ISSN: 1803-3814 (Printed), 2571-3701 (Online)
https://doi.org/10.13164/mendel.2023.1.001

MENDEL Soft Computing Journal

Offensive Language Detection Using Soft Voting Ensemble Model

Brillian Fieri[1], **Derwin Suhartono**[2,✉]

[1]Computer Science Department, BINUS Graduate Program – Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia
[2]Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia
brillian.fieri@binus.ac.id, dsuhartono@binus.edu✉

Abstract
Offensive language is one of the problems that have become increasingly severe along with the rise of the internet and social media usage. This language can be used to attack a person or specific groups. Automatic moderation, such as the usage of machine learning, can help detect and filter this particular language for someone who needs it. This study focuses on improving the performance of the soft voting classifier to detect offensive language by experimenting with the combinations of the soft voting estimators. The model was applied to a Twitter dataset that was augmented using several augmentation techniques. The features were extracted using Term Frequency-Inverse Document Frequency, sentiment analysis, and GloVe embedding. In this study, there were two types of soft voting models: machine learning-based, with the estimators of Random Forest, Decision Tree, Logistic Regression, Naïve Bayes, and AdaBoost as the best combination, and deep learning-based, with the best estimator combination of Convolutional Neural Network, Bidirectional Long Short-Term Memory, and Bidirectional Gated Recurrent Unit. The results of this study show that the soft voting classifier was better in performance compared to classic machine learning and deep learning models on both original and augmented datasets.

Keywords: *Offensive Language, Text Classification, Voting Classifier, Ensemble Model.*

Received: 12 February 2023
Accepted: 18 May 2023
Online: 31 May 2023
Published: 30 June 2023

1 Introduction

With internet usage uprising, one of the most appealing applications of the internet is social media. Social media has become popular among internet users through the ability to share information and communicate freely. The most used social media by U.S. adults were YouTube, Facebook, and Instagram, with 81%, 69%, and 40%, respectively [1]. However, with all the benefits, social media also comes with drawbacks.

Offensive language is one of the downsides of social media. It is an abusive behavior frequently displayed on social media and other platforms like streaming websites [4]. Offensive language is a type of language that can contain threats, profanities, discriminations, or direct insults [18]. One of the subtypes of offensive language, hate speech, is a type of content that focuses on an attack on certain groups by using nasty and aggressive words [16]. Due to how freely communication can occur in social media, hate speech has become quite common [2]. As an effect, 70.7% of respondents in a survey were exposed to hate speech in the past three months [14].

To overcome the situation, moderation can be used to filter out offensive language. There are several approaches to moderation, such as manual and automated. Manual moderation requires humans to manually check the potentially offensive content based on the predetermined guidelines. However, with the constant addition of new data, manual moderation is not enough to scan and filter out hate speech or offensive content. Conversely, automated moderation can constantly filter the social media data stream. The artificial intelligence model can be trained with machine learning and deep learning to identify and filter offensive content. However, to minimalize the false-positive identification error, the model should have great performance, such as high accuracy and precision.

Study on offensive classification has been done multiple times with various approach. In the machine learning category, a study has been conducted by Davidson et al. [3] by comparing Logistic Regression, Naïve Bayes, Decision Tree, Random Forest, and Linear SVM. This study created their dataset by collecting tweets from Twitter and manually classifying them with the voting method to hate speech, offensive but not hate speech, and neither class. The best model was Logistic Regression with L2 regularization with a precision of 0.91.

Another study [16] also used Davidson et al.'s dataset to classify offensive language and hate speech. This study combined the previous dataset with another dataset from Crowdflower and Github. Random Forest, SVM, and J48graft were trained and

evaluated. J48graft was the best model with 0.793 in precision. Several features like sentiment-based, semantic, unigram, pattern, and a combination of all the features were tested using J48graft. The combination was the best feature, with a precision score of 0.793. Besides classifying three classes, this study also experimented with binary classification, where the class only consisted of offensive and not offensive. To achieve it, hate speech and offensive but not hate speech classes were combined into a single offensive class. The model scored 0.88 on the precision score.

Aside from the machine learning model, deep learning was also used to classify offensive language. A study [10] was conducted to detect the offensive language in Arabic with YouTube comments as the dataset. AraVec, a pre-trained word embedding, was used as the features extractor. The deep learning models used in this study were CNN, Bi-LSTM, attention Bi-LSTM, and a combination of CNN & LSTM. The best model was CNN with the F1 score of 84.05, with CNN-LSTM in second place with 83.65 in the F1 score.

Another approach that was used to classify offensive language is by using the ensemble method. A study [13] in 2019 used the soft voting classifier with XGBoost, AdaBoost, and Logistic Regression as the estimators to classify whether a tweet is a targeted or untargeted offensive language. The soft voting classifier scored 0.706 on the F1 macro score. The study noted that the ensemble method could help decrease the errors in classifying with unbalanced datasets.

More studies [7] used a voting classifier by combining machine learning and deep learning models. This study used three datasets, consisting of one primary dataset and two augmentation data. The voting classifier was made of two CNN with different training dataset combinations and an SVM model. The voting classifier topped the standalone CNN and SVM model with the F1 macro score of 0.7325.

Besides classifying offensive content, soft voting classifiers were also used to detect fake news. The study [9] compared Naïve Bayes, SVM, Logistic Regression, Random Forest, hard voting, and soft voting to predict fake news on social media. The soft voting method surpassed the other classifiers with 93% in the F1 score. Another study [20] used a soft voting classifier with LSTM-Attention and Bi-LSTM Attention as its estimators to predict emotions from social media posts. The soft voting classifier came on top of other classifiers with an F1 score of 68.5%.

From previous research, this study concluded that several approaches were already used, such as machine learning, deep learning, and hybrid machine learning to classify offensive content. These studies showed performance improvement from the soft voting classifier compared to standalone models. However, with a tremendous amount of various machine learning and deep learning models, the combination of voting classifier estimators can be explored further.

This study aims to improve the performance of offensive classification by experimenting with the combination of machine learning and deep learning as the core of the ensemble model using the dataset from Davidson et al. Numerous machine learning such as Random Forest, Decision Tree, k-Nearest Neighbor (k-NN), Naïve Bayes, Logistic Regression, AdaBoost, and deep learning models such as CNN, LSTM, and Gated Recurrent Unit (GRU) were compared and used as the combination for the soft voting classifier.

2 Materials and Methods

2.1 Dataset

The dataset used in this study was acquired from the previous study [3]. It consists of tweet data from Twitter and their classification. The classification was decided with voting by Crowdflower, where each tweet was voted between three classes: hate speech, offensive language but not hate speech, and neither. The data count for each class is 1,430, 19,190, and 4,163, respectively. However, in order to identify offensive content, the hate speech and offensive but not hate speech classes were merged into one offensive class. This approach was implemented in a previous study [16]. This resulted in 20,620 offensive class and 4,163 not offensive class for the final data count.

2.2 Dataset Preprocessing & Augmentation

The next step after the class data were merged was to apply the preprocessing step to the data. The step can be found in Figure 1. The first step was to cleanse the dataset of imperfect data. In this stage, incomplete data were removed, along with the URL text, mentions, and punctuation marks. Afterward, the tweet data were stemmed to return all the words to their stem.

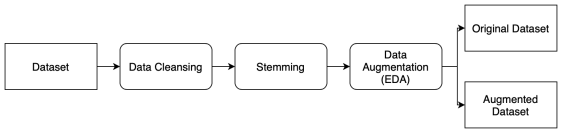

Figure 1: Dataset Preprocessing & Augmentation.

In addition to data preprocessing, the preprocessed data were augmented by applying Easy Data Augmentation (EDA) tool [17] to balance the dataset. This was done because the dataset was quite unbalanced, where 80% of the data were labeled as offensive. In EDA, the minority class data, in this case, was not the offensive class, were randomly selected and augmented using one of the techniques provided, which was also selected randomly. These techniques were synonym replacement, random insertion, random swap, and random deletion. The process was repeated until the minority class number was equal to the majority class. The result of this process was the original and augmented dataset.

2.3 Feature Extraction

Before the feature extraction process, the tweets were tokenized, where each sentence was broken into words. The extraction used for deep learning and machine learning was different. TF-IDF and sentiment analysis were used for machine learning. TF-IDF calculates the importance score of each word in a document, which can be used as a feature for text classification [12]. The other feature, sentiment analysis, was used to extract the sentiment values, such as positive, neutral, and negative scores, from the text content [15]. In this study, Valence Aware Dictionary and Sentiment Reasoner (VADER) [5] were used to extract the sentiment analysis. Feature extraction visualization can be found in Figure 2.

In deep learning, GloVe embedding [11] was used as the feature extraction method. This study used Twitter pre-trained word vectors with 27 billion tokens, 1.2 million vocabularies, and 200 dimensions vectors. Subsequently, the features extracted from the dataset can be used for model development in the next step.

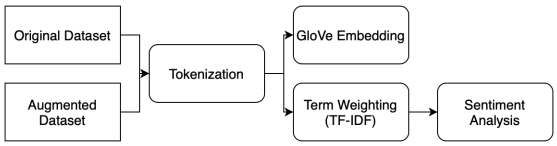

Figure 2: Feature Extraction.

2.4 Model Development

The model proposed in this study is a voting classifier, which is a type of ensemble model. The voting classifier combines various machine learning models, called estimators, to make a prediction by letting the models vote for the class prediction [6]. There are two types of voting classifiers based on how the voting is applied: soft and hard voting [19]. In hard voting, the final class prediction is determined by a majority vote from various estimator models. In soft voting, the model calculates the probability for each class from each estimator to determine the final class prediction [8]. Voting example for soft and hard voting can be seen in Figure 3.

In this study, the model development was split into two sections: machine learning estimators and deep learning estimators. For the machine learning estimators, several models were chosen to be compared based on popular models that were used in previous studies. They are Random Forest, Decision Tree, Logistic Regression, Naïve Bayes, AdaBoost, and k-Nearest Neighbor. The models were trained and evaluated using 5-fold cross-validation for each original and augmented dataset. Subsequently, the top three and top five of the best-performing models for each dataset based on their F1 scores were chosen as the voting classifier's estimators. The workflow for this part can be found in Figure 4.

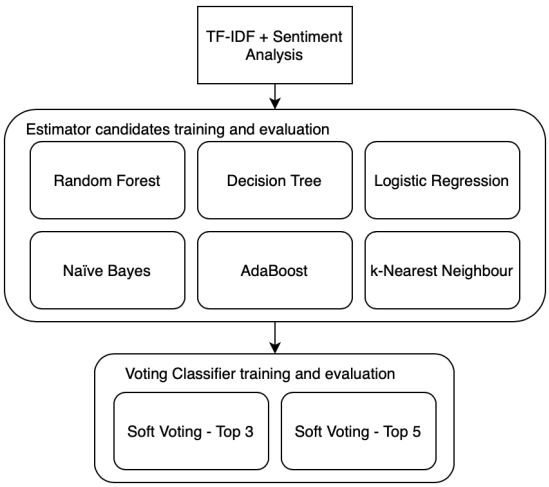

Figure 4: Voting Classifier with Machine Learning Estimator.

For the deep learning estimators, three deep learning groups were trained and evaluated using 5-fold cross-validation using both original and augmented datasets. These groups were CNN, Long Short-Term Memory (e.g., LSTM & Bidirectional-LSTM), and Gated Re-

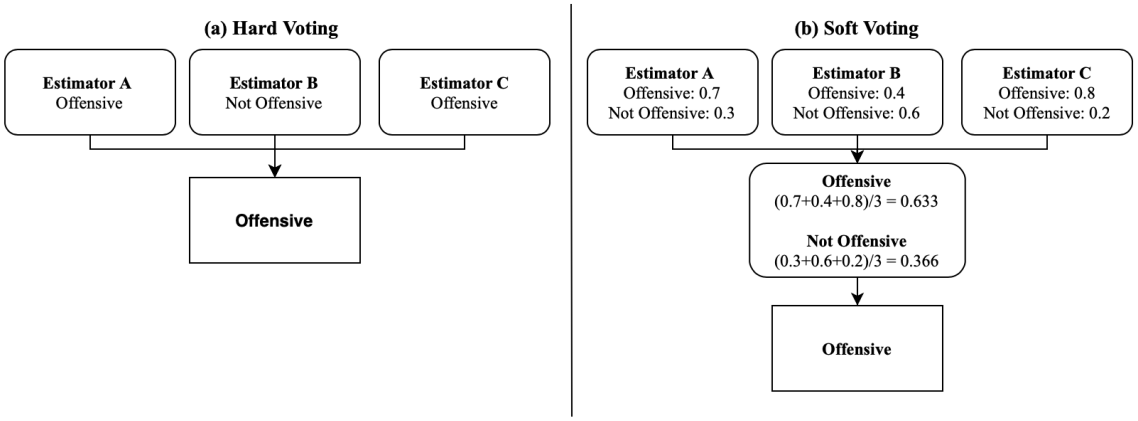

Figure 3: Voting Classifier. (a) Hard Voting. (b) Soft Voting.

current (e.g., GRU & Bidirectional-GRU), as seen in Figure 5. There is a conventional and bidirectional version of the model for particular groups, where the difference is only in the input flow directions. Consequently, the best-performing model based on the F1 score for each group would be taken to be the estimator for the voting classifier.

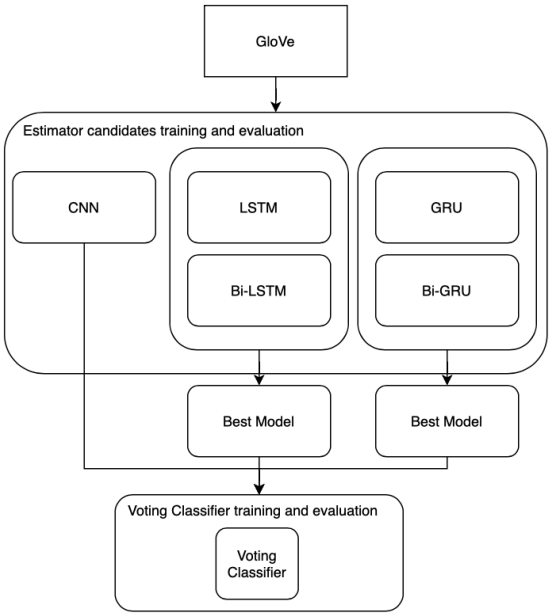

Figure 5: Voting Classifier with Deep Learning Estimator.

The general architecture for the deep learning models can be found in Figure 6. The embedded dataset using GloVe was then fed to the model layer with the output size of 512, which in this case, it could be CNN, LSTM, Bi-LSTM, GRU, or Bi-GRU layer. The output was then fed to the dropout layer at the rate of 0.2. The dropout layer was used to prevent model overfitting. Next, the output was sent to another model layer with an output size of 256, then to another dropout layer with a 0.2 rate. Afterward, the data were sent to a dense layer with the Rectified Linear Unit (ReLU) activation function to reduce the output to 64. The last step was to send the output to the dense layer with the softmax function. Softmax was chosen to get the probability of the offensive and not the offensive class. These probabilities were needed for the soft voting operation.

2.5 Model Evaluation

For every machine learning, deep learning, and voting classifier model, evaluation metrics such as accuracy, F1, precision, and recall were calculated using the macro averaging method. The evaluation metrics for each fold in 5-fold cross-validation would be stored and averaged out to be the final scores for each model.

3 Results and Discussion

The results were broken into machine learning and deep learning segments in this part. For each segment, the base models' accuracy, precision, recall, and F1 score were shown for both original and EDA datasets. The best models were then chosen as the voting classifier estimators, and the performance results were also displayed for both datasets respectively.

3.1 Machine Learning

Table 1 shows the performance result for the machine learning-based model using the original dataset. Based on the F1 score, the best model was the Random Forest, with an F1 score of 90.070%. The order of the models from best to worst was Random Forest, Logistic Regression, AdaBoost, Decision Tree, Naïve Bayes, and k-NN. From the order, the models' top three and top five were selected to be used as the estimators for the voting classifier. For the voting classifier results, the top three and top five voting classifiers were better than the best-performing machine learning model, Random Forest. The top three resulted in 91.370% on F1, which were higher by 1.3% from the Random Forest result. Moreover, the top five voting classifier marked a better F1 score, with a score of 91.509%, resulting in a 1.439% higher F1 score than the Random Forest model.

For the EDA dataset, the result can be found in Table 2. The order of the best-performing models for this dataset was slightly different from the original dataset result, with the best model, Random Forest, scoring 92.685% on F1. The best to worst models based on the F1 score was Random Forest, Decision Tree, Logistic Regression, AdaBoost, Naïve Bayes, and k-NN. The RF score using the EDA dataset was higher than the original dataset result, with an increase of over 2.5% on the F1 score. This increment was also achieved by most of the models, with the highest difference of 6.3% by the k-NN model.

From the F1 result, Random Forest, Logistic Regression, and AdaBoost were chosen to be the estimators

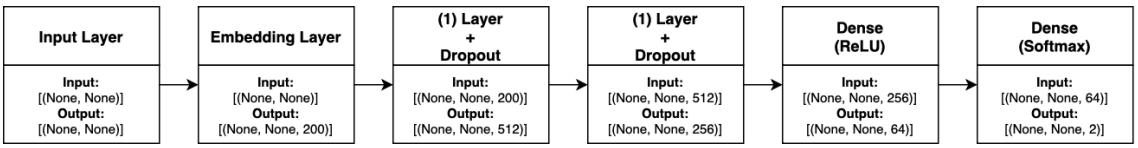

Figure 6: Deep Learning Architecture. CNN/LSTM/Bi-LSTM/GRU/Bi-GRU.

Table 1: Original Dataset – Machine Learning Performance.

Model	Accuracy	Precision	Recall	F1
Random Forest (RF)	94.460	90.184	89.957	90.070
Logistic Regression (LR)	93.762	86.786	94.494	89.945
AdaBoost (AB)	94.014	88.113	91.795	89.801
Decision Tree (DT)	92.941	87.176	87.817	87.491
Naïve Bayes (NB)	91.588	83.785	89.200	86.110
k-NN	89.195	83.636	74.224	77.685
Soft Voting – Top 3 (RF, LR, AB)	94.896	89.379	93.779	91.370
Soft Voting – Top 5 (RF, LR, AB, DT, NB)	95.034	89.866	93.423	**91.509**

Table 2: EDA Dataset – Machine Learning Performance.

Model	Accuracy	Precision	Recall	F1
Random Forest (RF)	95.518	89.610	96.890	92.685
Decision Tree (DT)	95.357	89.385	96.545	92.414
Logistic Regression (LR)	94.422	87.696	95.949	91.050
AdaBoost (AB)	93.572	86.355	94.943	89.773
Naïve Bayes (NB)	92.808	85.456	92.355	88.322
k-NN	89.494	80.515	90.566	84.030
Soft Voting – Top 3 (RF, DT, LR)	95.433	89.445	96.850	92.560
Soft Voting – Top 5 (RF, DT, LR, AB, NB)	95.571	89.743	96.820	**92.750**

for the top three soft voting estimators. These top three lists were dissimilar from the original dataset. The list contained the exact estimators for the top five soft voting. As a result, both the top three and the top five soft voting with the EDA dataset marked a higher F1 score, where the top five gained the biggest boost, with a score of 92.750%. The top five also outperformed the best machine learning model, Random Forest. Although, the difference was smaller than the increment in the original dataset

3.2 Deep Learning

In the deep learning segment, the three groups were trained and tested: CNN, Long Short-Term Memory (LSTM & Bi-LSTM), and Gated Recurrent Unit (GRU & Bi-GRU). The result of 5-fold cross-validation using the original dataset can be found in Table 3. Based on the result, in the Long Short-Term Memory group, Bi-LSTM was superior with a score of 91.001%, while in the Gated Recurrent Unit group, the Bi-GRU F1 score surpassed the conventional GRU model. This outcome resulted in soft voting with CNN, Bi-LSTM, and Bi-GRU as its estimators.

The soft voting result was higher than the best deep learning model, the Bi-LSTM. It scored 91.864% on F1, with a 0.86% increment from the Bi-LSTM. It was also better than the best original dataset's soft voting with machine learning models as its estimators. Moreover, all the F1 scores for the tested deep learning models were more significant than the machine learning models with the original dataset.

Furthermore, the deep learning models with the EDA dataset gained quite a boost compared to the ones with the original dataset. As shown in Table 4, the result of this dataset was CNN, Bi-LSTM, and Bi-GRU were chosen from each respective group to be the estimator of the soft voting model. The best model was Bi-LSTM with the F1 score of 93.56%, which were higher than the original dataset Bi-LSTM with a difference of 2.5%. It also surpassed the soft voting score from the original dataset by 1.6%.

Table 3: Original Dataset – Deep Learning Performance.

Model	Accuracy	Precision	Recall	F1
CNN	94.711	90.431	90.739	90.584
LSTM	94.707	90.715	90.264	90.487
Bi-LSTM	94.958	90.946	91.057	91.001
GRU	94.896	91.065	90.592	90.826
Bi-GRU	94.887	90.489	91.476	90.972
Soft Voting (CNN, Bi-LSTM, Bi-GRU)	95.381	91.275	92.482	**91.864**

Table 4: EDA Dataset – Deep Learning Performance.

Model	Accuracy	Precision	Recall	F1
CNN	95.713	89.979	97.018	92.976
LSTM	96.078	90.747	97.114	93.512
Bi-LSTM	96.102	90.755	97.230	93.560
GRU	96.050	90.676	97.131	93.473
Bi-GRU	96.083	90.756	97.117	93.520
Soft Voting (CNN, Bi-LSTM, Bi-GRU)	96.268	91.073	97.374	**93.818**

Moreover, the soft voting model with the EDA dataset gained 93.818% on the F1 score. This score beat the EDA Bi-LSTM with a 0.25% margin. This result shows that the difference between the soft voting and the deep learning models was not as huge in the EDA as in the original dataset. However, the soft voting model with deep learning estimator from the EDA dataset beat all the other soft voting models in this study.

4 Conclusion

This study focused on detecting offensive language from social media texts using the soft voting classifier. The dataset that was used was Twitter tweets from a previous study. To find the best estimator combination, two types of models were compared and used as the estimators, which were machine learning (e.g., Random Forest, Decision Tree, Logistic Regression, Naïve Bayes, k-Nearest Neighbor, and AdaBoost) and deep learning models (e.g., CNN, Long Short-Term Memory, Bidirectional LSTM, Gated Recurrent Unit, and Bidirectional-GRU). The dataset was also augmented using the EDA method to balance the class count. Both original and EDA datasets were compared and used on both deep learning and machine learning types.

Some conclusions can be drawn from this study:

a) For the machine learning models, the soft voting model with the same top five machine learning models as its estimator beat the performance score of other machine learning models on both the original and augmented dataset. The estimator combination was Random Forest, Decision Tree, Logistic Regression, Naïve Bayes, and AdaBoost.

b) The soft voting classifier with the exact estimators

in deep learning also exceeded other deep learning models on the original dan EDA dataset. The estimators were CNN, Bi-LSTM, and Bi-GRU.

c) The F1 and recall scores for both deep learning and machine learning from using the EDA dataset were generally higher than when using the original dataset.

Overall, the soft voting proposed in this study shows higher results in both machine learning and deep learning segments. Soft voting with deep learning estimators scored higher F1 scores than the ones with machine learning estimators. Further work can be done by experimenting with more combinations of machine learning and deep learning as its estimators. Furthermore, the soft voting classifier can also be applied to classify other types of data.

References

[1] Auxier, B., and Anderson, M. Social media use in 2021. *Pew Research Center 1* (2021), 1–4.

[2] Benesch, S. Defining and diminishing hate speech. *State of the world's minorities and indigenous peoples 2014* (2014), 18–25.

[3] Davidson, T., Warmsley, D., Macy, M., and Weber, I. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media* (2017), vol. 11, pp. 512–515.

[4] Gao, Z., Yada, S., Wakamiya, S., and Aramaki, E. Offensive language detection on video live streaming chat. In *Proceedings of the 28th International Conference on Computational Linguistics* (2020), pp. 1936–1940.

[5] Hutto, C., and Gilbert, E. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media* (2014), vol. 8, pp. 216–225.

[6] Kabari, L. G., and Onwuka, U. C. Comparison of bagging and voting ensemble machine learning algorithm as a classifier. *International Journals of Advanced Research in Computer Science and Software Engineering 9*, 3 (2019), 19–23.

[7] Kebriaei, E., Karimi, S., Sabri, N., and Shakery, A. Emad at semeval-2019 task 6: offensive language identification using traditional machine learning and deep learning approaches. In *Proceedings of the 13th International Workshop on Semantic Evaluation* (2019), pp. 600–603.

[8] Kumari, S., Kumar, D., and Mittal, M. An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier. *International Journal of Cognitive Computing in Engineering 2* (2021), 40–46.

[9] Lasotte, Y., Garba, E., Malgwi, Y., and Buhari, M. An ensemble machine learning approach for fake news detection and classification using a soft voting classifier. *European Journal of Electrical Engineering and Computer Science 6*, 2 (2022), 1–7.

[10] Mohaouchane, H., Mourhir, A., and Nikolov, N. S. Detecting offensive language on arabic social media using deep learning. In *2019 sixth international conference on social networks analysis, management and security (SNAMS)* (2019), IEEE, pp. 466–471.

[11] Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (2014), pp. 1532–1543.

[12] Perera, A., and Fernando, P. Accurate cyberbullying detection and prevention on social media. *Procedia Computer Science 181* (2021), 605–611.

[13] Rajendran, A., Zhang, C., and Abdul-Mageed, M. Ubc-nlp at semeval-2019 task 6: Ensemble learning of offensive content with enhanced training data. *arXiv preprint arXiv:1906.03692* (2019).

[14] Reichelmann, A., Hawdon, J., Costello, M., Ryan, J., Blaya, C., Llorent, V., Oksanen, A., Räsänen, P., and Zych, I. Hate knows no boundaries: Online hate in six nations. *Deviant Behavior 42*, 9 (2021), 1100–1111.

[15] Sudhir, P., and Suresh, V. D. Comparative study of various approaches, applications and classifiers for sentiment analysis. *Global Transitions Proceedings 2*, 2 (2021), 205–211.

[16] Watanabe, H., Bouazizi, M., and Ohtsuki, T. Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. *IEEE access 6* (2018), 13825–13835.

[17] Wei, J., and Zou, K. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196* (2019).

[18] Wiedemann, G., Ruppert, E., Jindal, R., and Biemann, C. Transfer learning from lda to bilstm-cnn for offensive language detection in twitter. *arXiv preprint arXiv:1811.02906* (2018).

[19] Żabiński, G., Gramacki, J., Gramacki, A., Miśta-Jakubowska, E., Birch, T., and Disser, A. Multi-classifier majority voting analyses in provenance studies on iron artefacts. *Journal of Archaeological Science 113* (2020), 105055.

[20] Zhou, Q., and Wu, H. Nlp at iest 2018: Bilstm-attention and lstm-attention via soft voting in emotion classification. In *Proceedings of the 9th workshop on computational approaches to subjectivity, sentiment and social media analysis* (2018), pp. 189–194.