ISSN 1533-9211

DOI 10.17605/OSF.IO/MKCHU

BULLYING TEXT ON TAMIL-ENGLISH COMMENTS IS CLASSIFIED USING ENHANCED FEATURE EXTRACTION AND HYBRID FEATURE SELECTION TECHNIQUES

V. INDUMATHI[1] and Dr. S. SANTHANAMEGALA[2]

[1]Research Scholar, Assistant Professor, School of Computer Studies, Rathnavel Subramaniam College of Arts and Science, Coimbatore. Email: indhumathi@rvsgroup.com
[2]Assistant Professor, School of Computer Studies, Rathnavel Subramaniam College of Arts and Science, Coimbatore. Email: santhanamegala@rvsgroup.com

Abstract

Every day, cyber bullying negative impacts on its victims get worse. Because of the harsh, emotionally abusive, and demeaning texts written by predators, several victims of cyber bullying have made suicide attempts. There have been several studies done on spotting bullying language in English comments. Bullying text categorization models are lacking in both bilingual (Tamil-English) texts, which makes the atmosphere hazardous. Due to the difficulty of classifying Bilingual texts, proposed models in Feature Extraction and Feature Selection phases which help in eliminate the Misclassification of bully texts. Proposed EW2V feature extraction model for handling the misspelled words and OOV and proposed a HPSO-GAFS hybrid model for constructing better feature sets. These models combined with the classification models to evaluate its performance. For evaluating the performance of the models Accuracy, F1-Score, Recall and Precision are used.

Keyword: Cyber Bully, Feature Extraction, Evolutionary Algorithm, PSO, GA, Feature Selection

1. INTRODUCTION

Internet usage is growing, and the accessibility of online communities opens up a channel for crimes like cyber bullying. A user spends the majority of their time updating their material, engaging with other users, and reading other users' accounts to discover specific data—all of which are crucial functions of social networking websites. People may now use these platforms to express themselves, their thoughts, and engage in discussion with others. It goes without saying that disagreements in viewpoint may lead to arguments in such a situation [1]. However, these discussions frequently take a negative turn and can lead to online confrontations where one side may use abusive language or poisonous remarks. These harmful remarks may contain threats, obscenities, insults, or xenophobic racial slurs. These clearly present the risk of online abuse and harassment. As a result, some people stop expressing their thoughts or stop looking for opposing viewpoints, which leads to unhealthy and unjust conversation. The heightened risk of an assault on internet users is a result of the internet's popularity. Many users make their private information available, which serves as a suggestion for the attacker to carry out particular malicious actions. The issue of cyber bullying has become more serious and has been seen as a social danger in various nations. As a result, various platforms and communities find it extremely challenging to promote fair discourse and are frequently compelled to either limit user comments or disintegrate by entirely shutting down user comments. An aggressive, intentional act against a helpless person utilizing the Internet or other electronic means, such as emails, website material, or text messages, is known as cyber bullying [2]. Detecting bullying on the internet when it occurs, reporting it to law enforcement authorities, Internet service providers, and others, as well as identifying predators and their victims, are issues in the fight against cyber bullying. Cyber bullying has received a great deal of attention from a social perspective, particularly with regard to comprehending its varied characteristics and its prevalence. In order to combat cyber bullying, automated detection of the behavior and the implementation of protective measures are required. Cyber bullying should be researched in terms of its identification, prevention, and mitigation. In order to filter and purify the Internet environment, research on harmful comment detection is essential on regional languages. The absence of ample training data, which is more accessible in a monolingual situation, particularly in English, which is still the focus of most work in perilous language analysis, makes multilingual bully text identification difficult [4]. To minimize harm to society, adhere to legal requirements, and foster an improved environment for its users, online platforms make a concerted effort to eliminate the harmful content. The issue still exists despite several techniques to automatically identify abusive phrases in internet platforms.

1.1. Motivation

The adverse effects of cyber bullying on its victims become worse day by day. Many victims of cyber bullying have attempted suicide as a result of the aggressive, emotionally abusive, and humiliating texts posted by predators. Many research conducted in identifying bully text only on English comments. There is a lack of bully text classification model in both Tamil-English comments make the environment unsafe.

1.2. Contribution

The paper contributes the following:

- To handle the misspelled words and Out of Vocabulary (OVV), proposed Enhanced Word2Vec (EW2V) feature extraction algorithm.
- To generate the best set of features based semantic features for the Tamil-English Comments Hybrid PSO-GA Feature Selection (HPSO-GAFS) approach proposed based on evolutionary algorithm.
- To classify the bully text from Tamil-English Comments through classification algorithm proposed earlier [1].

1.3. Organization of paper

The paper is organized as Section 2 covers the recent researches on the toxic or bully comments classification models, Section 3 elaborate the research work proposed, Section 4 compares the Results obtained by the models, and Section 5 concludes the paper.

2. RELATED WORK

In [5] discusses the models presented to the joint task on "Abusive Comment Detection in Tamil-ACL 2022" to solve the automatic identification of abusive phrases in online platforms. The collaborative endeavor focuses on the identification of offensive comments in Tamil texts written in both the original script and code. n-gram-Multilayer Perceptron (n-gram-MLP) model using MLP classifier fed with char-n-gram features and 1D Convolutional Long Short-Term Memory (1D Conv-LSTM) model were submitted as solutions to this challenge. The n-gram-MLP model performed better than the other two models, with weighted F1-scores of 0.560 for texts written in code-mixed Tamil and 0.430 for texts written in native Tamil script, respectively.

In [6] suggested dataset includes 9312 reviews that have been carefully classified into three categories: favorable, negative, and neutral by human experts. Create a manually annotated dataset for Urdu sentiment analysis as well as establish baseline findings utilizing rule-based, machine learning (SVM, NB, Adabbost, MLP, LR, and RF), and deep learning (CNN-1D, LSTM, Bi-LSTM, GRU, and Bi-GRU) approaches are the key objectives of the research work. A new multi-class Urdu dataset based on user evaluations was introduced for sentiment analysis. The information was compiled from a variety of industries, including those related to food and drink, plays and movies, applications and software, politics, and sports. They also improved Multilingual BERT (mBERT) for sentiment analysis in Urdu. To train our classifiers, employed four different text representations: word n-grams, char n-grams, pre-trained fastText, and BERT word embeddings. For assessment purposes, used two separate datasets to train these models. The suggested mBERT model with BERT pre-trained word embeddings outperformed rule-based classifiers, deep learning, and machine learning, according to the results, and earned an F1 score of 81.49%.

The research [7] intends to give a comparison analysis with the baseline models and maximise accuracy utilising the suggested model. Six machine learning models using two distinct methods of feature extraction were regarded as baseline models. Multi-class Sentiment Analysis (SA), a significant area of computer linguistics, uses NLP and text-mining algorithms to extract different views stated in a text. Existing research on ternary classification with fair classification performance is focused on Bengali's multi-class SA. A better performance score is also difficult to achieve because to the idiosyncrasies of Bengali language, the scarcity of ground truth datasets, and the limited power of preprocessing techniques. Furthermore, no study has demonstrated that deep learning algorithms outperform others on the four different categories of feelings. In order to conduct multi-class SA on Bengali social media comments classified as sexual, religious, political, and acceptable, we suggested a supervised deep learning classifier based on CNN and LSTM. On a labelled dataset of 42,036 Facebook comments, our suggested CLSTM architecture can significantly outperform SA with 85.8% accuracy and 0.86 F1 scores. To find the genuine sentiment of social media comments, a web application based on the suggested model and the best baseline model was created.

In [8], a study demonstrates that models utilizing transformers outperform methods based on machine learning. Models based on conventional machine learning methods, such as Bernoulli

ISSN 1533-9211

DOI 10.17605/OSF.IO/MKCHU

Naive Bayes, Support Vector Machine, Logistic Regression, and KNearest Neighbour, were developed to find the objectionable sentences. Additionally, attempts were made with multilingual transformer-based pre-trained models of natural language processing such mBERT, MuRIL (Base and Large), and XLM-RoBERTa (Base and Large). These models served as transformers for fine-tuning and adapting. In principle, adapters and fine-tuners achieve the same result; however adapters work by including additional layers and freezing the weights of the primary pre-trained model. Additionally, adapter-based strategies outperform fine-tuned models in terms of speed and effectiveness in low-resource languages like Tamil. XLM-RoBERTa (Large) was discovered to have the greatest accuracy of all adapter-based methods, with a score of 88.5%. The study also shows that adapter models require training of less parameter when compared to fine-tuning the models. The experiments also showed that the suggested models worked very well when applied to a cross-domain data set.

In [9], suggests a unique, straightforward, and efficient feature selection strategy to choose commonly distributed characteristics relevant to each class in order to address these issues. Due to the usage of formal language, misspellings, and shorter versions of words in brief texts, which provide high dimensionality and sparsity, sentiment analysis is a difficult process. In order to determine the pertinent feature for each class in this article, the feature selection approach is based on class-wise information. By contrasting the suggested feature selection approach with other feature selection techniques as chi-square (2), entropy, information gain, and mutual information, we assess its effectiveness. The performances are evaluated using the classification accuracy from the support vector machine, K nearest neighbours, and random forest classifiers on the Stanford Twitter dataset and the Ravikiran Janardhana dataset, two publicly accessible datasets. On the Stanford Twitter dataset, the suggested feature selection strategy performs better in terms of classification accuracy than the current feature selection methods. In majority of the feature subsets on the Ravikiran Janardhana dataset, the suggested technique performs satisfactorily on par with other feature selection methods in terms of classification accuracy.

3. RESEARCH METHODOLOGY

The model proposed here are shown in fig. 3.1

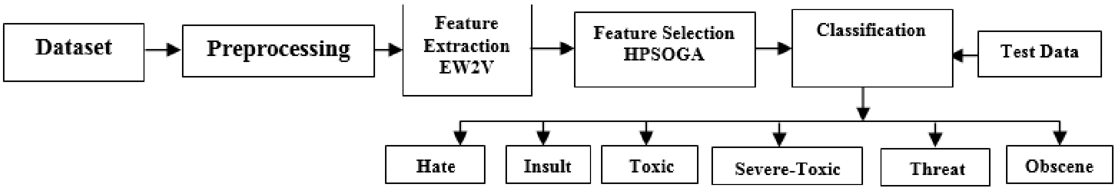


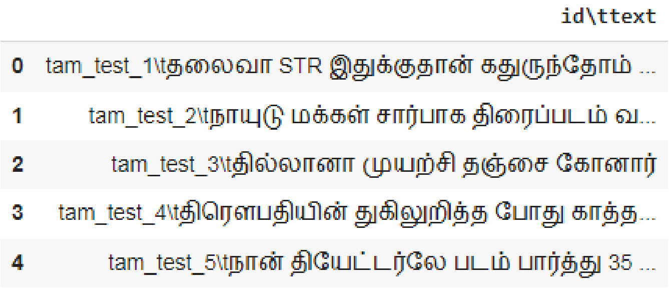
Figure: 3.1 Proposed Methodology

3.1. Dataset

Dataset used for the model are gathered from different sources of social media sites comments. The data was extracted from some of the most popular social media platforms, including Facebook, Twitter, and YouTube. Instead of utilizing any content linked to commercials or

DOI 10.17605/OSF.IO/MKCHU

sales; exclusively used text relating to fan comments as shown in fig. 3.2. Additionally, 151,532 items were assessed to be a big enough collection to train good word embeddings. 70% of data used for training and 30% used for testing the model.



	id\ttext
0	tam_test_1\tதலைவா STR இதுக்குதான் கதுருந்தோம் ...
1	tam_test_2\tநாயுடு மக்கள் சார்பாக திரைப்படம் வ...
2	tam_test_3\tதில்லானா முயற்சி தஞ்சை கோனார்
3	tam_test_4\tதிரௌபதியின் துகிலுறித்த போது காத்த...
4	tam_test_5\tநான் தியேட்டர்லே படம் பார்த்து 35 ...

Figure: 3.2 Sample Entries

3.2. Preprocessing

In Pre-Processing, the raw data is cleaned to make them ready for processing into the model. Data cleaning is the process of changing the all text into lower case, deleting hashtags, URLs, and user mentions as well as double spaces, emojis, punctuation marks, links, and other elements that are not necessary for the corpus.

After performing the above basic cleaning of data, next performed specific set of NLP models which is followed below:

- Language Identifier used to detect the words Tamil or English.
- Stemming process done to identify the root word, here an Rule based Iteration process performed to find the root word of Tamil texts. Separate stop words framed and used for Tamil.
- Lemmatization is the procedure of assembling various inflected versions of a single word. It takes context into account and changes the word to its Lemma, or meaningful basic form. This is to make the job easier in finding the nature of word.

3.3. Feature Extraction

Text feature extraction is the process of selecting from a document section to reflect information about the words' content. A text is seen as a dot in an N-dimensional space for the purposes of feature extraction, which is based on the vector space model. One aspect of the text in digital form is represented by one of the dot's dimensions. In most cases, feature extraction algorithms employ a keyword set. The feature extraction method determines the weights of the words in the text based on these predetermined keywords, and then creates a digital vector, which is the feature vector of the text. In generating the feature vectors proper word finding is highly essential even on the part of misspelled words and OOV. Initial thinking for handling misspell words was that the correct spelling should appear next to the misspell word. But that isn't the case; in vector space, all misspellings appeared to be clustered together. To handle the misspelling and OOV words proposed Enhanced Word2Vec (EW2V) feature extraction algorithm.

Algorithm: EW2V Feature Extraction Algorithm

Input: Misspelled Word

Output: Real Word

Step 1: Input the Misspelled word

Step 2: Run a cosine distance to get the 10 closest misspelled words to chosen word.

Step 3: Take the word vectors of each of the 10 closest words and deduct them from the word vector of the correctly spelled word as in equ. (1).

SUM_V=((GloVe[‘similary_word1’]- GloVe[‘similary_word2’])+ ………… s((GloVe[‘similary_word9’]- GloVe[‘similary_word10’])--- (1)

Step 4: In step 2, determine the average of each of these word vectors as in equ (2). In step 1, this turns into the translation vector for the chosen word as in equ. (3).

Trans_V=SUM_V/10 – (2)

REAL_W= GloVe[‘misspelled_word1’] +Trans_V – (3)

Step 5: Test satisfied, Display the real word.

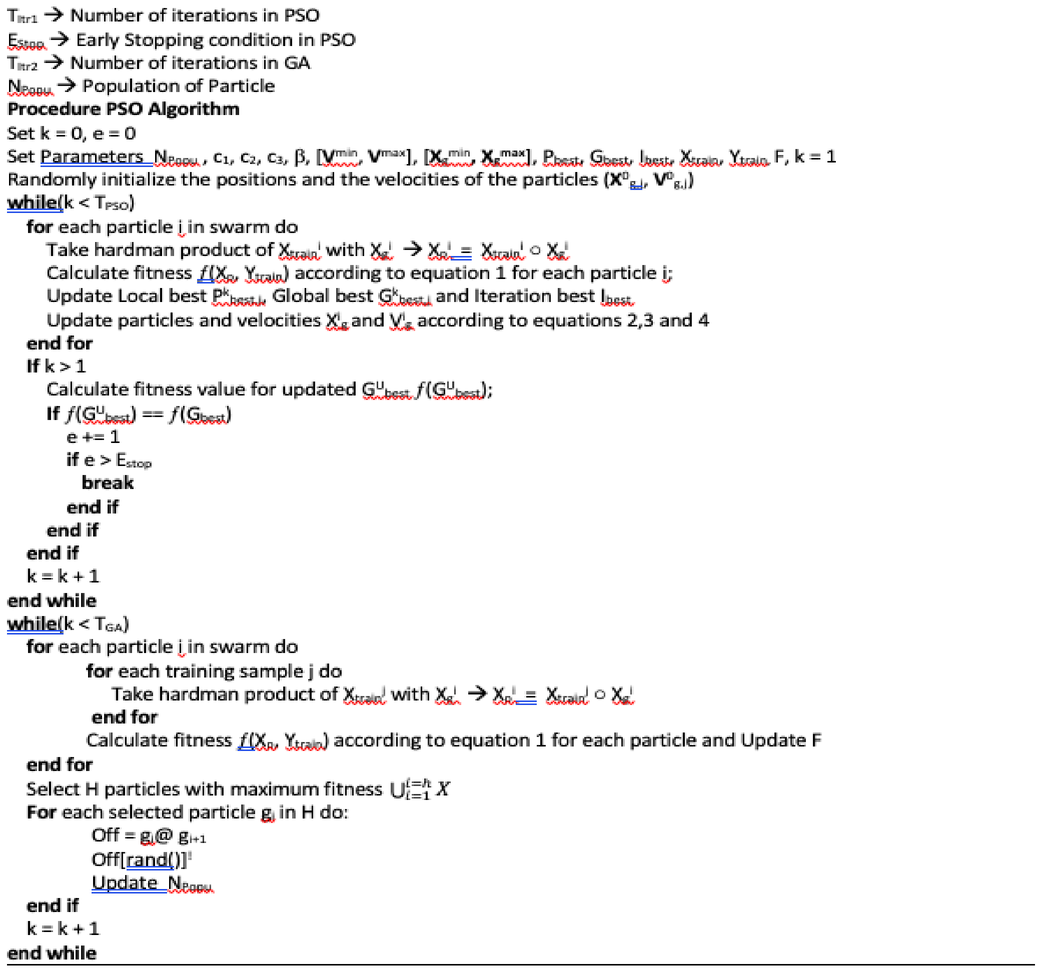
3.4. Feature Selection

The process of selecting a subset of the terms included in the training set and utilizing just this subset as features in text classification is known as feature selection. In this paper proposed a hybrid evolutionary algorithm for feature selection. Biological evolution theories serve as the foundation for evolutionary algorithms. The first step is to generate a "population" of potential solutions, and then each one is evaluated using a "fitness function" to determine how effective it is. Over time, the population changes and (hopefully) comes up with better solutions. Both PSO and G.A. were combined in this model. Since, each had a unique set of benefits. While experimenting with genetic algorithms, found that they operate best and achieve their global optimum when they are started with a healthy population. Thus, run PSO first to create a decent population, and then start the genetic algorithm with this population.

- Particle Swarm Optimization algorithm (PSO) [11] is a computer technique used in computational science to optimize a problem by repeatedly attempting to raise the quality of a candidate solution. By using a population of potential solutions, here referred to as particles and moving them across the search space in accordance with a straightforward mathematical formula over the particle's position and velocity, it solves problems. In addition to being led towards the best known positions in the search space, which are updated as other particles find better places, each particle's movement is also impacted by its local best known position.

- Genetic Algorithm [10] the mechanics of natural selection and natural genetics have inspired the development of genetic algorithms, which are randomized search algorithms. By adopting a randomized yet organized information exchange, genetic algorithms work on string structures that, like biological structures, are changing over time in accordance with the principle of the fittest. As a result, each generation generates a new collection of strings utilizing the best individuals from the previous generation.

Algorithm: HPSO-GAFS Algorithm



```
$T_{Itr1}$ → Number of iterations in PSO
$E_{Stop}$ → Early Stopping condition in PSO
$T_{Itr2}$ → Number of iterations in GA
$N_{Popu}$ → Population of Particle
Procedure PSO Algorithm
Set k = 0, e = 0
Set Parameters $N_{Popu}$ , $c_1$, $c_2$, $c_3$, β, [$V^{min}$, $V^{max}$], [$X_g^{min}$, $X_g^{max}$], $P_{best}$, $G_{best}$, $I_{best}$, $X_{train}$, $Y_{train}$, F, k = 1
Randomly initialize the positions and the velocities of the particles ($X^0_{g,i}$, $V^0_{g,i}$)
while(k < $T_{PSO}$)
  for each particle i in swarm do
    Take hardman product of $X_{train}^i$ with $X_g^i$ → $X_R^i$ = $X_{train}^i \circ X_g^i$
    Calculate fitness $f(X_R, Y_{train})$ according to equation 1 for each particle i;
    Update Local best $P^k_{best,i}$, Global best $G^k_{best,i}$ and Iteration best $I_{best}$
    Update particles and velocities $X^i_g$ and $V^i_g$ according to equations 2,3 and 4
  end for
  If k > 1
    Calculate fitness value for updated $G^U_{best}$ $f(G^U_{best})$;
    If $f(G^U_{best}) == f(G_{best})$
      e += 1
      if e > $E_{stop}$
        break
      end if
    end if
  end if
  k = k + 1
end while
while(k < $T_{GA}$)
  for each particle i in swarm do
        for each training sample j do
          Take hardman product of $X_{train}^j$ with $X_g^i$ → $X_R^i$ = $X_{train}^j \circ X_g^i$
        end for
        Calculate fitness $f(X_R, Y_{train})$ according to equation 1 for each particle and Update F
  end for
  Select H particles with maximum fitness $\bigcup_{i=1}^{i=h} X$
  For each selected particle $g_i$ in H do:
        Off = $g_i$@ $g_{i+1}$
        Off[rand()]$^1$
        Update $N_{Popu}$
  end if
  k = k + 1
end while
```

3.5. Classification

In [1], proposed an Enhanced Multi Label Classification is used for classify the bully text on both Tamil-English comments here. With base model Multinomial Navies Bayes and Decision Tress, previously proposed model Enhanced LinearSVC and Enhanced Logistic Regression models compared.

4. RESULT AND ANALYSIS

Compared the models proposed with base models and evaluated the model using the metrics such as Accuracy, F1-Score, Precision, and Recall.

Fig. 4.1 shows comparison of the proposed EW2V feature extraction with the base models of feature selection and proposed HPSO-GAFS model. From that EW2V feature extraction scores a good F1-Score and hence proved to be efficient. So it is further used in the analysis.

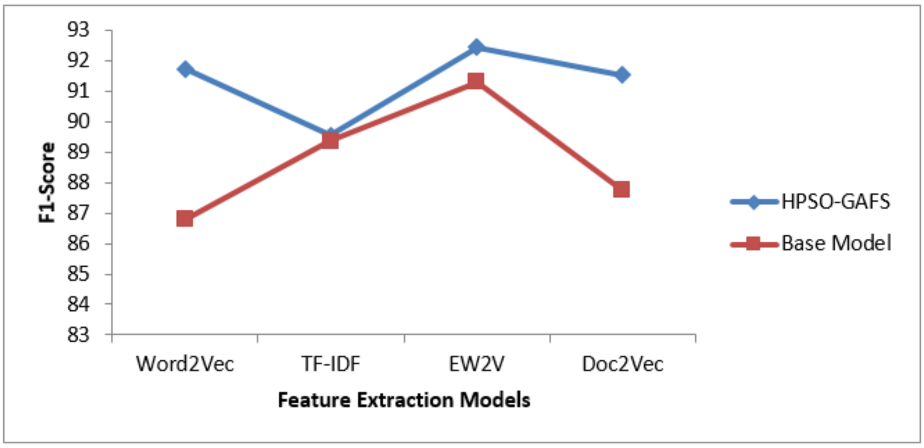
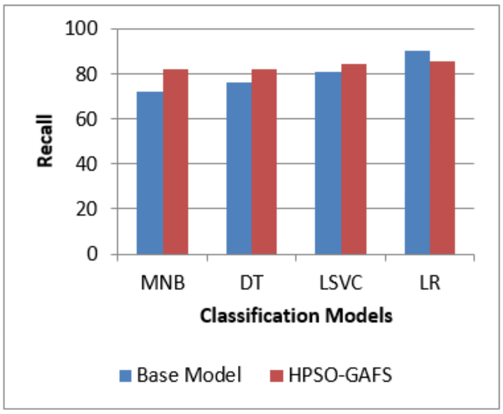


Figure: 4.1 F1-Score with Logistic Regression

From fig. 4.2 Proposed EW2VFS model shows a high precision rate in almost all the classification models. Comparatively on the LSVC and LR it increases the precision rate to 88.38 and 90.56 respectively.



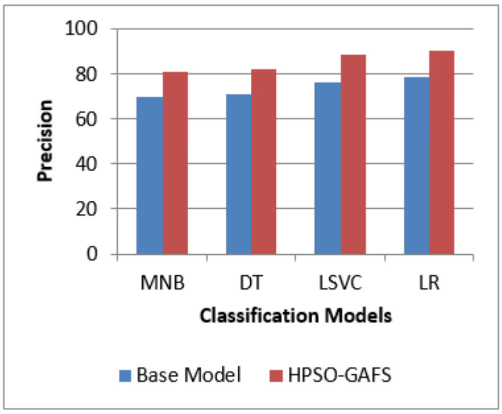


Figure: 4.2 Precision with EW2V **Figure: 4.3 Recall with EW2V**

From fig. 4.2 Proposed EW2VFS model shows a high recall rate in almost all the classification models. Comparatively on the LSVC and LR it increases the recall to 84.31 and 85.36 respectively.

Finally done the multi-label classification using base models and proposed models as shown in

DOI 10.17605/OSF.IO/MKCHU

fig 4.4, from that second combination EW2V, HPSO-GA and ELR yields good Accuracy rate compare to other two. On the label “Identity_hate” all the models performance shows a close rate of accuracy. On the label “Toxic”, “Sever_Toxic”, and “Obscence” second model performs well. On the label “Threat” and “Insult” first combination EW2V+HPSO-GAFS+ELSVC performs closer to second model. It is clear that EW2V+HPSO-GAFS+ELR is efficient in classifying the labels effectively.

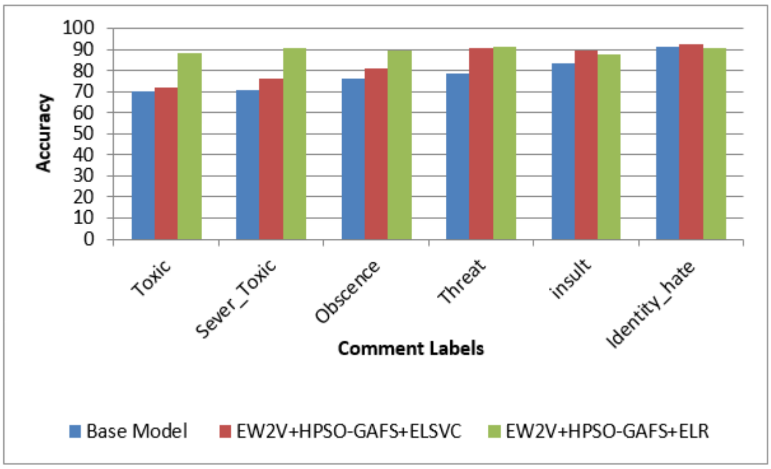


Figure: 4.4 Accuracy

5. CONCLUSION

Proposed two different combinations for classify the bully texts on both Tamil-English comments. The first model combines the EW2V feature extraction with EPSO-GAFS feature selection and ELSVC classification and the second model combines the EW2V feature extraction with EPSO-GAFS feature selection and ELR classification. These two models performance has been analyzed with base models and find that second combination gives accuracy of 88.38 % to 90.56% on almost all the labels. First combination shows a accuracy of 71.45% to 92.54% on the all labels. All models classify the “Identity_hate” comments at better rate of accuracy. From these performances it is clear that when feature extraction and feature selection phases improved can able to avoid the misclassification issue which is earlier happened.

References

1. Indumathi, V., and S. Santhana Megala. “Enhanced Multi-Label Classification Model for Bully Text Using Supervised Learning Techniques.” Proceedings of Data Analytics and Management, 2023, pp. 763–778., https://doi.org/10.1007/978-981-19-7615-5_62.

2. V. Indumathi, S.Santhana Megala, R.Padmapriya, “Classify Bully Text With Improved Classification Model Using Grid Search With Hyperparameter Tuning”, Advances and Applications in Mathematical Sciences, Vol. 21, Is. 9, pp. 4973-4980, 2022.

3. V. Indumathi, S.Santhana Megala, R.Padmapriya, M.Suganya, and B.Jayanthi, “Prediction and Analysis of Plant Growth Promoting Bacteria using Machine Learning for Millet Crops”, Annals of R.S.C.B., ISSN:1583-6258, 2021

4. Song, Guizhe, et al. “A Study of Multilingual Toxic Text Detection Approaches under Imbalanced Sample Distribution.” Information, vol. 12, no. 5, 2021, p. 205. https://doi.org/10.3390/info12050205.

5. F. Balouchzahi et. al., “MUCIC@TamilNLP-ACL2022: Abusive Comment Detection in Tamil Language using 1D Conv-LSTM”, Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages, pp. 64 – 69, 2022.

6. Khan, Lal, et al. “Multi-Class Sentiment Analysis of Urdu Text Using Multilingual Bert.” Scientific Reports, vol. 12, no. 1, 2022, https://doi.org/10.1038/s41598-022-09381-9.

7. Haque, Rezaul, et al. “Multi-Class Sentiment Classification on Bengali Social Media Comments Using Machine Learning.” International Journal of Cognitive Computing in Engineering, vol. 4, 2023, pp. 21–35., https://doi.org/10.1016/j.ijcce.2023.01.001.

8. Subramanian, Malliga, et al. “Offensive Language Detection in Tamil Youtube Comments by Adapters and Cross-Domain Knowledge Transfer.” Computer Speech & Language, vol. 76, 2022, p. 101404, https://doi.org/10.1016/j.csl.2022.101404.

9. Kumar, H. M., and B. S. Harish. “A New Feature Selection Method for Sentiment Analysis in Short Text.” Journal of Intelligent Systems, vol. 29, no. 1, 2018, pp. 1122–1134., https://doi.org/10.1515/jisys-2018-0171.

10. N. Bidi and Z. Elberrichi, "Feature selection for text classification using genetic algorithms," 2016 8th International Conference on Modelling, Identification and Control (ICMIC), Algiers, Algeria, 2016, pp. 806-810, doi: 10.1109/ICMIC.2016.7804223.

11. Aghdam, Mehdi Hosseinzadeh, and Setareh Heidari. “Feature Selection Using Particle Swarm Optimization in Text Categorization.” Journal of Artificial Intelligence and Soft Computing Research, vol. 5, no. 4, 2015, pp. 231–238., https://doi.org/10.1515/jaiscr-2015-0031.