

# Multi-level stacked ensemble learning for identifying hate speech spreaders on Twitter

Notebook for PAN at CLEF 2021

Darko Tosev<sup>1</sup> and Sonja Gievska<sup>1</sup>

<sup>1</sup> Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University, 16 Rugjer Boshkovikj St, Skopje, 1000, Republic of North Macedonia

## Abstract

There are growing signs of discontent with the anti-social behavior expressed on social media platforms. Harnessing the power of machine learning for the purpose of detecting and mediating the spread of malicious behavior has received a heightened attention in the last decade. In this paper, we report on an experiment that examines the predictive power of a number of sparse and dense feature representations coupled with a multi-level ensemble classifier. To address the research questions, we have used PAN 2021 Profiling Hate Speech Spreaders on Twitter task for English language. The initial results are encouraging pointing out to the robustness of the proposed model when evaluated on the test dataset.

## Keywords

Hate speech spreaders detection, Ensemble learning, Feature vector representation, Twitter, English

## 1. Introduction

Users' opinions and beliefs are on full display on social media platforms [24], where freedom of expression, diversity and inclusion should be at the forefront. Harnessing user-generated text on the web has been the playground for social analytics. It usually implies processing of linguistic information that sheds light on the complex phenomena of human nature and behavior. Regrettably, the platforms have been used for spreading fake news, abusive language and hate speech by people that hide behind the anonymity that the web offers [5]. And as potentially harmful antisocial behavior polarized public opinion in recent times, the demands for swift reactions from social media platforms are growing louder.

Analysis of anti-social behavior has emerged as a critical requirement for achieving a competitive edge in social media. A number of problems fall under the umbrella of anti-social behavior, including identifying one's prejudice expressed towards an individual or a particular social group. Hate speech in online text refers to expressing someone's view by using abusive and threatening language toward an individual or a social group, usually targeted on the basis of race, ethnicity, religion, gender, or sexual orientation.

The major limitation in the empirical research on detecting anti-social individual and collective behavior lies in the failure to construct benchmarking datasets for detecting spreaders that simulate scenarios comparable to realistic situations. The nuance and subtlety of language and word choices someone makes is just one aspect of someone's hostile and threatening attitude toward others. Insights into potential spreader's personality and their online profile, the history of their behavior and structural properties of their interaction exchanges and relationships on social media [14] might further augment the predictive models.

---

<sup>1</sup>CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania  
EMAIL: tosev.darko@students.finki.ukim.mk; sonja.gievska@finki.ukim.mk

The dataset used in our study was offered by PAN's 2021 Author Profiling task [2] - Profiling Hate Speech Spreaders on Twitter [18]. The research objective was to determine whether the author of a given Twitter feed has been spreading hate speech or not. The task was multilingual, namely, two datasets for independent analysis were offered, one in English and the other in Spanish language<sup>2</sup>. In our research, we have addressed the problem of detecting hate speech spreaders writing tweets in English.

Extracting social knowledge and meaning from users' contributions poses recurrent challenges for natural language understanding and machine learning. An extensive body of research has been done to address the issues of detecting abusive language and hate speech. The integration of a number of linguistic features, such as lexicons, psycholinguistic features, n-grams, hashtags and different types of distributed word embeddings has facilitated the process of unravelling the linguistic indicators of abusive language and hate speech in online posts [10, 20]. This task has been approached from noticeable distinct positions, including some researchers advocating the use of deep learning models as well-suited approach for experimental investigation on unlabeled and large datasets [10, 20].

After a brief summary of related research presented in Section 2, we introduce the datasets and present the details of the adopted methodology in Section 3. The findings and the interpretation of the results are presented in Section 4, while the last section concludes the paper and points to direction for future research.

## 2. Related work

Studying online human behavior is recurrently challenged by the gaps in the knowledge and models of the processes and driving forces underlying it. The present work draws on previous research on detecting abusive language and hate speech in online posts. The inherent ambiguity of language and the subjectivity of people's perceptions i.e., what should be considered abusive or hate speech, have resulted in various classification schemes and scoring methods to be proposed [19, 23]. The terms such as: abusive, hostile, discriminative, offensive, cyberbullying, and profanity language have been associated with the wide-spread problem under investigation.

The scarcity of large realistic datasets labeled for ground-truth presents a bottleneck in studying language indicators of anti-social behavior. Crowdsourcing, users' feedback and ratings as well as semi-supervised learning have been utilized to remedy the problem. The variety of proposed models i.e., sets of linguistic features and classification algorithms they rely upon, matches the subtle differences in the tasks and data at hand.

Language modeling using n-grams are frequently used in automatic hate speech detection and a number of tasks closely related to it [5, 6, 8, 11, 12, 15, 16]. Other low-level features besides words and characters [5, 6, 12, 15], such as part-of-speech tags [8, 16] and dependency relationships [5, 6, 7, 11, 16] have also been utilized for predictive modeling.

Various types of lexicons and dictionaries have been used to identify the presence of words representing different forms of verbal abuse, such as: insults, swear words and profanities [5, 6, 7, 11, 16, 25, 27]. The empirical evidence highlights the relevance of psycholinguistic features as linguistic indicators for detecting abusive or hate language in online text. Frequencies of personal pronouns, verbs or nouns have been prominently used in the predictive models [11, 16].

A number of researchers have endeavored to improve the classification accuracy by incorporating sentiment analysis in their modelling approach. A text expressing hateful and threatening language toward a targeted person or group is more likely to contain high frequency of negative sentiment words, so sentiment-related features extracted from variety of sentiment lexicons often complement the feature sets [4, 8, 11, 13, 21]. Sentiment-based analysis was employed to filter out objective tweets before proceeding with hate speech detection analysis on the subset of subjective tweets only [11].

A variety of machine learning algorithms have been employed for abusive and hate speech detection, such as Support Vector Machines (SVM) [1, 6, 12, 22], Random forests [1, 6] and Logistic regression [1, 9]. Current technological trends in deep learning has real potential for advancing the endeavor of detecting online malicious behavior, although the supervised approaches face the challenge of learning

---

<sup>2</sup> [https://zenodo.org/record/4603578#.YK\\_K2bczbIU](https://zenodo.org/record/4603578#.YK_K2bczbIU)

from rather small currently available datasets [1, 22, 26]. Integration of distributed word or paragraph representations (embeddings) regularly used in deep learning approaches have been related to performance gains on a number of datasets [1, 9, 22, 26].

Our modeling approach on the task of identifying users spreading hateful tweets was inspired by the work of Buda and Bolonyai [3]. Our research also adopts the ensemble learning approach to avoid overfitting, however the number and the type of features in our model differs. Aiming for building a more robust and generalized model, we propose a multi-level stacked ensemble model that serves both purposes, dealing with curse of dimensionality as well as gaining better predictive performance.

### 3. Methodology

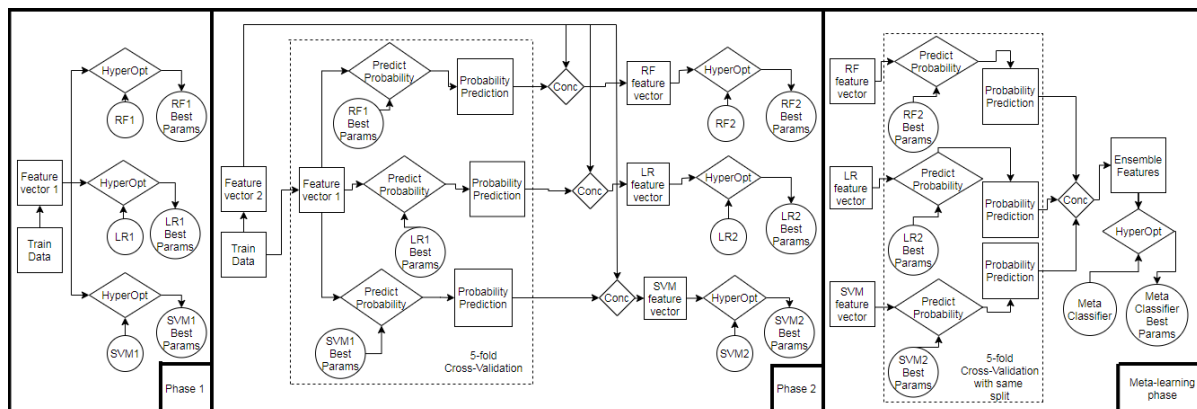
To address the research question of using linguistic analysis for automatically detecting hate speech spreaders, we have conducted extensive experiments on a diverse set of features and various types of traditional machine learning algorithms stacked as a multi-level ensemble learning model. After short introduction of the dataset on which our model was trained and evaluated, this section offers details of the proposed model and the phases in which the training of the multi-level ensemble model was conducted.

#### 3.1. Dataset

The dataset used in our study was made publicly-available for the PAN 2021 Profiling Hate Speech Spreaders on Twitter task. The training dataset contains Twitter feeds (200 tweets) of 200 users classified as hate speech spreaders or not. Some preprocessing of the training dataset has been done by the organizers, namely, for each tweet, the URLs, hashtags and user mentions were substituted with standardized tokens.

#### 3.2. Multi-level ensemble learning: Phase 1

The multi-level stacked ensemble model proposed in this research is presented in Figure 1. The inclusion of several machine learning classifiers in a stacked architecture that uses optimization at each level serve two purposes: 1) dimensionality reduction and 2) increased predictive performance.



**Figure 1:** Multi-level ensemble learning model. (*HyperOpt* denotes hyper-optimization, *Conc.* denotes vertical concatenation, *Best Params* are the parameters chosen after hyperparameter optimization. A shorthand names of the classifiers *SVM*, *RF*, *LM* are used for Support vector machine, Random forest and Logistic regression, respectively. Enumerations corresponds to the phase in the training.)

At the first level of our multi-level ensemble learning, the low-level sparse feature vectors were generated. A variety of features were extracted from the training data, following some suitable preprocessing steps. A feature vector that represent each user's Twitter feed was generated, denoted as

*Feature vector 1* in Fig. 1. Three classifiers, SVM, Logistic regression and Random forests were used at this level of training – Phase 1.

### 3.2.1. Sparse feature representation

A description of sparse n-gram-related features used at the first level of the ensemble model follows.

- **TF-IDF weighted n-grams** (unigrams, bigrams and their combination) were used and optimized for each classifier. The preprocessing techniques preceding the weighted TF-IDF extraction were: 1) expansion of shortened form of words, 2) substitution of emojis and emoticons with appropriate words or phrases, 3) removal of Twitter-specific elements (e.g., URLs, hashtags, user mentions and number of retweets - RTs), 4) conversion of characters to lower case letters, and 5) removal of non-alphabetical characters. In the case of unigrams, stop words were ignored and skipped. Each TF-IDF vector on a user level was normalized to be of unit length.
- **TF-IDF character n-grams** of two-, three and four-characters lengths were generated in a similar fashion. The minimum overall document frequency of character n-grams were optimized for each classifier. Converting the characters into lower case letters was the only preprocessing step performed. Each TF-IDF vector was normalized. The use of character n-grams was twofold: 1) to mitigate the effect of spelling variations, and 2) to take into account characters such as: numbers, punctuation marks and emojis that were removed by preprocessing steps in other vector representations.
- **Typed dependencies** were extracted using Stanza’s dependency parser, after preprocessing steps for substituting emojis and emoticons, and removing Twitter-related elements (e.g., URLs, hashtags, user mentions and RTs). All available typed dependencies (65 without the “ROOT” dependency) were used as a vocabulary and a vector of their frequencies for each user’s feed were constructed. Each vector was normalized to be of unit length. The rationale for the inclusion of deeper syntactic information that captures long-distance relationships was guided by previous research pointing to performance advantage when dependency structure is coupled with simple n-grams [5, 6].

### 3.2.2. Optimization Phase 1

At the first level (Phase 1) of our multi-level ensemble learning, the low-level sparse feature vectors were passed through each of the three classifiers (the leftmost pane labeled **Phase 1** in Fig 1). The hyperparameters of the classifiers were optimized – their names and the values that were explored are shown in Table 1 in shaded rows. The feature vectors of the model at this level have been selected by using a cross-validated grid search. Table 1 contains the list of features and values we have experimented with (non-shaded rows). The parameters and features not included in the table were set to their default values.

The hyperparameters of the classifiers and sparse feature vectors of the best performing model chosen during the optimization process in Phase 1 are shown in Table 2.

### 3.3. Multi-level ensemble learning: Phase 2

At the next level of the ensemble learning, the model was extended with a new set of sentiment-related and psycholinguistic features. 5-fold cross-validation was used to prevent overfitting. The same types of classifiers as in Phase 1 were used, namely, SVM, Random forest and Logistic regression classifiers. The output of the classifiers at the first level i.e. the probability predicted by each classifier has been concatenated with the new feature set as shown the middle pane labeled Phase 2 in Figure 1.

**Table 1**

Optimization of each classifier's hyperparameters and the features of the model in Phase 1. The parameters not included in the list have default values.

Classifier	Parameter	Values
SVM 1	Regularization coef. ( <i>C</i> )	{1, 10, 100, 1000}
	Word n-gram order ( <i>w_ord</i> )	{{(1,1), (1,2), (2,2)}
	Word n-gram minimum document frequency ( <i>w_min</i> )	{3, 4, 5, 6, 7}
	Character n-gram order ( <i>c_order</i> )	{{(2, 2), (3, 3), (4, 4)}
	Character n-gram min. doc. frequency ( <i>c_min</i> )	{3, 4, 5, 6, 7}
Logistic regression 1	Regularization coef. ( <i>C</i> )	{0.1, 1, 10, 100, 1000}
	Word n-gram order ( <i>w_ord</i> )	{{(1,1), (1,2), (2,2)}
	Word n-gram minimum document frequency ( <i>w_min</i> )	{4, 5, 6, 7}
	Character n-gram order ( <i>c_order</i> )	{{(2, 2), (3, 3), (4, 4)}
	Character n-gram min. doc. frequency ( <i>c_min</i> )	{6, 7, 8, 9}
Random forest 1	Number of trees in the forest ( <i>n_estimators</i> )	{100, 300, 400}
	Minimum number of cases on each leaf ( <i>min_samples_leaf</i> )	{7, 8, 9, 10}
	Word n-gram order ( <i>w_ord</i> )	{{(1,1), (1,2), (2,2)}
	Word n-gram minimum document frequency ( <i>w_min</i> )	{4, 5, 6}
	Character n-gram order ( <i>c_order</i> )	{{(2, 2), (3, 3), (4, 4)}
	Character n-gram min. doc. frequency ( <i>c_min</i> )	{4, 5, 6}

**Table 2**

Optimized hyperparameters and features of the best performing model in Phase 1.

Classifier	Optimized parameter and feature values
SVM 1	$C = 1, w\_ord = (1,1), w\_min = 5, c\_order = (2, 2), c\_min = 5$
Logistic regression 1	$C = 100, w\_ord = (1,2), w\_min = 5, c\_order = (3, 3), c\_min = 7$
Random forest 1	$min\_samples\_leaf = 9, n\_estimators = 400, w\_ord = (1,1), w\_min = 4, c\_order = (4, 4), c\_min = 6$

### 3.3.1. Lexicons

For the purposes of sentiment analysis and hateful terms recognition, a couple of lexicons were used. Hatebase<sup>3</sup> database was used as a dictionary of English words carrying hateful connotation.

A number of publicly-available sentiment lexicons, namely, NRC-VAD<sup>4</sup>, AFINN-111<sup>5</sup> and SentiStrength<sup>6</sup>, were utilized for the purpose of this research. Since, NRC-VAD (Valence, arousal and dominance lexicon) has valence scores ranging from 0 to 1, while the values in the AFINN-111 and SentiStrength range from -5 to 5, all scores were normalized in the range [0, 1]. The NLTK's

<sup>3</sup> <https://hatebase.org/>

<sup>4</sup> <https://saifmohammad.com/WebPages/nrc-vad.html>

<sup>5</sup> <http://www2.imm.dtu.dk/pubdb/pubs/6010-full.html>

<sup>6</sup> <http://sentistrength.wlv.ac.uk/#Download>

SentiWordNet lexicon was also integrated, which includes a positive and a negative score for each word, both ranging between 0 and 1. The overall score for a word was calculated as the difference between the positive and negative score, and then scaled into the range [0, 1]. Since a word might be present in more than one lexicon, we have given priorities to the lexicons. The presence of a word was searched in the lexicons in the following order, SentiWordNet, followed by NRC-VAD and AFINN-111, while SentiStrength was at the end of the priority list.

### 3.3.2. Dense feature representation

A number of sentiment and hateful lexicon-related features were used in Phase 2 of the learning process. At the onset of this phase, few preprocessing steps were taken: 1) expanding of contractions, 2) substituting emojis and emoticons with words or phrases, 3) removal of Twitter-specific elements (e.g., URLs, hashtags, user mentions and RTs), 4) converting of all characters to lower case letters, and 5) removal of non-alphabetical characters.

Different wordforms contained in the lexicons were handled in the following way. If a word is found in a lexicon, then its valence values were retrieved and used to calculate several sentiment-related frequencies. If the word was not found, then its lemma or stem is tried to match the lexicon items. Lemmatization, stemming and POS tagging were done with NLTK's WordNetLemmatizer, Porter stemmer and pos tagger. Similar procedure was used to check the words presence in the hateful lexicon. Valence shifting related to negations was taken into account; the valence of a word proceeded by a negation word (e.g., not, none, never) was switched to the valence of opposing emotion<sup>7</sup>. The following lexicon-related features were taken into account:

- **Number of positive words.** A word is considered positive if it has a sentiment value above 0.5.
- **Number of negative words.** A word is considered negative if it has a sentiment value below 0.5.
- **Number of neutral words.** A word is considered neutral if it has a sentiment value equal to 0.5.
- **Number of unknown words.** A word is considered unknown if it is not present in any of the lexicons.
- **Overall sentiment mean value and standard deviation.** For every user in the dataset, the mean value and standard deviation of the overall sentiment polarity was calculated as the sum of the sentiment values of the positive and negative words contained in each tweet of the author.
- **Number of hateful words.**

The following stylistic features were calculated for each user (their entire Twitter feed) without any preprocessing:

- **Word and character lengths mean value.**
- **Word and character lengths standard deviation.**
- **Word and character minimum and maximum lengths.**
- **Word and character lengths range.**
- **Number of Twitter-specific elements:** URLs, hashtags, user mentions and retweets were taken on a user level.
- **Type-token ratio** is a stylistic feature, measuring the lexical diversity of the style of each author.

### 3.3.3. Optimization Phase 2

The middle pane labeled **Phase 2** in Figure 1 shows the next phase in the ensemble learning process. The same type of classifiers, SVM, Logistic regression and Random forests were used at this level of training.

---

<sup>7</sup> <http://sentiment.christopherpotts.net/lingstruc.html>

The input of each classifier represent a concatenated vector consisting of two elements: 1) the output of the best performing model (classifiers) in Phase 1 (their predicted probability) and 2) dense lexicon-related feature vector, denoted as *Feature vector 2* in Figure 1.

A 5-fold cross-validation was used to avoid overfitting of the model. The optimization of each classifier’s hyperparameters and the grid search optimization of the selected features has been performed. Table 3 shows the parameter values we have experimented with, while Table 4 contains the hyperparameter of the best performing model in Phase 2.

**Table 3**

Optimization of each classifier’s hyperparameters and the features of the model in Phase 2. The parameters not included in the list have default values.

Classifier	Parameter	Values
SVM 2	Regularization coef. (C)	{1, 10, 100, 1000}
Logistic regression 2	Regularization coef. (C)	{0.1, 1, 10, 100, 1000}
Random forest 2	Number of trees in the forest ( <i>n_estimators</i> )	{100, 300, 400}
	Minimum number of cases on each leaf ( <i>min_samples_leaf</i> )	{5, 6, 7, 8, 9, 10}

**Table 4**

Optimized hyperparameters and features of the best performing model in Phase 2.

Classifier	Optimized parameter and feature values
SVM 2	$C = 1$
Logistic regression 2	$C = 10$
Random forest 2	$min\_samples\_leaf = 8, n\_estimators = 300$

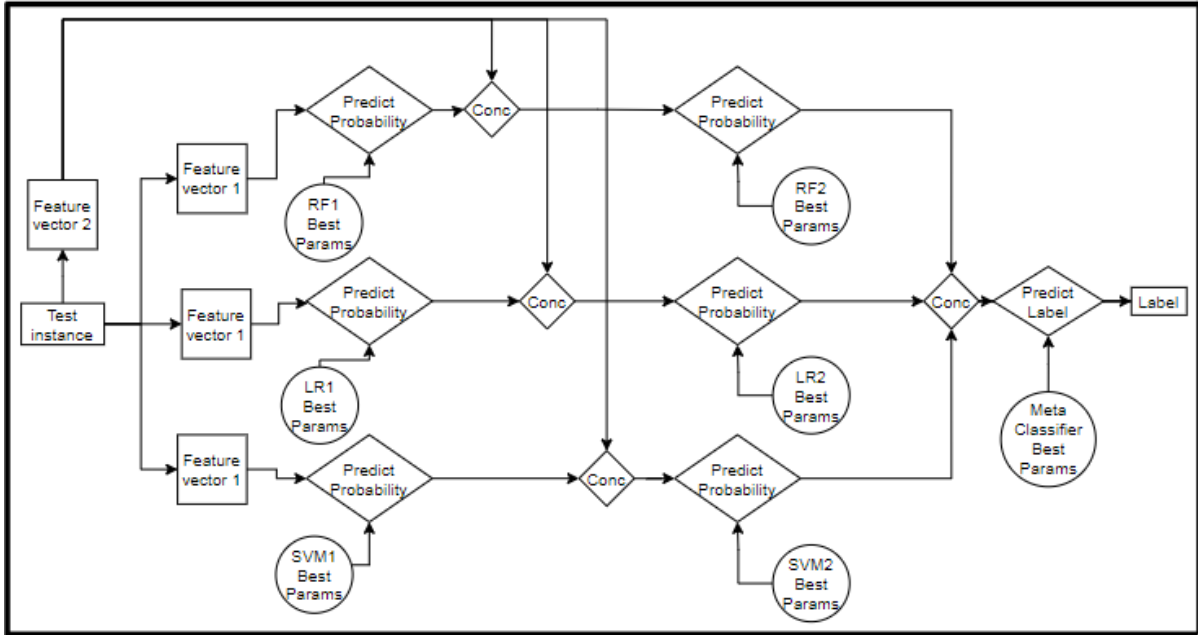
### 3.4. Multi-level ensemble learning: Meta-learning Phase

The final phase of the proposed multi-level ensemble model learning, shown in the rightmost pane labeled **Meta-learning phase** in Fig. 1, employs four classifiers: Majority, Nearest centroid, Logistic regression and the Ridge classifier. The probability predictions of the best performing classifiers of the previous Phase 2, were fed into the meta-learning classifiers. The four meta-learner classifiers were trained on the training set and tested on the development set. Instead of splitting the training dataset, 5-fold cross-validation was used.

The hyperparameters of the Logistic regression and Ridge classifiers were optimized using 5-fold cross-validation grid search. The best performing meta classifier was the Ridge classifier, which was selected as a meta-learner classifier in the final version of our stacked ensemble model. The best values of the Ridge classifier’s hyperparameters were the default ones, with the exception of alpha (best performance for value 1).

### 3.5. Prediction with the ensemble model

The trained multi-level ensemble model was tested on the test dataset through the TIRA platform [17], in the manner shown in Figure 2. The sparse feature representation (*Feature vector 1*) were generated for each test data sample (200 tweets for each user) and fed into the three classifiers (Phase 1) to get the initial probability predictions. The output of the first phase of the learning process were concatenated with *Feature vector 2*, which was extracted from each test sample. The concatenated dense vectors served as input of the classifiers of the second phase that produced new probability predictions. At the final phase, the meta-learner classifier predicted the label of the test sample. During testing, the mean and standard deviations of the trained model features were used.



**Figure 2:** Prediction with the trained multi-level ensemble model. (*HyperOpt* denotes optimization of hyperparameters, *Conc.* denotes vertical concatenation, *Best Params* stands for the best parameters chosen after hyperparameter optimization. A shorthand names of the classifiers *SVM*, *RF*, *LM* are used for Support vector machine, Random forest and Logistic regression, respectively. Enumerations denote the phase in the training in which a particular classifier and model belongs.)

## 4. Results

The evaluation metric proposed by the organizers of the PAN 2021 Profiling Hate Speech Spreaders on Twitter task was accuracy, so we report on the findings of the training and testing of our ensemble model in Table 5. The performance during training is given as a cross-validated average accuracy of the best performing version for each classifier in the ensemble independently. The accuracy of the best performing ensemble model obtained on the development and the testing dataset are also shown.

**Table 5**

Performance results on the training and testing dataset. Enumeration in classifier names denotes their level/phase in the multi-level ensemble model.

Model	Dataset	Accuracy
SVM 1	Training dataset	65.9%
SVM 2		64%
RF 1		69%
RF 2		67%
LR 1		65%
LR 2		65.9%
Multi-level Ensemble		62%
Multi-level Ensemble	Test set	70%

It should be noted that the variations of  $\pm 5\%$  in the performance of the classifiers observed during training, strongly suggest that some samples in the training dataset were more informative than others. We could hypothesize that a larger training dataset would stabilize the predictions and improve the performance of the ensemble model.

The unexpected degradation of the performance between phases yielded by the SVM and Random forest classifiers should be further investigated. The performance of the third Logistic regression



classifier was improved, pointing out that future exploratory analysis on the selection of feature sets might improve the overall quality of the ensemble. Furthermore, the lower accuracy results obtained by the ensemble model on the development set, indicate some future research direction for improvement.

The accuracy of 70% obtained on the unseen instance during testing, shows that the multi-level ensemble model outperforms the average accuracies obtained during training, pointing to the generalization power of the model.

## 5. Conclusions

This paper aims to contribute to research pertaining to detecting anti-social behavior, in particular spreading hate speech on social networks. A variety of sparse and dense feature representations have been explored, and used in conjunction with a multi-level stacked ensemble learning. The accuracy of 70% yielded on the test set is a promising evidence of the generalizability of the model.

Motivated by the insights gathered during our experimental study, we select few avenues for future research that are worth exploring. In particular, alternative combinations of preprocessing steps, feature sets, and representational models should be experimented with. The inclusion of additional psycholinguistic features, such as: the frequencies of using personal pronouns, verbs, articles, etc. might be more informative in modeling a particular user's characteristic and behavior. Consideration of transfer learning, pretrained vector embeddings and other machine learning methods including deep learning are expected to further advance the robustness of the model.

## 6. References

- [1] P. Badjatiya, S. Gupta, M. Gupta, V. Varma, Deep Learning for Hate Speech Detection in Tweets, in: Proceedings of the 26th International Conference on World Wide Web Companion, WWW '17 Companion, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2017, pp. 759–760. doi:10.1145/3041021.3054223.
- [2] J. Bevendorff, B. Chulvi, G. L. D. L. P. Sarracén, M. Kestemont, E. Manjavacas, I. Markov, M. Mayerl, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, E. Zangerle, Overview of PAN 2021: Authorship Verification, Profiling Hate Speech Spreaders on Twitter, and Style Change Detection, in: K. S. Candan, B. Ionescu, L. Goeuriot, B. Larsen, H. Müller, A. Joly, M. Maistro, F. Piroi, G. Faggioli, N. Ferro (Eds.), 12th International Conference of the CLEF Association (CLEF 2021), Springer, 2021.
- [3] J. Buda, F. Bolonyai, An Ensemble Model Using N-grams and Statistical Features to Identify Fake News Spreaders on Twitter, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névéol (Eds.), Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020, volume 2696 of *CEUR Workshop Proceedings*, CEUR-WS.org,2020. URL: [http://ceur-ws.org/Vol-2696/paper\\_189.pdf](http://ceur-ws.org/Vol-2696/paper_189.pdf).
- [4] P. Burnap, O. F. Rana, N. Avis, M. Williams, W. Housley, A. Edwards, J. Morgan, L. Sloan, Detecting tension in online communities with computational Twitter analysis, *Technological Forecasting and Social Change* 95 (2015) 96–108. doi:10.1016/j.techfore.2013.04.013.
- [5] P. Burnap, M. L. Williams, Cyber Hate Speech on Twitter: An Application of Machine Classification and Statistical Modeling for Policy and Decision Making, *Policy & Internet* 7 (2015) 223–242. doi:10.1002/poi3.85.
- [6] P. Burnap, M. L. Williams, Us and them: identifying cyber hate on Twitter across multiple protected characteristics, *EPJ Data Science* 5 (2016) 1–15. doi:10.1140/epjds/s13688-016-0072-6.
- [7] Y. Chen, Y. Zhou, S. Zhu, H. Xu, Detecting Offensive Language in Social Media to Protect Adolescent Online Safety, in: 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Conference on Social Computing, 2012, pp. 71–80. doi:10.1109/SocialCom-PASSAT.2012.55.
- [8] K. Dinakar, B. Jones, C. Havasi, H. Lieberman, R. Picard, Common Sense Reasoning for Detection, Prevention, and Mitigation of Cyberbullying, *ACM Trans. Interact. Intell. Syst.* 2(2012). doi:10.1145/2362394.2362400.

- [9] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, N. Bhamidipati, Hate Speech Detection with Comment Embeddings, in: Proceedings of the 24th International Conference on World Wide Web, WWW '15 Companion, Association for Computing Machinery, 2015, pp. 29–30. doi:10.1145/2740908.2742760.
- [10] P. Fortuna, S. Nunes, A Survey on Automatic Detection of Hate Speech in Text, ACM Comput. Surv. 51 (2018). doi:10.1145/3232676.
- [11] N. D. Gitari, Z. Zuping, H. Damien, J. Long, A Lexicon-based Approach for Hate Speech Detection, International Journal of Multimedia and Ubiquitous Engineering 10 (2015) 215–230. doi:10.14257/ijmue.2015.10.4.21
- [12] E. Greevy, A. F. Smeaton, Classifying Racist Texts Using A Support Vector Machine, in: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '04, Association for Computing Machinery, 2004, pp. 468–469. doi:10.1145/1008992.1009074.
- [13] C. V. Hee, E. Lefever, B. Verhoeven, J. Mennes, B. Desmet, G. D. Pauw, W. Daelemans, V. Hoste, Detection and Fine-Grained Classification of Cyberbullying Events, in: Recent Advances in Natural Language Processing, RANLP 2015, 7-9 September, 2015, Hissar, Bulgaria, RANLP 2015 Organising Committee / ACL, 2015, pp. 672–680.
- [14] A. Janchevski, S. Gievska, AndrejJan at SemEval-2019 Task 7: A Fusion Approach for Exploring the Key Factors pertaining to Rumour Analysis, in: Proceedings of the 13th International Workshop on Semantic Evaluation, Association for Computational Linguistics, 2019, pp.1083–1089. doi:10.18653/v1/S19- 2190.
- [15] Y. Mehdad, J. Tetreault, Do Characters Abuse More Than Words?, in: Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Association for Computational Linguistics, 2016, pp. 299–303. doi:10.18653/v1/W16- 3638.
- [16] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, Y. Chang, Abusive Language Detection in Online User Content, in: Proceedings of the 25th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2016, pp. 145–153. doi:10.1145/2872427.2883062.
- [17] M. Potthast, T. Gollub, M. Wiegmann, B. Stein, TIRA Integrated Research Architecture, in: N. Ferro, C. Peters (Eds.), Information Retrieval Evaluation in a Changing World, The Information Retrieval Series, Springer, Berlin Heidelberg New York, 2019. doi:10.1007/978-3-030-22948-1\_5.
- [18] F. Rangel, G. L. D. L. P. Sarracén, B. Chulvi, E. Fersini, P. Rosso, Profiling Hate Speech Spreaders on Twitter Task at PAN 2021, in: G. Faggioli, N. Ferro, A. Joly, F. P. Maria Maistro (Eds.), CLEF 2021 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2021.
- [19] B. Ross, M. Rist, G. Carbonell, B. Cabrera, N. Kurowsky, M. Wojatzki, Measuring the Reliability of Hate Speech Annotations: The Case of the European Refugee Crisis, NLP4CMC III:3rd Workshop on Natural Language Processing for Computer-Mediated Communication, 22 September 2016 (2016) 6–9. doi:10.17185/dupublico/42132.
- [20] A. Schmidt, M. Wiegand, A Survey on Hate Speech Detection using Natural Language Processing, in: Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, Association for Computational Linguistics, 2017, pp. 1–10. doi:10.18653/v1/W17- 1101.
- [21] S. O. Sood, E. F. Churchill, J. Antin, Automatic Identification of Personal Insults on Social News Sites, Journal of the American Society for Information Science and Technology 63 (2012) 270–285. doi:10.1002/asi.21690.
- [22] F. D. Vigna, A. Cimino, F. Dell'Orletta, M. Petrocchi, M. Tesconi, Hate me, hate me not: Hate speech detection on Facebook, in: Proceedings of the First Italian Conference on Cybersecurity (ITASEC17), Venice, Italy, January 17-20, 2017, volume 1816 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2017, pp. 86–95. URL:<http://ceur-ws.org/Vol-1816/paper-09.pdf>.
- [23] Z. Waseem, Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter, in: Proceedings of the First Workshop on NLP and Computational Social Science, Association for Computational Linguistics, 2016, pp. 138–142. doi:10.18653/v1/W16- 5618.
- [24] M. Wendling, 2015: The year that angry won the internet, 2015. URL: <http://www.bbc.com/news/blogs-trending-35111707>.

- [25] G. Xiang, B. Fan, L. Wang, J. Hong, C. Rose, Detecting Offensive Tweets via Topical Feature Discovery over a Large Scale Twitter Corpus, in: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, Association for Computing Machinery, 2012, pp. 1980–1984. doi:10.1145/2396761.2398556.
- [26] S. Yuan, X. Wu, Y. Xiang, A Two Phase Deep Learning Model for Identifying Discrimination from Tweets, in: Proceedings of the International Conference on Extending Database Technology, 2016, pp. 696–697. doi:10.5441/002/edbt.2016.92.
- [27] H. Zhong, H. Li, A. C. Squicciarini, S. M. Rajtmajer, C. Griffin, D. J. Miller, C. Caragea, Content-Driven Detection of Cyberbullying on the Instagram Social Network, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI'16, AAAI Press, 2016, pp. 3952–3958.