Over-sampling is a technique in which the number of samples of the minority class is increased in the ratio of the majority class. Over-sampling increases the size of data which generates more features for model training and could be helpful to increase the accuracy of the model. In this study synthetic minority over-sampling technique (SMOTE) is used for over-sampling. SMOTE is a state-of-art technique that was proposed in [45] to solve the overfitting problem for imbalanced datasets. SMOTE randomly picks up the smaller class and finds the K-nearest neighbors of each smaller class. The picked samples are evaluated using the K-nearest neighbor for that particular point to construct a new minority class. SMOTE is adopted on account of the results reported in [46]–[49].

G. PROPOSED METHODOLOGY

Ensemble learning is widely used to attain high accuracy for classification tasks. The combination of various models can perform well as compared to individual models. Owing to the high accuracy of ensemble models, this study leverage an ensemble model to perform toxic comments classification. Our experiments indicate the good performance from LR and SVC, so to further improve the performance, this study combines these models. The proposed approach is called regression vector voting classifier (RVVC) and combines these models using soft voting criteria as shown in Figure 2. The soft voting criteria ensure that the class with a high predicted probability by two classifiers will be considered as the final prediction.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3083638, IEEE Access

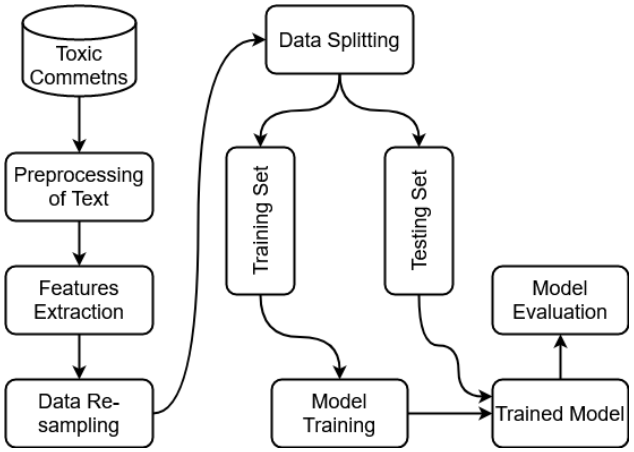


FIGURE 1: The flow of the proposed methodology.

Algorithm 1 Algorithm for toxic comments classification

Input: Corpus-text comments
Output: Class-Toxic or Non-Toxic

TLR→ Trained LR
TSVC→ Trained SVC
for i in Corpus **do**
  $ToxicPob_{LR} \rightarrow TLR(i)$
  $NonToxicPob_{LR} \rightarrow TLR(i)$
  $ToxicPob_{SVC} \rightarrow TSVC(i)$
  $NonToxicPob_{SVC} \rightarrow TSVC(i)$
  $RVCC_{Pred} \rightarrow argmax((ToxicPob_{LR} + ToxicPob_{SVC})/2, (NonToxicPob_{LR} + NonToxicPob_{SVC})/2)$
end for
$Toxic|Non-Toxic \rightarrow RVVC$ prediction

Algorithm 1 shows the working of the proposed RVVC models and explains how it combines the LR and SVC for toxic comment classification. Let LR and SVC be the two models and 'toxic' and 'non-toxic' be the two classes, then the prediction can be made using the following equation

$$RVVC = argmax\{Toxic_{prob}, NonToxic_{prob}\} \quad (4)$$

where argmax is used in machine learning for finding the class with the largest predicted probability. The $Toxic_{prob}$ and $NonToxic_{prob}$ indicate the joint probability of toxic and non-toxic classes by the LR and SVC models and are calculated as follows

$$Toxic_{prob} = \frac{ToxicProb_{LR} + ToxicProb_{SVC}}{2} \quad (5)$$

$$NonToxic_{prob} = \frac{NonToxicProb_{LR} + NonToxicProb_{SVC}}{2} \quad (6)$$

where $ToxicProb_{LR}$, and $ToxicProb_{SVC}$ are the probability for toxic class by LR and SVC, respectively while $NonToxicProb_{LR}$, and $NonToxicProb_{SVC}$ are the probability scores for the non-toxic class by LR and SVC, respectively.

To illustrate the working of the proposed RVVC, the values for one sample are taken from the dataset used for the experiments. LR and SVC given probabilities for the sample data are

- $ToxicProb_{LR}$ = 0.6
- $NonToxicProb_{LR}$ = 0.4
- $ToxicrPob_{SVC}$ = 0.5
- $NonToxicProb_{SVC}$ = 0.5

The combined $Toxic_{prob}$ and $NonToxic_{prob}$ are calculated as follows

$$Toxic_{prob} = \frac{0.6 + 0.5}{2} \quad (7)$$

$$NonToxic_{prob} = \frac{0.4 + 0.5}{2} \quad (8)$$

Then argmax function is applied to select the class with the higher probability. Here the largest prediction probability is for the Toxic class so the final prediction by RVVC will be Toxic class.

$$RVVC = argmax\{0.55, 0.45\} \quad (9)$$

The flow of the proposed methodology is shown in Figure 1. In the proposed methodology, the toxic comment classification problem is solved using LR and SVC. For classification, the dataset is obtained from Kaggle [30] which contains toxic comments. Several preprocessing steps are carried out on the dataset to clean the data. After data cleaning, two feature extraction approaches including TF-IDF and BoW are applied.

Owing to the higher difference in the number of samples for toxic and non-toxic classes, various re-sampling approaches area applied. Random undersampling and SMOTE oversampling approaches are leveraged to balance the dataset and improve the performance of the proposed methodology. The ratio of the number of samples after re-sampling is given in Table 3.

TABLE 3: Number of samples after applying re-sampling.

Category	Count	Exp.Data	Under-sampling	Over-sampling
Non-Toxic	143346	70000	15294	70000
Toxic	15294	15294	15294	70000
Total	158640	85294	30588	140000

In under-sampling random samples of the majority class are removed while in over-sampling, the samples of the minority class are generated using SMOTE. After re-sampling, the data is split into training and testing sets with a 75:25 ratio. The number of training and testing samples after data split is given in Table 4.

We used the training set to train the machine learning models and the proposed ensemble classifier on extracted features and evaluate the performance of machine learning

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3083638, IEEE Access

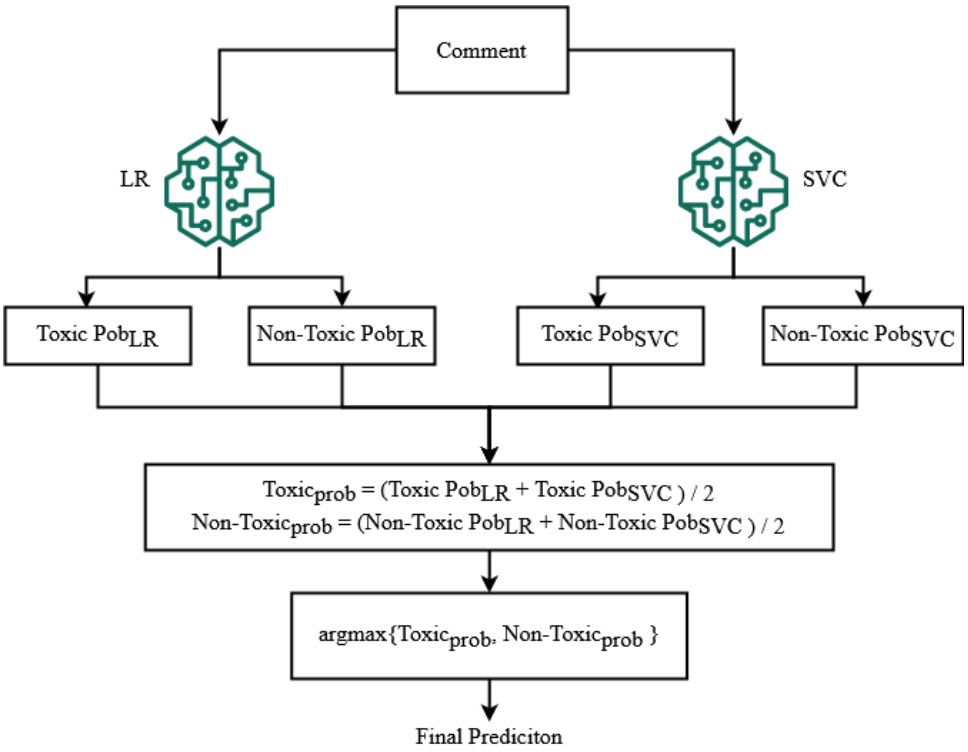


FIGURE 2: Architecture of the proposed ensemble model RVVC.

TABLE 4: Number of samples for train and test data.

Re-sampling	Set	Toxic	Non-Toxic	Total
Without re-sampling	Training	11520	52450	63970
	Testing	3774	17550	21324
	Total	15294	70000	85294
After Under-sampling	Training	11511	11430	22941
	Testing	3783	3864	7647
	Total	15294	15294	30588
After Over-sampling	Training	52342	52658	105000
	Testing	17658	17342	35000
	Total	70000	70000	140000

TABLE 5: Machine Learning Models Parameters

Algorithm	Hyper parameters
RF	n_estimators=300, random_state=5, max_depth=100
GBM	n_estimators=100, max_depth=100
LR	C=1.0, max_iter=100, penalty='l2'
SVM	kernel='linear', C=2.0, random_state=500
KNN	algorithm='auto', leaf_size=30, n_neighbors=3

models on the test data. For performance evaluation, various metrics including accuracy, precision, recall, and F1 score are used.

H. SUPERVISED MACHINE LEARNING MODELS

Various machine learning models are adopted to perform toxic comments classification. Machine learning algorithms are implemented using the Scikit-learn library. We used two tree-based models such as RF, GBM, two linear models LR, SVM, and one non-parametric model KNN. The hyper-parameters of all the machine learning models are given in Table 5.

1) Support Vector Machine

SVM is a supervised machine learning model used for both classification and regression problems. The straightforward approach to classifying the data starts by constructing a function that divides the data points into consistent labels with (a) the least amount of errors possible or (b) the highest possible margin. That is because larger empty areas next to the splitting function contribute to fewer errors. After the function is constructed, the labels are better separated from each other. Hyperparameters of SVM are listed in Table 5 in which the kernel='linear' specifies the kernel type used for SVM. The linear kernel is used to ensure high accuracy and reduced time complexity. The term C=2.0 is used as the regularization parameter and the strength of the regularization is inversely proportional to C. The parameter random_state = 500 is used for the seed of the pseudo-random number which is used for likelihood calculations when shuffling the results.

2) Random Forest

RF is a tree-based ensemble classifier, which generates predictions that are extremely accurate by combining several poor apprentices (weak learners). RF uses bootstrap bagging to train a variety of decision trees using various bootstrap samples. In RF, a bootstrap sample is produced by subsampling the training data set, where the size of a sample dataset

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3083638, IEEE Access

and the training dataset sample are the same. RF and other ensemble classifiers utilize decision trees for the prediction using the decision trees. The identification of the attribute for the root node at each stage is a major challenge for constructing the decision trees.

$$p = mode\{T_1(y), T_2(y), ..., T_m(y)\} \tag{10}$$

$$p = mode\{\sum_{m=1}^{m}[T_m(y)]\} \tag{11}$$

where p is the final decision of the decision trees by majority vote, while $T_1(y)$, $T_2(y)$, $T_3(y)$, and $T_m(y)$ are the number of decision trees involved in the prediction process.

To improve the accuracy, RF was implemented with n as 100 which indicates the number of trees that contribute to the prediction in an RF. The 'max_depth' is set to 60 which shows the every decision tree can go to a maximum depth of 60 levels. By specifying the depth point, the 'max_depth' parameter decreases uncertainty in the decision tree and decreases the probability of the decision tree over-fitting. The parameter 'random state' is used for the randomness of the samples during the training. For our experiments, we attain good results with RF by using only two hyperparameters.

3) Gradient Boosting Machine

Gradient boosting classifiers is a collection of algorithms for machine learning that combine several weak learners to construct a strong prediction model. A loss function relies on the GBM and a customized loss function can also be used. The GBM supports several generic loss functions, but the loss mechanism has to be differentiable. Classification algorithms also use logarithmic loss, while squared errors can be used in regression algorithms. Every time the boosting algorithm is implemented, the gradient boosting system does not need to derive a new loss function, rather any differentiable loss function can be applied to the system. Several hyperparameters are tuned to get good accuracy from the GBM. For example, n is set to 100 indicating the number of trees which contribute to the prediction. Equipped with 100 decision trees, the final prediction is made by voting all predictions of the decision trees. Value of 'max depth is used 60 allowing a decision tree to a maximum depth of 60 levels.

4) Logistic Regression

Logistic regression is one of the most widely used approaches for binary classification problems. LR is known for the method that it uses, i.e., the logistic equation also called the sigmoid function. The sigmoid function is an S-shaped curve that can take any evaluated number and maps it to a value between 0 and 1 [50].

$$\frac{1}{(1 + e^{-}value)} \tag{12}$$

where e is the base of the normal logarithms and value is the real numerical value that is to be converted. Below is a plot of numbers between -5 and 5, transformed by the logistic function into ranges 0 and 1.

$$y = \frac{e^{(b0+b1*x)}}{(1 + e^{(b0+b1*x)})} \tag{13}$$

where $b0$ is the bias or intercept, y is the expected performance and $b1$ is the coefficient for the single input value x. Every column of the input data has a coefficient b correlated with it (a constant actual value) to be learned from the training data.

To attain high accuracy, LR is used with 100 'max_iter' for the solvers to converge. The parameter 'penalty' is set to 'l2' which is used to specify the norm used in the penalization. The parameter C =1.0 is used to specify the inverse of the regularization strength.

5) K-Nearest Neighbor

KNN is one of the simplest supervised classification methods in machine learning. The KNN identifies the similarities between the new data and existing cases and puts the new data in the group with high similarity. The similarity is calculated using distance calculation between the new data and the existing classes. For distance measurement, various distance estimation methods are used such as Euclidean, Manhattan, and Cityblock, etc. KNN algorithm can be used for both regression and classification, but it is mainly used for classification problems. KNN is a non-parametric algorithm, implying that it considered no inference to the underlying data. KNN has multiple parameters that can be refined to achieve high accuracy. For the current study, leaf size is set to 30 which is passed to the ball tree or KD Tree. The optimal value depends on the nature of the problem. Minkowski is used as the distance metric while the number of the neighbor is set to 3.

I. EVALUATION METRICS

We evaluate the performance of machine learning models in terms of accuracy, precision, recall, and F1 score.

1) Accuracy

Accuracy indicates the ratio of correct predictions to the total predictions from the classifiers on test data. The maximum accuracy score is 1 indicating that all predictions from the classifier are correct while the minimum accuracy score can be 0. Accuracy can be calculated as

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}, \tag{14}$$

Another form to calculated accuracy is using

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{15}$$

where TP is a true positive, TN is a true negative, FP is a false positive and FN is a false negative.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3083638, IEEE Access

2) Precision

Precision is also known as a positive predictive value and represents the relative number of correctly classified instances among all true classified instances. A precision value of 1 means that every instance of data that is categorized as positive which is positive. It is important to note, however, that this does not influence the number of positive instances with the label negative which are predicted as positive.

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

3) Recall

Recall often called sensitivity represents the relative number of positive classified instances from all positive instances. The recall is defined as

$$Recall = \frac{TP}{TP + FN} \tag{17}$$

4) F1 Score

Precision and recall are not regarded as true representers of the performance of a classifier individually. F1 has been deemed more important as it combines both precision and recall and gives a score between 0 and 1. It is the harmonic mean of precision and recall and calculated using

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{18}$$

IV. RESULTS AND DISCUSSION

Several experiments are performed to evaluate the performance of both the selected machine learning classifiers, as well as, the proposed RVVC ensemble classifier. The experiments are divided into three categories: experiments without re-sampling, experiments with under-sampling, and experiments with over-sampling.

A. PERFORMANCE OF MACHINE LEARNING MODELS ON IMBALANCED DATASET

Initial experiments are performed using the original imbalanced dataset with TF-IDF and BoW separately. Tables 6 shows the values of performance evaluation metrics on the imbalanced dataset using TF-IDF features. There is a lot of fluctuation in the values of evaluation parameters. For example, the accuracy of RF using TF-IDF is 0.92 but the F1 score is 0.83. The difference in the values of accuracy and F1 score is similar for other machine learning models.

TABLE 6: Performance results of all models on imbalanced dataset using TF-IDF.

Classifier	Accuracy	Precision	Recall	F1 score
RF	0.92	0.94	0.78	0.83
SVC	**0.94**	0.90	0.87	0.89
KNN	0.86	0.89	0.61	0.64
DT	0.91	0.85	0.86	0.86
LR	0.93	0.93	0.84	0.88
RVVC	**0.94**	0.92	0.86	0.89

Table 7 shows the results for machine learning models when trained and tested using the BoW features on the imbalanced dataset. Results indicate that the models get overfitted on the majority class data because the models get more data from the majority class as compared to the minority class. Consequently, the number of wrong predictions for the minority class is higher than the majority class.

TABLE 7: Performance results of all models on imbalanced dataset using BoW.

Classifier	Accuracy	Precision	Recall	F1 score
RF	0.92	0.94	0.78	0.83
SVC	0.92	0.87	0.87	0.87
KNN	0.89	0.86	0.74	0.78
DT	0.91	0.84	0.85	0.85
LR	**0.94**	0.91	0.87	0.89
RVVC	0.93	0.91	0.85	0.88

Owing to the high difference in the values of accuracy and F1 score, correct predictions (CP) and wrong predictions (WP) are important evaluation parameters to be analyzed. Table 8 shows the TP, TN, FP, CP, and WP for both TF-IDF and BoW for all the classifiers. Results show that RVVC gives the highest number of correct predictions, i.e., 20,007 out of 21,324 total predestines, and gives only 1,317 wrong predictions with TF-IDF. RVVC performs somehow better than other classifiers, on the imbalanced dataset because of its ensemble architecture. The ensemble architecture model can perform better than individual models. Using the soft voting criteria on the predictions from two well-performing models increases the probability of correct prediction.

TABLE 8: Correct and wrong predictioins from all classifiers on the imbalanced dataset.

Feature	Classifier	TP	TN	FP	FN	CP	WP
TF-IDF	RF	17475	2093	75	1681	19568	1756
	SVC	17032	2931	518	843	19963	1361
	KNN	17472	832	78	2942	18304	3020
	DT	16597	2912	953	862	19509	1815
	LR	17309	2618	241	1156	19927	1397
	RVVC	17243	2764	307	1010	**20007**	1317
BoW	RF	17449	2155	101	1619	19604	1720
	SVC	16732	2975	818	799	19707	1617
	KNN	17111	1912	439	1862	19023	2309
	DT	16525	2885	1025	889	19410	1914
	LR	17137	2848	413	926	**19985**	1339
	RVVC	17191	2740	359	1034	**19931**	1393

B. PERFORMANCE OF MODELS ON BALANCED DATASET USING UNDER-SAMPLING

Further experiments are performed using a balanced dataset with a random under-sampling technique. Results using TF-IDF features on the under-sampled data are shown in Table 9. Results suggest that the performance of the selected models has been degraded on the under-sampled dataset. As under-sampling reduces the size of the dataset, the number of features to train the models is also reduced which affects the accuracy of machine learning models. It is observed that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3083638, IEEE Access

the difference in the values of accuracy and other evaluation parameters has been reduced and the values for accuracy and F1 are similar now. It indicates the good fit of machine learning models. RVVC model outperforms all other models in terms of accuracy, precision, recall, and F1 score when used with TF-IDF feature from the under-sampled dataset. It achieves the highest accuracy and F1 with a value of 0.91 each and performs better than all other classifiers.

TABLE 9: Performance results of all models using TF-IDF features from under-sampled dataset

Classifier	Accuracy	Precision	Recall	F1 score
RF	0.88	0.89	0.88	0.88
SVC	0.90	0.90	0.90	0.90
KNN	0.59	0.75	0.59	0.51
DT	0.86	0.86	0.86	0.86
LR	0.90	0.90	0.90	0.90
RVVC	**0.91**	0.91	0.91	0.91

Table 10 shows the performance of machine learning models after under-sampling with the BoW features. This performance shows that the ensemble model can also perform well on a small dataset resulting from the under-sampling. The proposed RVVC model is a combination of LR and RF which is a good combination for small and large datasets. RVVC performs better with BoW features in the under-sampling case with 0.91 accuracy which is the highest of all the classifiers.

TABLE 10: Performance results of all models using BoW features from under-sampled data.

Classifier	Accuracy	Precision	Recall	F1 score
RF	0.89	0.89	0.89	0.89
SVC	0.89	0.89	0.89	0.89
KNN	0.76	0.77	0.76	0.76
DT	0.86	0.86	0.86	0.86
LR	0.90	0.90	0.90	0.90
RVVC	**0.91**	0.91	0.91	0.91

Table 11 shows the performance of machine learning models in terms of correct and wrong predictions. In both TF-IDF and BoW cases, the RVVC model outperforms all other models in terms of correct predictions. RVVC gives 6,927 correct predictions with TF-IDF features and gives only 720 wrong predictions as compared to LR which is the 2nd highest performer and gives 748 wrong predictions. Similarly, with BoW features, RVVC gives 6,940 correct predictions and 707 wrong predictions which is also the lowest number for any model. In light of these results, RVVC shows the highest performance among all the machine learning classifiers.

C. EXPERIMENT RESULTS OF MODELS USING SMOTE OVER-SAMPLED DATA

Experiments are performed using the SMOTE balanced dataset both with TF-IDF and BoW features. Table 12 shows the performance of all models with TF-IDF features. The performance of machine learning models has improved significantly when trained on TF-IDF features from the SMOTE

TABLE 11: Number of correct and wrong predictions using the TF-IDF and BoW features from the under-sampled dataset.

Feature	Classifier	TP	TN	FP	FN	CP	WP
TF-IDF	RF	3601	3152	263	631	6753	894
	SVC	3519	3378	345	405	6897	750
	KNN	3829	699	35	3084	4528	3119
	DT	3286	3269	578	514	6555	1092
	LR	3605	3294	259	489	6899	748
	RVVC	3586	3341	278	442	**6927**	720
BoW	RF	3622	3169	242	614	6791	865
	SVC	3454	3346	410	437	6800	847
	KNN	3339	2502	525	1281	5841	1806
	DT	3302	3238	562	545	6540	1107
	LR	3554	3360	310	423	6905	733
	RVVC	3594	3346	270	437	**6940**	707

over-sampled dataset. Over-sampling increases the dataset size which increases the number of features for training the models. Consequently, it helps to a good fit of models and increases their performance. However, at the same time the performance of the KNN model, which does not perform well on large features set, has degraded. Among all models, RVVC achieves the highest accuracy of 0.97 and outperforms all other models when trained on TF-IDF features.

TABLE 12: Performance results of all models using TF-IDF features from over-sampled dataset.

Classifier	Accuracy	Precision	Recall	F1 score
RF	0.96	0.96	0.96	0.96
SVC	0.95	0.95	0.94	0.94
KNN	0.55	0.75	0.55	0.44
DT	0.94	0.94	0.94	0.94
LR	0.94	0.94	0.94	0.94
RVVC	**0.97**	0.97	0.97	0.97

Table 13 shows the performance of the models with BoW features from the over-sampled dataset. RVVC performs well on BoW features as well and achieves a joint accuracy of 0.93 with LR. However, its recall score is higher than LR which shows its superior performance. KNN models show poor performance among all the classifiers with an accuracy of 0.64 while RF and SVC perform well with 0.90 and 0.91 accuracies, respectively. As a whole, the performance of all the models has been reduced with BoW features than that of TF-IDF features.

TABLE 13: Performance results of all models on over-sampled data using BoW features.

Classifier	Accuracy	Precision	Recall	F1 score
RF	0.90	0.90	0.90	0.90
SVC	0.91	0.92	0.91	0.91
KNN	0.64	0.78	0.64	0.58
DT	0.87	0.88	0.87	0.87
LR	**0.93**	0.93	0.92	0.93
RVVC	**0.93**	0.93	0.93	0.93

TF-IDF shows superior performance than the BoW feature for toxic comments classification. It is important to point out

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3083638, IEEE Access

that boW contains only frequency (count) of word occurrence for a given comment and does not record any information regarding the importance of a word. For BoW, no word is a rare or common word, it just counts how many times it has appeared in a given comment. On the other hand, TF-IDF counts the occurrence of a word and its importance. As a result, it performs better than BoW. Besides, with the increase in the size of comments, the size of the vocabulary also increases, which leads to sparsity in BoW. The increased size of the training vector affects the performance of the classifiers and degrades the accuracy.

The performance of machine learning models using SMOTE technique is also evaluated in terms of correct and wrong predictions as shown in Table 14. Results suggest that RVVC gives the highest number of correct predictions when used with TF-IDF features from SMOTE over-sampled dataset. RVVC gives 33,857 correct predictions out of 35,000 predictions and only 1,143 predictions are wrong. RF and SVC are behind the RVVC model with 33,552 and 33,076 correct predictions, respectively.

TABLE 14: Number of correct and wrong predictions using SMOTE over-sampled dataset.

Feature	Classi.	TP	TN	FP	FN	CP	WP
	RF	16840	16712	569	879	33552	1448
	SVC	15851	17225	1491	433	33076	1924
	KNN	1728	17620	15614	38	19348	15652
TF-IDF	DT	16006	16837	1336	821	32843	2157
	LR	16323	16658	1019	1000	32981	2019
	RVVC	16701	17156	641	502	**33857**	1143
	RF	14980	16498	2362	1160	31478	3522
	SVC	15130	16837	2212	821	31967	3033
	KNN	4825	17546	12517	112	22371	12629
BoW	DT	13917	16582	3425	1076	30499	4501
	LR	15516	16863	1826	795	32379	2621
	RVVC	15800	16636	1542	1022	**32436**	2564

Table 9 and 10 contain the results using the random under-sampling technique with TF-IDF and BoW features, respectively, and Tables 12 and 13 contain the results with SMOTE. The performance of models with oversampling technique is significant in comparison to under-sampling. In random under-sampling random data are deleted from the majority class to balance the samples for the majority and minority class. As a result, the size of data, as well as, the size of the feature set is reduced. When trained on a small feature vector, the performance of the machine learning models is degraded. Additionally, in deleting procedure of random under-sampling, many important records that can be influential in models' training may be deleted which affects the performance of the models. On the other hand, oversampling increases the size of data by generating new records which help to generate a large feature set and improves the performance of learning models.

D. EXPERIMENTAL RESULTS USING RNN WITH OVER-SAMPLED, UNDER-SAMPLING, AND IMBALANCED DATASET

Along with the machine learning models and proposed RVVC, a recurrent neural network is also tested for toxic comment classification. Deep learning approaches tend to show higher performance for text classification tasks [51]. The RNN is used with the architecture given in Figure 3.

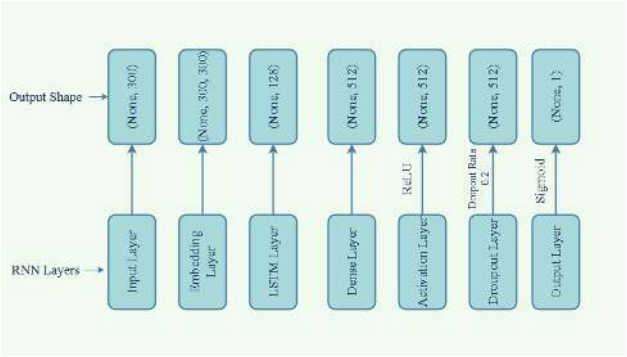


FIGURE 3: The architecture of the used recurrent neural network for toxic comments.

Experimental results to classify the toxic and non-toxic comments using RNN are given in table 15. RNN model produces the output based on previous computation by using sequential information and gives better results for this reason. The results using the SMOTE over-sampled dataset show higher accuracy due to the large feature set. However, with the under-sampled dataset, the size of the dataset is decreased which decreases the feature vector for training and the performance of RNN is reduced. RNN achieves the highest accuracy of 0.95 when trained on an over-sampled dataset while the lowest accuracy of 0.887 with the imbalanced dataset. However, RNN's highest accuracy of 0.95 is lower than the accuracy of the proposed RVVC which is 0.97 with TF-IDF features from SMOTE over-sampled dataset.

TABLE 15: Accuracy of RNN with under-sampled, over-sampled and imbalanced dataset.

Sampling	Accuracy	Loss
Under-Sampling	0.887	0.267
Over-Sampling	**0.95**	0.11
Without Sampling	0.93	0.13

E. PERFORMANCE ANALYSIS WITH STATE-OF-THE-ART APPROACHES

Performance comparison of the proposed RVVC is done with five state-of-the-art approaches including both machine and deep learning approaches for toxic comments classification. Table 16 shows the performance appraisal results for RVVC and other models. Results prove that the proposed RVVC performs better than other approaches to correctly classify the toxic and non-toxic comments.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3083638, IEEE Access

TABLE 16: Performance comparison resutls for state-of-the-art approaches and proposed RVVC.

Ref.	Year	Model	F1 score	Prec.	Recall	Acc.
[14]	2020	LSTM	73%	81%	76%	-
[15]	2021	Hybrid DL	80%	-	-	98%
[16]	2019	BPMLL	-	-	-	60%
[17]	2018	Bidir. GRU	78%	74%	87%	98%
[18]	2019	BiGRU	65%	75%	70%	-
Proposed	2021	RVVC	97%	97%	97%	**97%**

F. STATISTICAL T-TEST

To show the significance of the proposed RVVC model, a statistical significance test, a T-test has been performed. To support the T-test we suppose two hypotheses as follow:

- Null hypotheses: The proposed model RVVC is statistically significant.
- Alternative hypotheses: The Proposed model is not statistically significant.

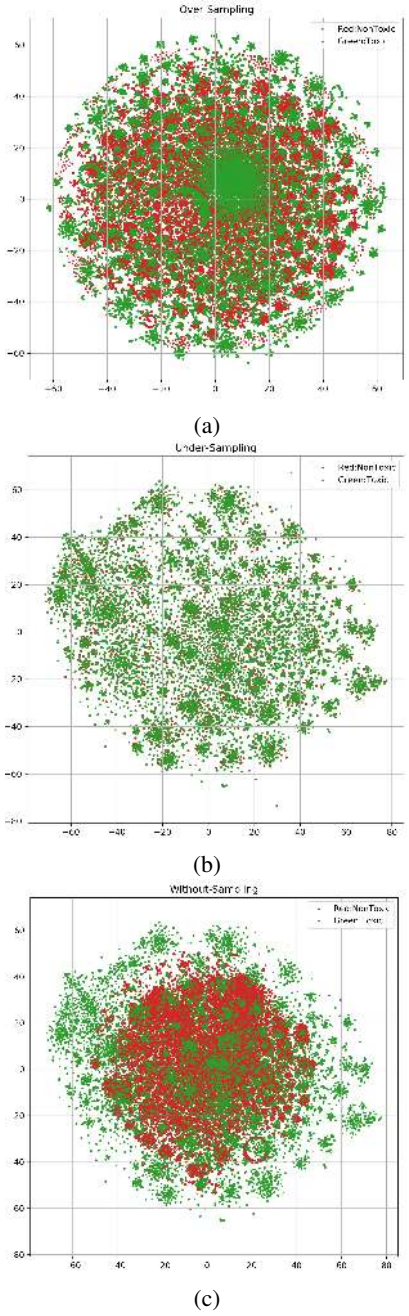
Statistical T-test results that the RVVC is statistically significant for all resampling cases. RVVC accepts null hypotheses without resampling, under-sampling, and oversampling. T-test also shows that TF-IDF gives more significance for machine learning models. Model reject null hypotheses when trained on BoW features in comparison to TF-IDF.

G. DISCUSSION

Experiments are carried out to analyze the impact of SMOTE oversampling and random under-sampling approaches on the performance of selected machine learning models and the proposed RVVC model. Experimental results signify the superior performance of machine learning models with SMOTE over-sampling. Results of applying SMOTE over-sampling and random under-sampling are shown in Figure 4 using a t-distributed stochastic neighbor embedding (TSNE) plot. Figure 4a indicates clearer and impact features generation using the SMOTE. SMOTE technique provides an equal number of features for training the models which leads to a significant increase in the performance of models.

Conversely, for the under-sampling case, although features are equal, the reduced size of the features degrades the performance of the models. Moreover, the features are too scattered for the under-sampling case, as shown in Figure 4b. As a result, the distinctiveness of comments is reduced which reduces the classification accuracy. Figure 4c shows that the data without sampling contain more data for the non-toxic class as compared to the toxic class. Additionally, toxic data is scattered and it is difficult for machine learning models to learn on the scattered dataset. So, the models give higher accuracy for the non-toxic class than the toxic class. SMOTE helps to overcome these limitations and shows higher performance than both the imbalanced dataset and under-sampled dataset for toxic comments classification.

The proposed model RVVC outperforms both the RNN and machine learning models due to its structure. RVVC is an ensemble model which is a combination of LR and SVC and uses soft voting criteria to make the final prediction. RVVC performs better than the individual models because it combines the predictions from two well-performing models including LR and SVC. Computing the probability for each class using its individual models and then finding the target class with maximum probability to make the final prediction elevates its performance as compared to the individual models. The deep architecture of RVVC makes it more accurate.



(a)

(b)

(c)

FIGURE 4: Impact of SMOTE as compared to under-sampling and without-sampling

V. CONCLUSIONS

This study analyzes the performance of various machine learning models to perform toxic comments classification and proposes an ensemble approached called RVVC. The influence of an imbalanced dataset and balanced dataset using random under-sampling and SMOTE over-sampling on the performance of the models is analyzed through extensive experiments. Two feature extraction approaches including TF-IDF and BoW are used to get the feature vector for models' training. Results indicate that models perform poorly on the imbalanced dataset while the balanced dataset tends to increase the classification accuracy. Besides the machine learning classifiers like SVM, RF, GBM, and LR, the proposed RVVC and RNN deep learning models perform well with the balanced dataset. The performance with an over-sampled dataset is better than the under-sampled dataset as the feature set is large when the data is over-sampled which elevates the performance of the models. Results suggest that balancing the data reduces the chances of models over-fitting which happens if the imbalanced dataset is used for training. Moreover, TF-IDF shows better classification accuracy for toxic comments than BoW as TF-IDF records the importance of a word contrary to BoW which simply counts the occurrence of a word. The proposed ensemble approach RVVC demonstrates its efficiency for toxic and non-toxic comments classification. The performance of RVVC is superior both with the imbalanced and balanced dataset, yet, it achieves the highest accuracy of 0.97 when used with TF-IDF features from SMOTE over-sampled dataset. The performance comparison with state-of-the-art approaches also indicates that RVVC shows better performance and proves good on small and large feature vectors. Despite the better performance of the proposed ensemble approach, its computational complexity is higher than the individual models which is an important topic for our future research. Similarly, dataset imbalance can overstate the results because data balancing using SMOTE or random under-sampling approach may have a certain influence on the reported accuracy. Moreover, we intend to perform further experiments on multi-domain datasets and run experiments on more datasets for toxic comment classification.

REFERENCES

[1] Elias Aboujaoude, Matthew W Savage, Vladan Starcevic, and Wael O Salame. Cyberbullying: Review of an old problem gone viral. Journal of adolescent health, 57(1):10–18, 2015.

[2] How Much Data is Created on the Internet Each Day? https://blog.microfocus.com/how-much-data-is-created-on-the-internet-each-day/. Accessed: 2020-06-06.

[3] World Internet Users and 2020 Population Stats. https://www.internetworldstats.com/stats.htm. Accessed: 2020-06-06.

[4] Maeve Duggan. Online harassment. Pew Research Center, 2014.

[5] Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. Deep learning for hate speech detection in tweets. In Proceedings of the 26th International Conference on World Wide Web Companion, pages 759–760, 2017.

[6] Man jailed for 35 years in Thailand for insulting monarchy on Facebook. https://www.theguardian.com/world/2017/jun/09/man-jailed-for-35-years-in-thailand-for-insulting-monarchy-on-facebook. Accessed: 2020-06-06.

[7] Mississippi teacher fired after racist Facebook post; black parent responds. https://www.clarionledger.com/story/news/2017/09/20/mississippi-teacher-fired-after-racist-facebook-post/684264001/. Accessed: 2020-06-06.

[8] Ellery Wulczyn, Nithum Thain, and Lucas Dixon. Ex machina: Personal attacks seen at scale. In Proceedings of the 26th International Conference on World Wide Web, pages 1391–1399, 2017.

[9] Michal Ptaszynski, Juuso Kalevi Kristian Eronen, and Fumito Masui. Learning deep on cyberbullying is always better than brute force. In LaCATODA@ IJCAI, pages 3–10, 2017.

[10] Sweta Agrawal and Amit Awekar. Deep learning for detecting cyberbullying across multiple social media platforms. In European Conference on Information Retrieval, pages 141–153. Springer, 2018.

[11] Mai Ibrahim, Marwan Torki, and Nagwa El-Makky. Imbalanced toxic comments classification using data augmentation and deep learning. In 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pages 875–878. IEEE, 2018.

[12] Mujahed A Saif, Alexander N Medvedev, Maxim A Medvedev, and Todorka Atanasova. Classification of online toxic comments using the logistic regression and neural networks models. In AIP Conference Proceedings, volume 2048, page 060011. AIP Publishing LLC, 2018.

[13] Spiros V Georgakopoulos, Sotiris K Tasoulis, Aristidis G Vrahatis, and Vassilis P Plagianakos. Convolutional neural networks for toxic comment classification. In Proceedings of the 10th Hellenic Conference on Artificial Intelligence, pages 1–6, 2018.

[14] Sara Zaheri, Jeff Leath, and David Stroud. Toxic comment classification. SMU Data Science Review, 3(1):13, 2020.

[15] Rohit Beniwal and Archna Maurya. Toxic comment classification using hybrid deep learning model. In Sustainable Communication Networks and Application, pages 461–473. Springer, 2021.

[16] ANM Jubaer, Abu Sayem, and Md Ashikur Rahman. Bangla toxic comment classification (machine learning and deep learning approach). In 2019 8th international conference system modeling and advancement in research trends (SMART), pages 62–66. IEEE, 2019.

[17] Betty van Aken, Julian Risch, Ralf Krestel, and Alexander Löser. Challenges for toxic comment classification: An in-depth error analysis. arXiv preprint arXiv:1809.07572, 2018.

[18] Hafiz Hassaan Saeed, Khurram Shahzad, and Faisal Kamiran. Overlapping toxic sentiment classification using deep neural architectures. In 2018 IEEE International Conference on Data Mining Workshops (ICDMW), pages 1361–1366. IEEE, 2018.

[19] Shervin Malmasi and Marcos Zampieri. Detecting hate speech in social media. arXiv preprint arXiv:1712.06427, 2017.

[20] Ziqi Zhang, David Robinson, and Jonathan Tepper. Detecting hate speech on twitter using a convolution-gru based deep neural network. In European semantic web conference, pages 745–760. Springer, 2018.

[21] P Ozoh, O MO, and AA Adigun. Identification and classification of toxic comments on social media using machine learning techniques. International Journal of Research and Innovation in Applied Science (IJRIAS), 2019.

[22] Salvatore Carta, Andrea Corriga, Riccardo Mulas, Diego Reforgiato Recupero, and Roberto Saia. A supervised multi-class multi-label word embeddings approach for toxic comment classification. In KDIR, pages 105–112, 2019.

[23] Björn Gambäck and Utpal Kumar Sikdar. Using convolutional neural networks to classify hate-speech. In Proceedings of the first workshop on abusive language online, pages 85–90, 2017.

[24] S Rahamat Basha, J Keziya Rani, JJC Prasad Yadav, and G Ravi Kumar. Impact of feature selection techniques in text classification: an experimental study. J. Mech. Cont. & Math. Sci., Special Issue, (3):39–51, 2019.

[25] S Rahamat Basha and J Keziya Rani. A comparative approach of dimensionality reduction techniques in text classification. Engineering, Technology & Applied Science Research, 9(6):4974–4979, 2019.

[26] M Suleman Basha, SK Mouleeswaran, and K Rajendra Prasad. Sampling-based visual assessment computing techniques for an efficient social data clustering. The Journal of Supercomputing, pages 1–25, 2021.

[27] Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. Abusive language detection in online user content. In Proceedings of the 25th International Conference on World Wide Web, page 145–153, Republic and Canton of Geneva, CHE, 2016. International World Wide Web Conferences Steering Committee.

[28] Ricardo Martins, Marco Gomes, José João Almeida, Paulo Novais, and Pedro Henriques. Hate speech classification in social media using emo-

tional analysis. In 2018 7th Brazilian Conference on Intelligent Systems (BRACIS), pages 61–66. IEEE, 2018.

[29] Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. Deceiving google's perspective api built for detecting toxic comments. arXiv preprint arXiv:1702.08138, 2017.

[30] Toxic comment classification challenge. https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge. Accessed: 2020-05-05.

[31] Saqib Alam and Nianmin Yao. The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis. Computational and Mathematical Organization Theory, 25(3):319–335, 2019.

[32] Furqan Rustam, Imran Ashraf, Arif Mehmood, Saleem Ullah, and Gyu Sang Choi. Tweets classification on the base of sentiments for us airline companies. Entropy, 21(11):1078, 2019.

[33] Murugan Anandarajan, Chelsey Hill, and Thomas Nolan. Practical text analytics. Maximizing the Value of Text Data.(Advances in Analytics and Data Science. Vol. 2.) Springer, 2019.

[34] Zar Zar Wint, Theo Ducros, and Masayoshi Aritsugi. Spell corrector to social media datasets in message filtering systems. In 2017 Twelfth International Conference on Digital Information Management (ICDIM), pages 209–215. IEEE, 2017.

[35] Sidi Yang and Haiyi Zhang. Text mining of twitter data using a latent dirichlet allocation topic model and sentiment analysis. Int. J. Comput. Inf. Eng, 12:525–529, 2018.

[36] Felipe F Bocca and Luiz Henrique Antunes Rodrigues. The effect of tuning, feature engineering, and feature selection in data mining applied to rainfed sugarcane yield modelling. Computers and electronics in agriculture, 128:67–76, 2016.

[37] Jeff Heaton. An empirical analysis of feature engineering for predictive modeling. SoutheastCon 2016, pages 1–6, 2016.

[38] Shahnoor C Eshan and Mohammad S Hasan. An application of machine learning to detect abusive bengali text. In 2017 20th International Conference of Computer and Information Technology (ICCIT), pages 1–6. IEEE, 2017.

[39] Furqan Rustam, Madiha Khalid, Waqar Aslam, Vaibhav Rupapara, Arif Mehmood, and Gyu Sang Choi. A performance comparison of supervised machine learning models for covid-19 tweets sentiment analysis. Plos one, 16(2):e0245909, 2021.

[40] Boutkhoum Omar, Furqan Rustam, Arif Mehmood, Gyu Sang Choi, et al. Minimizing the overlapping degree to improve class-imbalanced learning under sparse feature selection: Application to fraud detection. IEEE Access, 9:28101–28110, 2021.

[41] T Elhassan and M Aljurf. Classification of imbalance data using tomek link (t-link) combined with random under-sampling (rus) as a data reduction method. 2016.

[42] Joseph Prusa, Taghi M Khoshgoftaar, David J Dittman, and Amri Napolitano. Using random undersampling to alleviate class imbalance on tweet sentiment data. In 2015 IEEE international conference on information reuse and integration, pages 197–202. IEEE, 2015.

[43] Wei-Chao Lin, Chih-Fong Tsai, Ya-Han Hu, and Jing-Shang Jhang. Clustering-based undersampling in class-imbalanced data. Information Sciences, 409:17–26, 2017.

[44] Ahnaf Rashik Hassan and Mohammed Imamul Hassan Bhuiyan. Automated identification of sleep states from eeg signals by means of ensemble empirical mode decomposition and random under sampling boosting. Computer methods and programs in biomedicine, 140:201–210, 2017.

[45] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16:321–357, 2002.

[46] David J Dittman, Taghi M Khoshgoftaar, Randall Wald, and Amri Napolitano. Comparison of data sampling approaches for imbalanced bioinformatics data. In The twenty-seventh international FLAIRS conference, 2014.

[47] Rok Lara Blagus Lusa. Smote for high-dimensional class-imbalanced data. BMC Bioinformatics, 14, 2013.

[48] Alberto Fernández, Salvador Garcia, Francisco Herrera, and Nitesh V Chawla. Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. Journal of artificial intelligence research, 61:863–905, 2018.

[49] Lara Lusa et al. Evaluation of smote for high-dimensional class-imbalanced microarray data. In 2012 11th International Conference on Machine Learning and Applications, volume 2, pages 89–94. IEEE, 2012.

[50] Ml | cost function in logistic regression - geeksforgeeks. https://www.geeksforgeeks.org/ml-cost-function-in-logistic-regression/. Accessed: 2019-12-16.

[51] Muhammad Umer, Imran Ashraf, Arif Mehmood, Saru Kumari, Saleem Ullah, and Gyu Sang Choi. Sentiment analysis of tweets using a unified convolutional neural network-long short-term memory network model. Computational Intelligence, 2020.

VAIBHAV RUPAPARA received a Master of Science degree in Computer Science from Florida International University, Miami, FL, USA. He has worked in different domains including Finance and Healthcare. His expertise contributed towards achieving high quality, scalable deliverability with security. His recent research interests include machine learning, AI, and deep learning.

FURQAN RUSTAM received his MCS degree in the Department of Computer Science, Islamia University of Bahawalpur, Pakistan (Oct-2015 to Oct-2017). Since Nov-2018, he got himself enrolled in Master of Computer Science, Department of Computer Science, Khwaja Fareed University of Engineering and Information Technology (KFUEIT), Rahim Yar Khan, 64200, Pakistan. He is also serving as Research Assistant at Fareed Computing & Research Center, KFUEIT, Pakistan. His recent research interests are related to Data Mining, Machine Learning, and Artificial Intelligence, mainly working on Creative Computing, and Supervised Machine Learning.

HINA FATIMA SHAHZAD received the B.S degree in computer science from The Government Sadiq College Women University (GSCWU), Bahawalpur, in 2018, and currently pursuing the M.S. degree in computer science with the Department of Computer Science, Khwaja Fareed University of Engineering and Information Technology (KFUEIT), Rahim Yar Khan, Pakistan. Her research interest includes text mining, data mining, and machine learning and deep learning-based IoT.

ARIF MEHMOOD received his Ph.D. degree from the Department of Information & Communication Engineering, Yeungnam University, Korea in 2017. From December 2017 to November 2019, he served as an Assistant Professor at the Information Technology department of Khawaja Fareed University of Engineering and Information Technology. Currently, he is serving as an Assistant Professor at Islamia University Bahawalpur, Pakistan. He is working mainly in deep learning-based text mining and data science management techniques.

IMRAN ASHRAF received his Ph.D. in Information & Communication Engineering from Yeungnam University, South Korea in 2018, and the M.S. degree in computer science from the Blekinge Institute of Technology, Karlskrona, Sweden, in 2010. He has worked as a postdoctoral fellow at Yeungnam University, as well. He is currently working as an Assistant Professor at the Information and Communication Engineering Department, Yeungnam University, Gyeongsan, South Korea. His research areas include indoor positioning and localization, advanced location-based services in wireless communication, smart sensors (LIDAR) for smart cars, and data mining.

PROFESSOR GYU SANG CHOI received his Ph.D. from the Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, the USA in 2005. He was a research staff member at Samsung Advanced Institute of Technology (SAIT) for Samsung Electronics from 2006 to 2009. Since 2009, he has been a faculty member in the Department of Information & Communication, Yeungnam University, Korea. His research areas include non-volatile memory and storage systems.