Chapter 7

Combating Hate Speech using an Adaptive Ensemble Learning Model with a case study on COVID-19

7.1 Introduction

Hate speech on social media is defined as an online post that demonstrates hatred towards a race, colour, sexual orientation, religion, ethnicity or one's political inclination. Hate speech is not a trivial task to define, mainly because it is subjective. The classification of content as hate speech might be influenced by the relationships between individual groups, communities, and language nuances.

Davidson et al. [67] define hate speech as *"the language that is used to express hatred towards a targeted group or is intended to be derogatory, to humiliate, or to insult the members of the group."* The critical point to note here is that hate speech is usually expressed towards a group or a community and causes/ may cause social disorder.

Psychologists claim that the anonymity provided by the Social Media Platforms (SMPs) is one of the reasons why people tend to be more aggressive in such environments [174], [175]. This aggression sometimes turns into hate speech. Also, people tend to be more involved in heated debates on social media rather than involving in a face to face discussion[1]. The impact of social media can not be underestimated. The propagation of hate speech has the potential for societal impact. It has been observed in the past that

[1] `https://www.bbc.com/news/blogs-trending-35111707`

posts shared on social platforms or textual exchanges may instigate individuals or groups affecting the democratic process [174]. Along with the societal impact of the phenomenon, hate speech makes it uncomfortable for users who use social media only for entertainment.

According to the European Union Commission directives[2], hate speech is a criminal offence that needs to be legislated [174]. European Union has instructed the various SMPs to improve their automatic hate speech detection mechanisms so that no objectionable content remains available online for more than 24 hours[3].

European Union defines hate speech as *"All conduct publicly inciting to violence or hatred directed against a group of persons or a member of such a group defined by reference to race, colour, religion, descent or national or ethnic"*[4] International minorities associations (ILGA) define hate speech as *"Hate speech is public expressions which spread, incite, promote or justify hatred, discrimination or hostility toward a specific group. They contribute to a general climate of intolerance which in turn makes attacks more probable against those given groups."* [5]

The key points to take from the above definitions are:

- Hate speech is meant to target a group or a community based on their ethnicity, religion, origin, sexual orientation, physical appearances or political inclinations.

- Hate speech has the potential to instigate violence or social disorder. Hate speech is commonly observed when people engage themselves in heated arguments based on their political inclinations. In extreme cases, it may also affect the democratic processes.

- There is a fine line between humorous content and hate speech. Although humorous content may offend people, its nature is only to entertain people and not to cause a stir in society. Facebook differentiates between hate speech and humorous content as *"...We do, however, allow clear attempts at humour or satire that might otherwise be considered a possible threat or attack. This includes content that many people may find to be in bad taste (ex: jokes, stand-up comedy, popular song lyrics, etc.)."*

[2] https://money.cnn.com/2017/06/01/technology/twitter-facebook-hate-speech-europe/index.html

[3] https://www.theguardian.com/technology/2016/may/31/facebook-youtube-twitter-microsoft-eu-hate-

[4] ec.europa.eu/commission/presscorner/detail/en/SPEECH_17_403

[5] https://ilga-europe.org/what-we-do/our-advocacy-work/hate-crime-hate-speech

6

Studies have been conducted to determine the more affected or frequently targeted groups by online hate speech. Some of the common phenomena observed in the studies are:

- Racism: The majority of the hate speech content online is based on racism where people are attacked based on their race [176]. In [177], the authors conducted a study on why social media contents are flagged as racist. They found out that in most cases (86%), the reason is "the presence of offensive words". Other than that, "the presence of stereotypes and threatening" and "references to painful historical contexts" make the content racist.

- Sexism: Another category of hate speech is based on sexism which is majorly caused by the misogynistic language used on SMPs [178–182]. Studies conducted on UK-based Twitter profiles found around 100,000 instances of the word "rape", out of which around 12% of the cases were threatening. The sad part is the observation that in about 29% of these instances, the word "rape" is used casually or metaphorically[7]. The same study also shows that women are as likely as men to post offensive tweets against women.

Frequent use of swearing in the text does not necessarily imply spreading hate speech [183]. At the same time, hateful comments can be propagated in subtle and sarcastic ways.

It is a general misconception to see offensive language as hate speech. While the former is morally wrong and flaunts ill-mannerism, the latter is a crime and prohibited by law[8] (Hate speech is prohibited in the United States under the free speech provisions of the first amendment[9] [184]. Countries like United Kingdom, France, Canada, etc., have imposed laws on propagating hate speech that affects minorities and may result in community violence). As these laws extend to social media and digital platforms, it becomes essential for the SMPs to administer their provisions for hate speech detection. SMPs like Facebook and Twitter have instituted policies for banning posts suspected of hate speech, but most of their mechanisms are based on manual inspection [174].

[6] https://www.facebook.com/communitystandards/objectionable_content

[7] https://demosuk.wpengine.com/files/MISOGYNY_ON_TWITTER.pdf?1399567516

[8] http://www.mandola-project.eu/m/filer_public/8d/b3/8db3c018-4e9f-4120-95cc-bf80b38f1749/mandola_dd21b.pdf

[9] https://en.wikipedia.org/wiki/Hate_speech_in_the_United_States

While the manual review of the content is accurate to a greater extent, it is relatively slow because of which the content in question can be available online for a long time. In addition, in the unprecedented times caused by COVID19, it is unfeasible for SMPs to flag all the hateful content manually. Therefore, the need of the hour is an efficient expert model for automatic hate speech detection.

The rise in hate speech propagation on social media has alerted the SMPs to take decisive actions against it. Twitter, for instance, has declared an update on its rules for flagging content as hate speech[10]. The examples given below are now considered as hateful conduct and will be removed *if reported*.

> "All [Age Group] are leeches and do not deserve any support from us."
>
> "People with [Disease] are rats that contaminate everyone around them."
>
> "[Religious Group] should be punished. We are not doing enough to rid us of those filthy animals."

Hate speech detection is a challenging task because of the nature of hate speech. As human editorial methods are not feasible on enormous volumes of tweets, automated approaches or expert models are required to address the issue. The scientific study of hate speech involves analyzing the existing and proposing novel automated approaches that programmatically classify social media posts as hate speech [185]. The current procedures for hate speech detection consider it an application of supervised learning with an assumption that the ground truth is available [186]. The state-of-the-art methods achieve excellent performance within specific datasets [7], [8]. Unfortunately, the performance of these methods degrades drastically, when tested on cross-datasets (i.e., similar but not same dataset) [55]. Thus, we claim that to incorporate the data-bias, the model requires to be adaptive towards the properties of data.

Therefore, we propose an adaptive model for automatic hate speech detection, which can overcome the data-bias and perform well on cross-datasets. Our proposed method is based on our previous work A-Stacking [6], an ensemble-based classifier used originally for spoof fingerprint detection. A-Stacking is an adaptive classifier that uses clustering to conform to the dataset's features and generate hypotheses dynamically.

In Table 7.1, we highlight the difference in the proposed work, [7] and [8]'s methods for hate speech detection. As reported by [55], the papers, as mentioned above, have

[10] `https://blog.twitter.com/en_us/topics/company/2019/hatefulconductupdate.html`

Table 7.1: Comparison with state-of-the-art.

Method	Classifier	Cross-Dataset Evaluation	Data-Bias Control
[7]	GBDT	✗	✗
[8]	BiLSTM	✗	✗
Proposed Work	A-Stacking	✓	✓

flaws in their experimental settings, which is why they overestimate their results. Specifically, the authors have misdealt with data overfitting and oversampling techniques and overstated the results. [55] corrected the experimental settings and reported the actual results. In addition, they also found out a strong user bias in the popular datasets. They observed a drastic change in the performance of the state-of-the-art methods on cross-datasets and when user bias is removed from the datasets. We take care of overfitting issues and compare our performance with the corrected results reported in [55] and observe that our adaptive model proves to be outperforming on cross-datasets environments while maintaining a decent performance on the within-dataset environment.

We list our contributions as follows:

- We provide a comprehensive study on the importance of automatic hate speech detection in the times of COVID'19 and the US presidential election. We highlight the worrying rise in hate speech on SMPs during the pandemic and the democratic process, and the need for achieving non-discriminatory access to digital platforms.
- Unlike the state-of-the-art methods that fail to achieve cross-dataset generalization, our proposed adaptive model yields adequate performance under cross-datasets environments.
- We perform our experiments on standard high-dimensional datasets. We use multiple experimental settings to explore the model's behaviour while considering the user-overfitting effect, data-bias, and restricting the number of tweets per user.

7.2 Importance of Automatic Hate Speech Detection

Hate speech detection has become a popular research area in recent years. The need for automatic mechanisms for detecting hate speech on social media becomes urgent in the difficult times of global pandemic and during democratic processes such as the US presidential elections. We describe the importance of automatic hate speech detection in the following subsections:

7.2.1 Hate Speech in the Times of COVID-19

While Internet communication has helped the world survive this unprecedented period, it has also grown more toxic than before. After the widespread outbreak of COVID-19, an AI-based start-up L1ght analysed the SMPs and observed that the propagation of hate speech has increased by 900% during the pandemic[11]. They have also observed a 70% rise in toxicity among teens and youngsters during the period of December 2019 to June 2020. Although we do not endorse these numbers, it is evident that there has been a worrying rise in hate speech propagation during the pandemic. Most of the hateful conduct regarding the novel coronavirus situation is directed towards China and the Chinese. As we have defined earlier, targeting a community [187] or a race online is against the guidelines and is considered to be hate speech [188]. The ongoing global pandemic has forced the people to live entirely within four walls and rely on online platforms for work, education, communication and entertainment. The study claims that Asians are being targeted online for allegedly carrying the coronavirus and being the cause of it. The popular hashtags on Twitter are *#Kungflu, #communistvirus, #whuanvirus, #chinesevirus*, which shows the level of hatred present on social media.

On May 8, 2020, the UN chief appealed to combat hate speech to avoid social disorder during the pandemic.

> *"...Yet the pandemic continues to unleash a tsunami of hate and xenophobia, scapegoating and scare-mongering...We must act now to strengthen the immunity of our societies against the virus of hate. That's why I'm appealing today for an all-out effort to end hate speech globally...I call on the media, especially social media companies, to do much more to flag and, in line with*

[11] `https://l1ght.com/Toxicity_during_coronavirus_Report-L1ght.pdf`

international human rights law, remove racist, misogynist and other harmful content."

Therefore, while we fight the pandemic of COVID19, it is also essential to deal with hate speech on social media. In these challenging times, when the information is sensitive, the research on automatic hate speech detection must be encouraged.

7.2.2 Hate Speech Related to US Presidential Election

Late in June 2020, Reddit, the most popular comment forum, had to ban one of its subreddits *"The_Donald"* (constituted by 790,000 users) due to the crackdown of hate speech[12]. Civil rights groups in the United States have advocated that these SMPs are not doing enough to retaliate against the spread of racist and violent content. Twitter moved one step further and hid one of the tweets made by the president to his 83 Million followers, as the tweet violated the policy against hate speech and glorifying violence[13]. Amazon-owned streaming platform Twitch suspended the president's account over 'hateful conduct'. On June 1, 2020, Twitter removed the tweet posted by one of the US Government officials stating that the tweet *glorified violence*.

"Now that we clearly see Antifa as terrorists, can we hunt them down like we do those in the Middle East?"

This shows the amount of seriousness these SMPs are willing to consider against the propagation of hate speech as it has the power to affect the democratic process.

7.3 Proposed Model for Automatic Hate Speech Detection

In this section, we describe the proposed methodology used for automatic hate speech detection. As mentioned earlier, the major limitation of the existing methods is their inability to perform on cross-dataset environments, i.e., when the trained model is tested on a similar but different dataset, the performance drops drastically. Therefore, the

[12] https://www.nytimes.com/2020/06/29/technology/reddit-hate-speech.html

[13] https://www.washingtonpost.com/technology/2020/07/10/hate-speech-trump-tech/

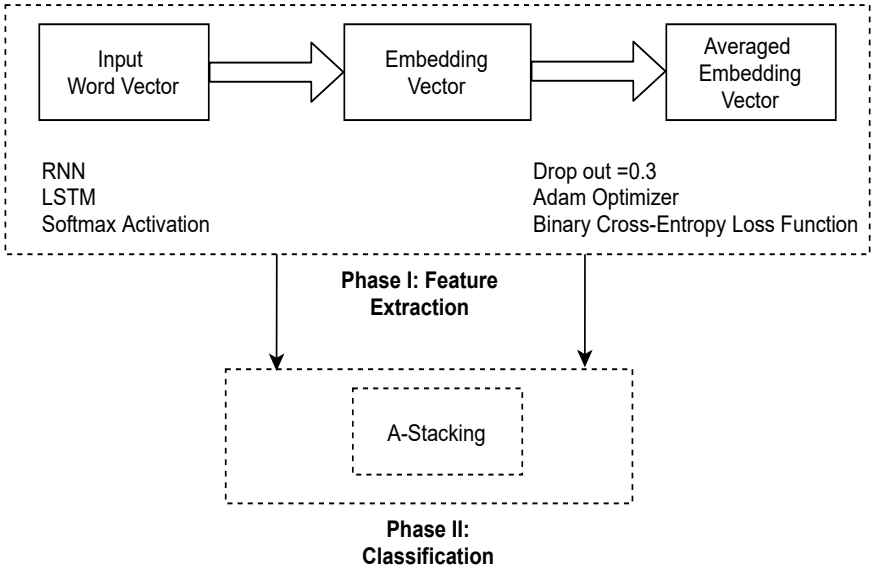

Figure 7.1: Schematic diagram of the proposed model.

existing models show poor generalization abilities and the need to adapt to the changing environment.

We make use of our previous work A-Stacking [6], which is an adaptive ensemble learning model proposed initially for spoof fingerprint detection. We claim that the required adaptiveness for hate speech detection can be achieved by using the A-Stacking classifier.

A-Stacking is a hybrid classifier based on ensemble learning that uses clustering to form weak hypotheses that are carefully integrated using a meta-classifier in a later stage. The model is adaptive because it studies the properties of data for generating the hypotheses to be used by base-classifiers. We have shown in [6] that A-Stacking adapts to the features of fingerprint images and efficiently classifies the new images on which the model has not been trained yet. Therefore, the motivation is to use this adaptive classifier for achieving better cross-dataset generalization for automatic hate speech detection.

We divide the whole architecture into two phases, as represented in Figure 7.1. Phase-I is responsible for feature extraction, and phase-II is for classification. We use the Recurrent Neural Network (RNN) [189] for generating dense-vector representations for the tweets, also known as word-embeddings. RNN makes it possible to consider the context of words while generating embeddings. The embedding layer is followed by Long Short-Term Memory (LSTM) network [54] and a softmax activation. We use binary cross-entropy [190] as the loss function and a dropout of 0.3 along with the Adam optimizer. The

embedding dimension is 200. This process is carefully done while following the guidelines by [55] so that the overfitting bias in the performance can be avoided.

As described earlier, a tweet can be seen as a vector of words, $t =< w_1, w_2, .., w_k >$. The task is to create a dictionary of words and pass it to the embedding layer that generates a sequence of vectors, $E_t =< e_{w_1}, e_{w_2}, .., e_{w_k} >$. These vectors are averaged and a single vector $\overrightarrow{E_t}$ for each tweet is generated. In this study, we do not perform end-to-end classification, rather we generate these sets of vectors for train set D_{Train} and test set D_{Test} and later classify D_{Test}.

In phase-II, D_{Train} is used to train the A-Stacking model. First, D_{Train} is partitioned into two parts: one is used for validation D_{Valid} and the rest of it is used for actual training. In this study, we have considered D_{Valid} to be 20% of D_{Train}. This way, train, test and validation sets are disjoint from each other.

The classifier studies the data and constitutes a set of clusters $C =< c_1, c_2, .., c_n >$, where the number of clusters n is decided apriori. Later, base classifiers are trained on these individual clusters and a set of hypotheses $H =< h_1, h_2, .., h_n >$ is generated. Each hypothesis h_j is tested on multiple base classifiers (e.g., SVM, GBDT, etc.) and the best performing base classifier is chosen based on its performance on D_{Valid} and fixed for the particular hypothesis.

Later, each instance i of D_{Test} is passed through the set H and the individual decisions v_j of each h_j are recorded and sent to a meta-classifier M. This meta-classifier is responsible for carefully integrating the set of v_j's and coming up with the final decision. This final decision is the class label for the tweet, i.e., sexist, racist, non-hateful etc.

7.4 Parallel Ensemble Learning Models

The parallel architecture followed in this study is depicted in Figure 7.2. Our methodology starts with passing input tweets to the model. Later, these tweets are processed by extracting features from them. These features are used to train various base classifiers that run in parallel. The central part of the proposed architecture is the explicit parallelization introduced with the motive of accelerating the training of base classifiers. The base classifiers are trained individually in parallel and later integrated carefully to make the final decision. We consider the following ensemble algorithms in our analysis:

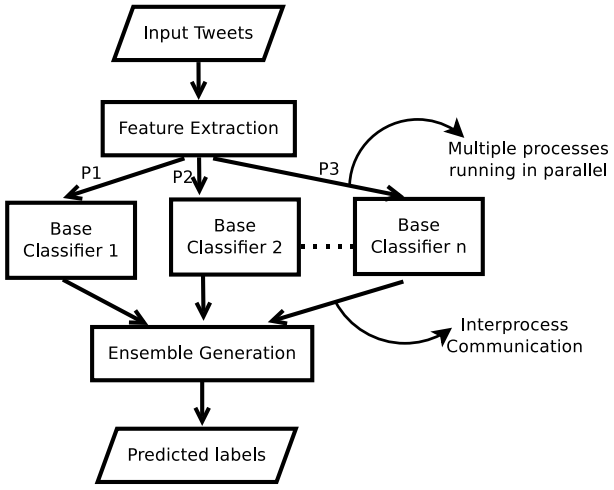

Figure 7.2: Schematic representation of the proposed parallelization of ensemble models.

7.4.1 Parallelized Bagging

The algorithm for the parallelized version of bagging is presented in Algorithm 5. P-Bagging requires D_{Train} for training the model. D_{Train} consists D instances and f features acquired from the feature extraction phase. In addition, P-Bagging requires test data D_{Test}, and a base classifier K. Similar to bagging, P-Bagging expects the number of base classifiers n to be defined prior to training. The output of the algorithm is an ensemble Z of base classifiers, which is used for classifying the query instances.

Algorithm 5 is initiated by sampling bags of instances $c_1, c_2, .., c_n$ from D_{Train}. It is to be noted that each c_i contains D instances that are picked randomly and replaced in the original data. Therefore, it is not guaranteed that $c_i \cap c_j = \phi$ or $c_1 \cup c_2 \cup ..c_n = D_{Train}$. For parallelization, we assign different processes to each of the bags formed in Step-1. As these processes do not have any dependencies associated with each other, they run in parallel. Each process uses the classification algorithm K to generate the base classifier k_j with respect to the bag c_j. In the end, an ensemble Z is generated, which is a collection of all $k_j's$. Later, Z is used to classify query instances based on the aggregated decisions made by individual $k_j's$.

7.4.2 Parallelized A-Stacking

The algorithm for the parallel version of A-Stacking [6] is described in Algorithm 6. A-Stacking makes use of a clustering algorithm C to form clusters of instances and a meta-classifier M to combine the individual decisions made by base classifiers. Similar to

Algorithm 5: P-Bagging.

Data: training data $D_{Train} =< x_s, y_s >$, with f features and D samples

test data $D_{Test} =< x_t, y_t >$,

a classification algorithm K

number of base classifiers n

Result: ensemble classifier Z

1 uniformly sample train sets c_1 , c_2 , ., c_n from D_{train} each having f features and D samples;

2 **for** *j= 1 to n* **do**

3 /*run in parallel*/

4 $k_j \leftarrow K(c_j)$;

5 $Z \leftarrow \{k_1, k_2, .., k_n\}$;

6 Classify instances in D_{test}, using the statistical mode of all the classifications done by $\{k_1, k_2, ., k_n\}$ for each sample;

bagging, A-Stacking requires training data D_{Train} and test data D_{Test}. Along with this, it requires validation data D_{Valid} to decide the best classifier per cluster.

Algorithm 6 starts by forming n clusters of instances from D_{Train}. The clusters $\{c_1, c_2, ..c_n\}$ formed in Step-1 are used by different classification algorithms $\{L_1, L_2, .., L_m\}$ to generate a set of base classifiers $\{l_1, l_2, .., l_m\}$ per each cluster. To parallelize the algorithm, we assign different processes to each of the generated base classifiers. These processes operate separately and communicate the predicted decision in the end. In step-5 of the Algorithm 6, the performance of each base classifier is tested on D_{Valid} to select the best learning algorithm for each c_i according to the precision score of the algorithms, and the rest are discarded. Once the best classifier is selected for each cluster c_i, the individual classifier outputs on D_{Train} are used as inputs for the meta-classifier M which is responsible for classifying the test instances.

7.4.3 Parallelized Random-Subspace

The method of parallelizing Random Subspcace is described in Algorithm 7. Similar to other ensemble methods, it requires to generate an ensemble of base classifiers. These base classifiers are constituted by sampling f features from the original training set D_{Train}. The

Algorithm 6: P-AStacking.

Data: training data $D_{train} =< x_s, y_s >$ with f features and D samples

test data $D_{test} =< x_t, y_t >$

validation data $D_{valid} =< x_v, y_v >$

a clustering algorithm C

a set of classification algorithms $\{L_1 , L_2 , ., L_m\}$

a meta-classifier M

number of base classifiers n

Result: ensemble classifier Z

$\{c_1 , c_2 , , c_n\} \leftarrow C(D_{train})$

for *i= 1 to n* **do**

  /*run in parallel*/

  $\{l_i^1 , l_i^2 , ., l_i^n\} \leftarrow\{L_1, L_2, ., L_n\}$ (c_i);

  Check the accuracies $\{a_i^1 , a_i^2 , ., a_i^m\}$ of $\{l_i^1, l_i^2 , ., l_i^m\}$ on D_{valid};

  Select the best performing classifier l_i from $\{l_i^1 , l_i^2 , ., l_i^m\}$ and send it to M;

  Integrate the qualified base classifiers $\{l_1, l_2, ., l_n\}$ using the meta-classifier M;

  return the ensemble classifier Z integrated by M to classify the instances in D_{test}

Classify instances in D_{test}, using the output of the meta-classifier, which integrates the classifications done by $\{ l_1, l_2, ., l_n \}$ for each sample in D_{test};

target is to create n sets of instances $\{c_1, c_2, .., c_n\}$, where each set contains D instances and f features, but sets c_i and c_j may have different features. To achieve parallelization, multiple processes are assigned to sets of instances generated in the previous step. Each of these processes is responsible for generating a base classifier l_i by applying a learning algorithm L on the set with sampled features c_i*. Later, these processes communicate with their individual decisions and the final decision is made by the ensemble.

7.5 Experimental Setup, Analysis and Discussion

In this section, we describe the experimental results performed under various settings along with a discussion. We explore the proposed model's behaviour thoroughly by cre-

Algorithm 7: P-Random_Subspace.

Data: training data $D_{train} =< x_s, y_s >$ with f features and D samples

test data $D_{test} =< x_t, y_t >$

a classification algorithm L

number of base classifiers n

Result: ensemble classifier Z

1 uniformly sample train sets c_1 , c_2 , ., c_n from D_{train} each having D samples each containing a subset of all of features in D_{train};

2 **for** *i= 1 to n in parallel* **do**

3 /*run in parallel*/

4 from c_i sample [0,f] number of features into c_i*;

5 l_i = L (c_i*);

6 return the ensemble Z of $l_1, l_2, ., l_n$ to classify the instances in D_{test} ;

7 Classify instances in D_{test}, using the statistical mode of all the classifications done by $l_1, l_2, ., l_n$ for each sample, such that each classifier is fed only the features that it has been trained on;

ating multiple environments, such as within dataset, cross-dataset, and by limiting the number of tweets per user (to avoid user-bias). Next, we give details of the datasets used in the study:

7.5.1 Datasets

As mentioned earlier, there is a scarcity of annotated datasets for hate speech detection. SMP such as Facebook does not impose a restriction on the size of text, making it challenging to analyse. In this study, we focus on textual data from the micro-blogging site Twitter. The most popular datasets are Waseem and Hovy [45] and SemEval2019 [46]. Waseem and Hovy dataset comes with tweet identifiers along with their associated class labels, i.e., sexist, racist and non-hateful. The actual tweets can be extracted using any tweet crawler. 16K tweet identifiers constitute the dataset; the actual number of tweets was lesser than that. Despite its popularity, the dataset is highly biased. It has been observed that only a few users have communicated the majority of the hateful tweets in this dataset. Specifically, 65% of the hateful tweets are posted by only two users, which

Table 7.2: Description of Datasets.

Dataset	#Tweets	#Tweets Extracted	Labels	Class Distribution
Waseem and Hovy [45]	16,000	14,949	Sexist, Racist, Non-Hateful	Hateful: 4839 Non-Hateful: 10,110
SemEval 2019 [46]	9,000	9,000	Hateful, Non-Hateful	Hateful: 3783 Non-Hateful: 5217
$Covid-Hate_{HL}$ [191]	2,400	1,637	Hate, Neutral	Hate: 677 Neutral: 960
$Covid-Hate_{ML}$ [191]	30M	10,674	Hate, Neutral	Hate: 4,968 Neutral: 5,706
US Elections	1,105	1,105	Hate, Neutral	Hate: 665 Neutral: 440

causes a significant user-overfitting effect. Therefore, it is advised to restrict the number of tweets per user and then conduct another set of experiments. Ideally, if the model is capable of overcoming the user-overfitting effect, it must not observe a performance drop.

In addition, there was a massive loss of tweets when we extracted them using the identifiers. We crawled the tweets in mid-2020 and achieved only 10,147 tweets. Out of which, only four were from the "racist" class. While considering the three categories of the dataset, the results could not reflect the models' behaviours correctly. Therefore, we requested [55] to provide us with the extracted data used in their paper. The authors generously supported us and provided useful links.

SemEval 2019 dataset is from "Multilingual detection of hate speech against immigrants and women in Twitter" [46]. The dataset comprises 9K tweets with only two labels: hateful and non-hateful. It provides the tweets and the associated labels, but not the user identifiers. The user information is protected under the General Data Protection Regulation (EU GDPR).

For providing case studies on hate speech propagation during the ongoing pandemic and US elections 2020, we used some samples from the Covid-Hate dataset [191]. The

dataset contains a collection of tweets related to COVID-19[14]. A part of the dataset includes $2,400$ hand labelled tweets ($Covid-Hate_{HL}$). The other version consists of over $30M$ machine labelled tweets. We sampled around $10,000$ tweets ($Covid-Hate_{ML}$) to test our models' efficacy. Note that we do not consider the tweets belonging to 'counterhate' and 'non-Asian aggression' categories in this study. For the US election-related experiments, we handpicked some tweets from the Covid-Hate datasets related to the election. The resulting dataset is the combination of hand labelled and machine labelled tweets. In all these datasets, retweets are eliminated as they have little relevance in training the models.

The details of the datasets are given in Table 7.2.

7.5.2 Experimental Setup

Next, we describe the implementation details and experimental setup use for the study. As defined earlier, we work in two phases: phase-I makes use of RNNs to get word embeddings while considering the textual context. Phase-II uses the A-Stacking [6] model for classifying the input vectors into predefined labels. A-Stacking is an ensemble-based classifier that uses multiple base classifiers with a combination of a meta-classifier. In this study, we have used Support Vector Machine Classifier (SVM) [192], Gradient Boosting Decision Trees (GBDT) [53], Multi-Layer Perceptron Classifier (MLP) [193], kNeighbors Classifier [194], ELM classifier[15] along with Logistic Regression [137] as the meta-classifier. For clustering, we have used SimpleKMeans [111] clustering algorithm with varying values of k.

Evaluation Metrics

Precision: Precision is used for evaluating the correct positive predictions. It is calculated as the ratio of correctly classified positive instances to total predicted positive instances.

Recall: Recall is the fraction of total correctly predicted instances among all positive instances.

F1: F1-measure conveys the balance between precision and recall. It is calculated as

[14] `http://claws.cc.gatech.edu/covid/\#dataset`

[15] `https://scikit-elm.readthedocs.io/en/latest/user_guide.html`

$2 * ((precision * recall)/(precision + recall))$.

A macro-average computes the metric independently for each class and then takes the average, whereas a micro-average aggregates all classes' contributions to computing the average metric.

7.5.3 Results

We performed our experiments under three categories. Category-I is for testing the within dataset performance of the model, where we train and test the model on the same dataset. The dataset is partitioned into train and test sets using 10-fold cross-validation. Category-II explores the model's behaviour in the cross-dataset environment, where we train it on a dataset and test it on another dataset and vice-versa. We have designed Category-III for restricting the user distribution among the dataset so that we can avoid the user-overfitting effect. For this purpose, we have limited the number of tweets per user to 250. In general, the average number of tweets per day (TPD) is 4.42 for an active user[16]. Waseem & Hovy dataset [45] was collected in the span of two months. Therefore 250 is the ideal choice for such restriction. If a user constitutes more than 250 tweets in the dataset, we randomly select 250 of their tweets.

Category-I: Within Dataset Performance

In category-I, we test the models' behaviour following the within-dataset environment, where the train and the test sets are from the same dataset. We use 10-fold cross-validation for partitioning the data. The results of various models on Waseem & Hovy and SemEval 2019 datasets under category-I are given in Table 7.3. The best results for the proposed model were achieved while considering five clusters on (a) and ten clusters on (b). As shown in Table 7.3, the proposed model achieves a good performance but does not outperform both rivals on these datasets.

Table 7.4 shows the performance of various models on datasets related to COVID-19. The proposed model works better on machine labelled dataset with the highest F1 score reaching 89.90%. The proposed model outperforms rival-2 in all the cases but lags behind rival-1 with a small margin.

[16] `https://blog.hubspot.com/blog/tabid/6307/bid/4594/is-22-tweets-per-day-the-optimum`

Table 7.3: Performance evaluation of various models under within-dataset environment. (a) Waseem & Hovy, (b) SemEval 2019. The first row for each dataset represents the Micro average and the second represents the Macro average.

Method	Dataset	F1	Prec.	Rec.
Proposed Model	(a)	80.59	80.56	80.87
		74.20	75.61	73.15
	(b)	73.97	74.06	73.95
		73.31	73.29	73.39
[7]	(a)	80.70	82.30	82.10
		73.10	81.60	68.90
	(b)	75.26	75.30	75.32
		74.54	74.73	74.46
[8]	(a)	84.30	84.70	84.10
		79.60	78.00	81.70
	(b)	70.85	76.51	71.19
		71.81	74.40	70.96

Table 7.4: Performance evaluation of various models on COVID-19 datasets under within-dataset environment. (a) $Covid - Hate_{HL}$, (b) $Covid - Hate_{ML}$. The first row for each dataset represents the Micro average, and the second represents the Macro average.

Method	Dataset	F1	Prec.	Rec.
Proposed Model	(a)	77.85	77.88	78.26
		72.65	73.63	72.28
	(b)	89.90	89.97	89.89
		89.85	89.80	89.97
[7]	(a)	78.86	79.18	79.95
		73.22	76.62	71.83
	(b)	90.12	90.32	91.82
		90.53	90.32	91.34
[8]	(a)	58.66	50.10	70.76
		41.43	35.38	50.00
	(b)	89.65	89.70	89.67
		89.59	89.71	89.52

Table 7.5: Performance evaluation of the various models on US presidential election dataset. The first row for each method represents the Micro average and the second represents the Macro average.

Method	F1	Prec.	Rec.
Proposed	88.99	89.49	89.13
Model	88.37	89.49	87.97
[7]	78.86	79.18	79.95
	73.22	76.62	71.83
[8]	60.54	65.36	65.85
	56.64	64.88	59.66

Table 7.5 shows the performance of various models on US presidential election dataset. It is evident that the proposed model outperforms the rivals and achieve the highest F1 score of 88.99%.

Category-II: Cross Dataset Generalization

In category-II, we evaluate the performance of various models under the cross-dataset environment. It has been claimed in the past that for a hate speech detector, it is essential to perform reasonably well when tested on new data which were not seen while training the model. We design this setup by training the models on Waseem & Hovy dataset and testing them on SemEval 2019 dataset. For managing the compatibility issues, we merge the three classes of the Waseem and Hovy dataset into two: Hateful and Non-Hateful. It can be observed from Table 7.6 that the performance of the models degrade drastically and incurs a loss of about 20%. The proposed model performs adequately well and manages a good score.

In addition, we also tested the efficacy of the proposed model while using SemEval 2019 dataset for training and Waseem & Hovy dataset for testing. We achieve a better performance than the rivals, which is adequate for the cross-dataset generalization. The best results for the proposed model were achieved while considering ten clusters on both datasets.

Table 7.6: Performance evaluation of the models under cross-dataset environment. (a) Train: Waseem & Hovy, Test: SemEval 2019 (b) Train: SemEval2019, Test: Waseem & Hovy. The first row for each dataset represents the Micro average and the second represents the Macro average.

Method	Dataset	F1	Prec.	Rec.
Proposed Model	(a)	62.73	62.65	63.13
		61.45	61.85	61.36
	(b)	61.04	60.98	61.11
		55.34	55.36	55.33
[7]	(a)	61.85	62.07	62.84
		60.22	61.44	60.28
	(b)	60.66	60.01	61.60
		54.03	54.41	54.00
[8]	(a)	56.57	63.57	62.39
		53.28	63.9	56.8
	(b)	54.82	56.55	53.71
		49.81	50.15	50.16

Table 7.7: Performance evaluation of the proposed model while considering the user-distribution on Waseem and Hovy dataset. We have restricted the number of tweets per user to 250. The first row for each method represents the Micro average, and the second represents the Macro average.

Method	F1	Prec.	Rec.
Proposed	81.54	82.21	81.38
Model	74.09	72.69	76.32
[7]	82.13	82.16	82.96
	74.76	77.89	73.01
[8]	73.89	77.71	72.39
	67.11	66.63	70.76

Category-III: Controlling the User Overfitting Effect

As mentioned earlier, the available annotated datasets are highly biased, with only a few users constituting the majority of the hateful tweets [195]. Therefore, we claim that it is vital to highlight the model's behaviour while restricting the number of tweets per user. In this study, we limit the number of tweets to 250 per user in waseem and Hovy dataset and achieve 4,984 tweets. The drop in the number of tweets from 14,949 to 4,984 by only one restriction proves a strong user-bias presence. Moreover, after imposing the restriction, we could not get enough tweets belonging to the "racist" class (only 107). Therefore, we merged the three classes into two: Hateful and Non-hateful. The proposed adaptive model can manage the user overfitting effect. As shown in Table 7.7, there is no significant drop in the performance of the proposed model compared to the performance mentioned in Table 7.3. Therefore, we can establish that the proposed model is able to overcome the user-overfitting effect.

The performance results for the proposed parallel models under a within-dataset environment are described in Table 7.8. On SemEval 2019 dataset, the highest speedup is achieved by Bagging classifier with 20 processes, whereas on Waseem and Hovy dataset Random Subspace yields the best speedup and efficiency. 1‘

Table 7.8: Performance evaluation of serial and parallel versions of ensemble classifiers under within-dataset environment. The top row for each classifier denotes the weighted average and the bottom one represents the macro average.

Dataset	Method	F1	Prec.	Rec.	Ts	Tp	S	E
SemEval 2019	P-AStacking (20)	73.88 73.23	74.01 73.21	73.85 73.35	59.56	8.82	6.75	0.34
	P-Bagging (20)	74.05 73.40	74.16 73.37	74.01 73.51	119.47	12.85	9.30	0.46
	P-Random Subspace(20)	74.99 74.34	75.06 74.34	74.98 74.40	135.81	15.89	8.55	0.43
Waseem & Hovy	P-AStacking (10)	80.80 77.94	80.79 78.21	80.88 77.74	96.67	19.11	5.06	0.51
	P-Bagging (10)	81.89 79.16	81.84 79.55	82.00 78.83	135.60	23.66	5.73	0.57
	P-Random Subspace(10)	82.22 79.43	82.15 80.20	82.42 78.84	149.29	25.78	5.79	0.58

The cross-dataset evaluation results are given in Table 7.9. It can be observed from the table that Bagging is the best classifier in terms of speedup and efficiency for both dataset combinations.

The results for the third setup are given in Table 7.10. The target is to show that the performance is not affected while controlling the user-bias present in the dataset. Here also, Bagging proves to be the best classifier with speedup= 6.03 and efficiency = 0.60.

7.5.4 Discussion

We worked towards proposing an adaptive automatic hate speech detection model that can perform reasonably well on cross-datasets. The results reported in Tables 7.3-7.7 show that our proposed model is capable of adapting to the properties of data and behave accordingly when the test environment is changed. We emphasized that the available annotated datasets have a strong bias in them. For the correct assessment of the model, it is necessary to restrict the number of tweets per user. Therefore, we explored the model's ability to adapt to this change.

In the bar diagram represented in Figure 7.3, we show the percentage difference in the performance of various models when tested in the same environment v/s—tested in a cross environment. It is evident that the accuracy degrades significantly, but the proposed

Table 7.9: Performance evaluation of serial and parallel versions of ensemble classifiers under cross-dataset environment. The top row for each classifier denotes the weighted average and the bottom one represents the macro average.

Dataset	Method	F1	Prec.	Rec.	Ts	Tp	S	E
SemEval 2019 + Waseem & Hovy	P-AStacking (10)	61.04 55.34	60.98 55.36	61.11 55.33	72.42	18.94	3.82	0.38
	P-Bagging (10)	60.78 55.38	61.03 55.33	60.57 55.45	99.30	15.70	6.33	0.63
	P-Random Subspace(10)	61.16 55.71	61.31 55.68	61.01 55.77	102.87	16.32	5.77	0.58
Waseem & Hovy + SemEval 2019	P-AStacking (10)	62.73 61.45	62.65 61.85	63.13 61.36	122.24	27.49	4.45	0.44
	P-Bagging (10)	62.82 61.28	63.00 62.39	63.68 61.26	246.83	38.12	6.47	0.65
	P-Random Subspace(10)	61.82 60.06	62.29 61.75	63.08 60.02	254.69	44.10	6.30	0.63

Table 7.10: Performance evaluation of serial and parallel versions of ensemble classifiers while controlling the user-bias. The top row for each classifier denotes the weighted average and the bottom one represents the macro average.

Dataset	Method	F1	Prec.	Rec.	Ts	Tp	S	E
Waseem & Hovy (250)	P-AStacking (10)	81.39 74.10	81.25 75.64	81.88 73.11	31.11	6.37	4.88	0.49
	P-Bagging (10)	81.70 74.31	81.55 76.48	82.35 72.97	15.02	2.49	6.03	0.60
	P-Random Subspace(10)	82.01 74.61	81.94 77.45	82.80 72.97	15.40	2.75	5.61	0.56

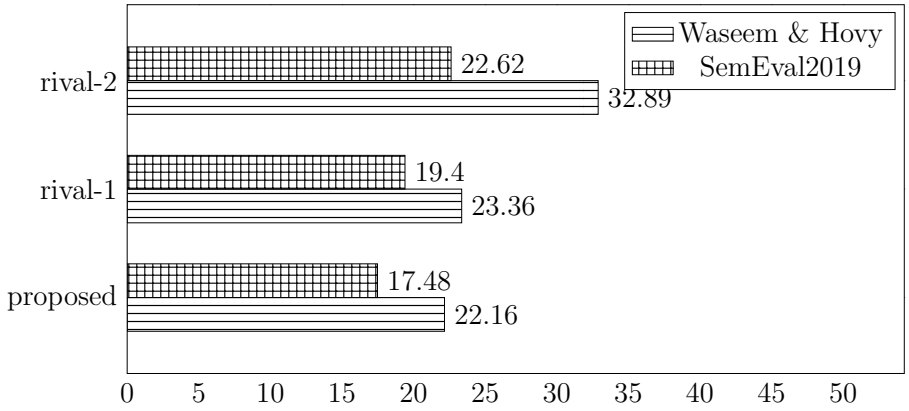

Figure 7.3: Calculating the percentage drop in F1-score (micro) of various models when tested under cross-dataset environment in comparison with within-dataset environment. The significant drop in performance justifies the need of cross-dataset generalization. rival-1= [7], rival-2= [8]

model shows the slightest deviation. Therefore, the model is able to overcome the poor generalization ability of the state-of-the-art.

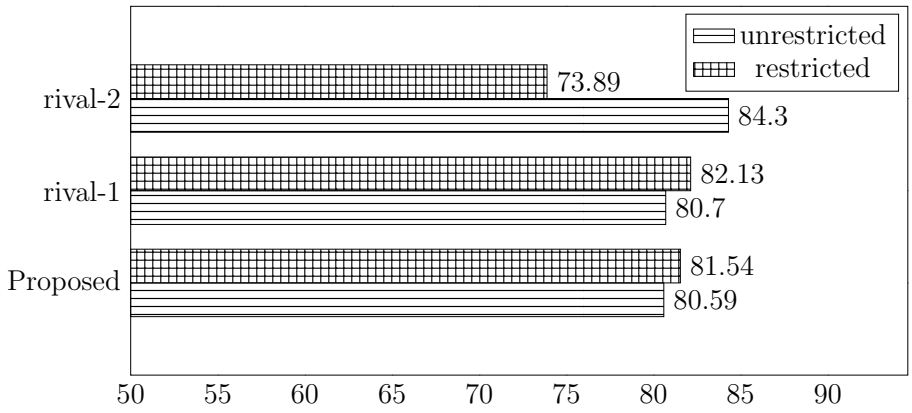

Figure 7.4: Performance comparison of various models on Waseem and Hovy dataset under two environments: dataset with no restrictions on the number of tweets and dataset with a cap of 250 tweets per user. We show the F1 score-micro values for this comparison. Proposed model observes no drop.

The bar diagram represented in Figure 7.4 shows the difference in the proposed model's performance when the dataset is not restricted v/s the performance after imposing the limit. It was a prerequisite for the hate speech detector to avoid a performance drop after

the limit is imposed. It is evident that our model shows no drop and performs reasonably well in the changing environment.

We are able to parallelize these models without any loss of accuracy and a significant speedup. We first perform a same-dataset performance evaluation of these models, as seen in Table 7.8. Here we see the Random subspace algorithm outperforming both the other algorithms on both datasets. However, in the cross dataset testing (Table 7.9), we find that Bagging has the highest F1 score for WaseemHovy-SemEval2019 evaluation while Random Subspace again has the highest F1 score in SemEval2019-WaseemHovy evaluation. Performance of all models significantly increases when correcting for user bias (i.e. include only 250 tweets from any particular user) as can be seen in Table 7.10. Here as well, we see Random Subspace outperforming the other models. We observe that the speedup obtained by data splitting, i.e. increasing the number of classifiers increases for all the three models but the efficiency decreases.