Linguistic Model for Analyzing and Detecting Inappropriate Comments

Chennuru Giridhar, Syed Roshan, Thotamadugula Gowtham Reddy
Sir Padampat Singhania University, Udaipur, Rajasthan.

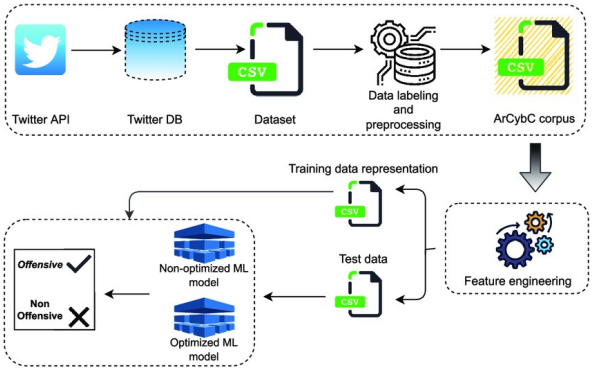
Abstract—The world has become interactive and socially active nowadays because of the increase in different types of content-sharing applications. These content-sharing applications are social media platforms that provide various features so that users can effectively interact and share their thoughts and ideology. One such platform is a X which promises the anonymous posting of users' views and complaints. As the growth In the craze of the X, they are being targeted by spammers for their work. Though these platforms act as a medium of knowledge sharing, all of the users don't use these platforms for a positive cause. They are also being used to abuse or bully targeted people taking advantage of their anonymous feature. Spamming and cyberbullying have grown rapidly to the limit that social media is being termed harmful. By reading spam and vulgar comments, readers will be in diverted and this results in several misconceptions that are harmful. The main aim is to detect these bad comments which are vulgar, inappropriate or not related to the specific context.

Keywords: Keywords— PPM Algorithm · TEM Algorithm · SAM Algorithm · Latent Dirichlet Allocation (LDA) · Natural Language Processing (NLP) · Machine Learning · Topic extraction.

Introduction

Social media platforms allow us to interact and share our ideas and opinions by means of sharing posts, tweets, comments, and so many other possible ways. By reading comments we get to know the ideology of the people and it is most useful in the case of online shopping where we try to buy a product by reading the comments and we can come to a view on that product. These comments are allowed to be posted anonymously to get more genuine views. Though we have access of reading the comments and coming to a decision but they may be spam and they cause an irrelevant and irresponsible impact on the reader's brain. As we already know the feature in one of the social media sites YouTube and the feature is it will delete a comment based on the number of dislikes until and unless reaches a particular number. By this action, we can understand the real motive is to not entertain any spam comments. In our research in this paper, our approach is to deal not only with spam comments but also to look after the bad, vulgar, and irrelevant comments that manipulate the reader's mind and are out of topic and are of no use. The first elimination is based on removing all the extra spaces and tabs in order to make them into tokens. The above-mentioned preprocessing is done only after checking the vulgarity of the comment, also based on the topic relevancy. After checking the above conditions the comments are deleted. Then we deal with topic extraction and topic similarity. We built a mechanism to identify the spam comments and apply the natural processing techniques in the later stages along with the machine learning algorithms. Additionally, social networks, online forums and e-commerce sites also challenge the moderation of the content generated by the users. When it comes to user-generated text in these media, however, the daily amount of comments about any topic is so impressive that conducting human moderation successfully is not feasible. There is a high demand for tools to ease the repetitive, burdensome, and time-consuming task of the human moderator in order to detect inappropriate, harmful, harmful, or illegal content. Natural Language Processing (NLP)techniques can be used for this purpose.

In this work, we explore the use of Machine Learning techniques applied to text classification to detect inappropriate content in text. We have assessed twelve models resulting from the combinations of text encoders, Term Frequency-Inverse Document Frequency (TF-IDF), Lemmitization, and Word2vec together with four classifiers (Support Vector Machines (SVMs), Logistic Regression (LR), k-Nearest Neighbors and Random Forests (RFs)).



I. Problem Statement

With this paper, we intend to create a system where forums, websites, and all social media sites will be spam free. Society nowadays is fully dependent on social media and it is responsible for changing and routing the behavior of the people. This attention paves a way for the spammers to promote irrelevant content and promote malicious behavior. The main idea of this paper is to provide a system where spam will detect and how the sentiment of the people is dependent based on the comments is calculated.

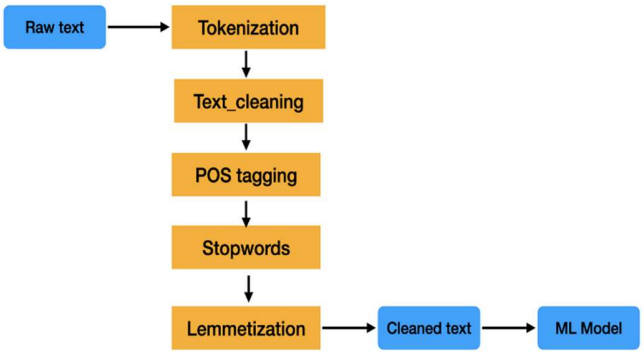
II. Literature Survey

In paper [1] Nobata, Chikashi, etal. Abusive language detection in online user content.Proceedings of the 25th international conference on world wide web. 2016.In addition to addressing the aforementioned gap in the area, algorithms presented in this research seek to create a cutting-edge way for identifying offensive language in user comments. The contributions made in this study are as follows:To outperform a deep learning system, this research uses supervised classification methodology with NLP features. utilises and modifies a number of the previous art features in an effort to compare their performance with the same data set. Add features from distributional semantics techniques to the feature list as well.Making a fresh data set of a few thousand user comments gathered from various domains public. This set comprises three judgments per remark and, for those considered abusive, a more detailed assessment of each comment's nature. In paper Davis, Dincy,

Reena Murali, and Remesh Babu. Inappropriate Comment Detection and Characterization of Twitter Behavior." arXiv preprint arXiv:2009.14261 (2020).The main goal is to concentrate on numerous abusive behaviours on Twitter and determine whether or not a communication is abusive. The suggested BiRNN is a better deep learning model for automatically detecting abusive speech, according to results of comparisons between Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) approaches for various abusive behaviours in social media.The suggested method for detecting abusive language was assessed using two measures, namely accuracy and F1-measure. In the future, the effectiveness of the proposed system will be assessed using a variety of domain datasets, including those from Facebook, Wikipedia, Twitter, and other online communities, in order to generalize user behavior. In the paper Vidgen, Bertie, et al. "Challenges and frontiers in abusive content detection." Association for Computational Linguistics,2019. This paper discusses the issues constraining, the performance, efficiency, and generalizability of abusive content detection systems. Abusive content detection is a pressing social challenge for which computational methods can have a hugely positive impact. In this paper, methods deal with critical insights into the challenges and frontiers facing the use of computational methods to detect abusive content. They differ from most previous research by taking an interdisciplinary approach, routed in both the computational and social sciences. In paper [4] Rajamanickam, Santhosh, et al. "Joint modeling of emotion and abusive language detection." arXiv preprint arXiv:2005.14028 (2020). Aiming to tackle this problem, the natural language processing (NLP) community has experimented with a range of techniques for abuse detection. While achieving substantial success, these methods have so far only focused on modeling the linguistic properties our work aims to investigate the relationship between emotion and abuse detection, which is likely to be independent of the biases that may exist in the annotations. They proposed a new approach to abuse detection, which takes advantage of the affective features to gain auxiliary knowledge through an MTL framework In paper [5] Kanan, T., Aldaaja, A., & Hawashin, B. (2020). Cyber-bullying and cyber-harassment detection using supervised machine learning techniques in Arabic social media content. Journal of Internet Technology,This paper mainly focused on detecting these phenomena in English text, few works studied this phenomenon in Arabic. To evaluate the performance of the classifiers, we use Recall, Precision, and F1 Measure. Future scope: it has provided a comprehensive comparison that would aid future research works in this direction. In paper Kaur, S., Singh, S., & Kaushal, S. (2021). Abusive Content Detection in Online User-Generated Data: A survey. Procedia Computer Science, 189, 274-281. This survey paper aims to help newcomers and budding researchers obtain an overall perspective of this research area by offering a thorough overview to gain insights into the area, including recent trends and proposed techniques. The researchers have successfully applied methods from the machine-learning field, with Bag-of-Words (BoW) and N-grams being the most frequently used features in classification.

III. METHODOLOGY

The main goal of the paper is to remove spam from all social networking sites. As we know social media is responsible for influencing people's minds and spamming results in a change of perspective and is harmful. So we proposed a system where the comments will be streamed from the forums and these comments will be assessed based on the vulgarity using a profanity module. The preprocessing and later stages will be continued based on the profanity check results and further topic detection and sentiment of the people is calculated.



NLP is a large and multidisciplinary field that can be defined as the automatic processing of human languages. Any practical application that makes use of text is a candidate for NLP and its success is largely driven by the advances in the machine learning field. These text-processing tasks include classification, translation, structured prediction, and sequential decision among others. This work focuses on text classification, i.e., the process of assigning categories to text according to its content. It applies to a wide variety of tasks like spam detection, sentiment analysis, detection of hate speech, cyberbullying, or the detection of inappropriate content that we tackle in this paper. The detection of inappropriate erotic/sexual content is a very challenging NLP problem. This process takes a considerable time to perform manually considering the vast amount of information published on social media. The development of fast and efficient tools for the automated discovery of this content becomes crucial to protect minors on the Internet. The goal of this work is to explore how the detection of sexual erotic content can benefit from the use of NLP and Machine Learning techniques. In this section, we present the methodology used to build the proposed sexual/erotic text classifier. The typical

scheme of a supervised text classifier has two main components: text representation to encode the input text samples into feature vectors and a classifier that categorizes these feature vectors into one of the two categories. Improvements and advances in computer hardware have made it possible to use deep learning-based classifiers with remarkable performance. Nonetheless, it has been seen that for some text classification problems classical machine learning techniques also give similar performance results with less training time and hardware requirements. In this paper, we assess this latter approach. We address the task of detecting sexual erotic text assessing different frameworks for text categorization. The classification models evaluated in this work result from the combination of different text encoders together with some classical machine learning techniques that have shown good performance on text classification tasks. For any of the proposed schemes, we consider a preprocessing step that includes tokenization, lemmatization, and removal of stop words, among others. Documents can be considered as a set of words dependent on one another, and they need to be represented as numerical vectors that, hopefully, reflect the relationships among words. Next, we present different text encoding techniques for this task.

IV. Encoding Techniques

A. Introduction

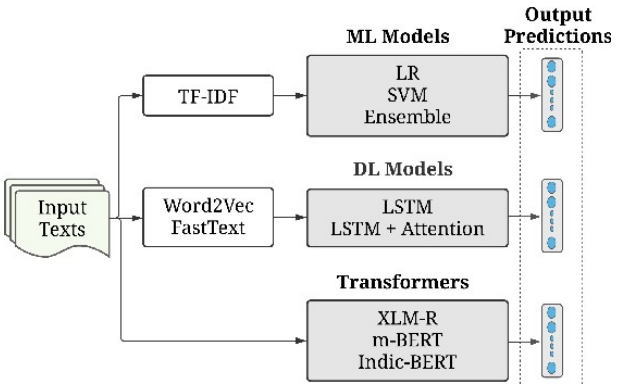
There are two main approaches to tackling the text-encoding task: The well-known Vector Space Model (VSM) approach and(b) the recent proposals based on word embedding. The VSM approach converts a text document into a numerical vector whereas the word embedding approaches turn individual words into numerical vectors with arbitrary dimensionality. Within the VSM framework, each vector component is a measure of the importance of the corresponding term in the represented document. Several techniques have been proposed to compute the weights (i.e., the vector components) for the different words in a given document. Term frequency (TF) (also known as BoWs) and TF-IDF are some of the most well-known weighting techniques. In this case, each instance xi is represented as a p-dimensional vector xi = [xi1,xi2,…xip] where each component xij represents the importance of a given word fj for sample i. Besides, over the last few years, there has been a high interest in word embedding. Word2Vec, Glove, or FastText are just some examples of popular word embedding-based approaches. The word vectors are obtained by training the algorithm with a text corpus. After such training has finished, each word in the corpus is represented by a numerical q-dimensional vector. The value of q is chosen at the time of training and it is smaller than the number of unique words in the corpus, i.e., q << p.

B. Term Frequency-Inverse Document Frequency

TF-IDF is one of the most commonly used term weighting schemes in nowadays information retrieval systems [46,47]. The term TF refers to the frequency of appearance of a word in the text, whereas IDF is a term weighting function that measures how important each word is to the text document. IDF was first proposed by Karen Spärck Jones.

$$w_{i,D} = tf_{i,D} \log \frac{N}{df_i},$$

Let the vector of weights for document D be wD = (w1, D, w2, D,…, wp, D). The weight of the ith term in document D is defined as where tfi, D is the frequency of appearance of term I in D, dfi is the number of documents in which that term appears and N is the total number of documents in the document set. The intuition behind this weighting technique is that (a) the more frequently a word appears in a document, the more representative it is for that document but (b) the more documents the word appears in, the less informative it is.



C. Word2vec

Word2vec is a method, proposed in 2013 by T. Mikolov and Col Leagues, to efficiently create word embeddings that has gained a lot of attention in the past few years.

The Word2vec model creates a vocabulary from the training text data and learns dense word embeddings, i.e., the representation of a word, i.e., useful for the prediction of other words in the sentence. This vector representation can be subsequently used in machine learning applications like text categorization. The word vectors generated with Word2vec seem to capture many linguistic regularities as long as the models are trained with large enough datasets. Next, we briefly describe this method and refer the interested reader to details about the implementation. Word2vec generates word embeddings using a shallow neural network, i.e., a fully connected neural network with a single hidden layer. The input layer has as many neurons as words in the training vocabulary. The size of the hidden layer is the dimensionality of the feature space and the weights adjusted during the learning

stage are then used as the embeddings. The size of the output layer is the same as that of the input layer. There are two main learning algorithms in word2vec: Continuous Bag-Of-Words (CBOW) and Skip-Gram (SG). CBOW tries to predict the target word(i.e., the center word)given some context words (i.e., surrounding words), whereas SG predicts context words based on the target word. Some studies suggest that SG is slower but better for infrequent words while CBOW is faster. An illustration of the architecture of both techniques can be seen in F. For instance, considering a simple sentence, "the greedy dog story writing," there can be pairs of (context, target word) where if we consider a context window of size1, we have examples like([greedy, story], dog), ([the dog], greedy), ([dog, writing], story) and so on. Thus, with CBOW, the model is trained to predict the target word based on the surrounding context words and the other way around in the case of SG. The training algorithms could be either hierarchical softmax or negative sampling. The former tends to be better for infrequent words while the latter is better for frequent words and also better with low dimensionality of the feature space. This leads to a dense word embedding with lower dimensionality than the traditional sparse vector models

D. *Lemmatization*

The lemma of the word is found. So we can see that the extra endings are removed in lemmatization. The word which is returned is called the lemma. These two terms are not the same, Stemming is just finding the root word but most times it's not preferable lemming is a technique where morphological analysis of the words. It returns a lemma.

Flow of Pre-processing module

1. Algorithm: PPM
2. Input: Comments entered in the X
3. Output: Filtered Tokens
4. Tokenize the comment using any tokenizer present in the nltk module
5. Summarize the tokens
6. Lemmatize the tokens using Wordnet Lemmatizer

E. *POS Tagging*

In this process, for each token that has been formed after the preprocessing, we assign the part of speech to which it belongs. The words are classified based on their part of speech, tense, number, case, etc.

The set of all the POS tags used in the POS tagging are called tag sets where they differ from language to language. Basic tagsets are like N for nouns, A for adjectives, etc. For that, we will have a list of tagsets but each one of them is not useful for topic extraction because as we know mostly to obtain the topic being discussed in a given sentence or paragraph or document we rely on the nouns being discussed. Not only the nouns but we are also deciding it based on the adjectives and verbs being discussed as they describe the nouns and the situation which is being talked about in a sentence

F. *Sentiment Analysis Module*

Sentiment calculation shows the sentiment of the people based on the topic being discussed how this results in future comments and how the people's opinions are based. This is a classification where the inserted phrase is decided In our research we used sentiwordnet. Senti wordnet is a document containing all the synsets of wordnet along with their "positivity", "negativity", and "neutrality". Each synset has three scores Positive score, Negative score, and Objective score which represent how positive, negative, and "objective" (i.e., neutral) the synset is. These scores may range from 0.0 and go up to 1.0, the sum of all three scores being 1.0. Each score for a synset term has a non-zero value. SentiWordNetsynset term scores have been computed semi-automatically based on a semi-supervised algorithm. So the result obtained which shows the sentiment of the phrase describes how the opinion of the people is and also the opinion of the topic being discussed which helps a lot in the case of our forum where students will be discussing all their issues which paves a way for the management and the teachers to look after the issues which needed to be taken care and how they need to be handled are also discussed as we provided their suggestions section also so they can reach the staff and be resolved. This system helps not only the faculty and institution but also the students who want their issues to be solved.

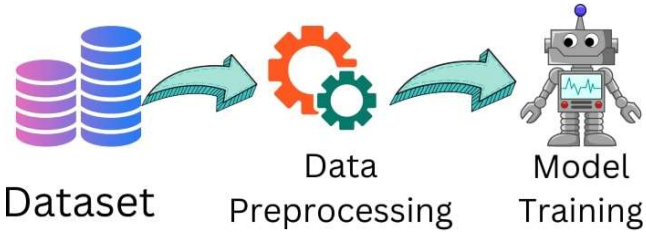
G. *Nave Bayes*

Nave Bayes is a family of probabilistic algorithms that take advantage of probability theory and Bayes Theorem [6]. It is also used in NLP applying the Bayes Theorem to predict the probability for each class such as the probability that a given data point belongs to a particular class [45]. In this study, the multinomial Nave Bayes is applied using the MultinomialNB function from the package sklearn of Python to train the model.

V. *Text Preprocessing*

A. Preprocessing

Before classification, the documents must be preprocessed i.e., the original text is transformed into a set of tokens. Afterward, a set of features must be extracted from such preprocessed text, which are thereafter passed as in putt another classification model of the system. Rigorous testing, validation, and redundancy mechanisms are employed to ensure the system's performance meets stringent safety standards and operational requirements, minimizing the risk of accidents or malfunctions.

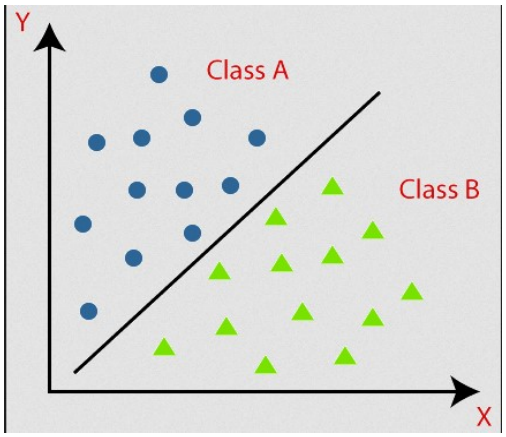


B. *Feature Extraction*

The feature extractor is responsible for converting the tokens into numerical values that can be used by a learning algorithm. In this work, we chose three feature extraction methods to encode the pre-processed text (i.e., the tokens) into the data that can be processed by the classification model: BOW, TF-IDF, and Word2Vec. All of them were executed using n-grams, which means that the words (i.e., the tokens) were analyzed in groups of n. In this work, we used unigram (i.e., n = 1) and bigram (i.e., n = 2). BOW and TF-IDF as well as the mentioned preprocessing methods were implemented through the Scikit-learn5 library. The minimum document frequency has been set to 3 (i.e., the documents with a document frequency of less than 3 have been ignored). Word2Vec was implemented using the Gensim6 library. More specifically, we evaluated the CBOW model architecture with the following parameters: the dimensionality of the word embeddings wassetto500, the context window size(i.e., how many words before and after a given target word would considered as context) was set to 5, the number of epochs was set to 30 and the minimum document frequency was set to 3. For feature learning we used the training algorithm negative sampling, setting to 5 the number of negative samples. Finally, the threshold for randomly downsampling higher-frequency words was set to 0.001.

C.Classification Model

We investigate a binary classification model employing logistic regression, known for its straightforward implementation and efficacy in binary decision tasks. Our model was trained and tested on a dataset containing XYZ records, with features A, B, and C subjected to preprocessing steps like normalization and handling missing values. The dataset was partitioned into 80% for training and 20% for testing, and hyperparameter tuning was conducted using cross-validation during training. *For model evaluation, standard performance metrics including accuracy,*



precision, and recall were employed. The model achieved an accuracy of 85%, precision of 0.82, and recall of 0.88 on the test set, demonstrating its proficiency in classifying instances into two distinct classes based on input features.

In our discussion, we recognize the model's strengths in achieving high accuracy and its resilience to dataset variations. However, we also note potential limitations such as susceptibility to overfitting on certain features. Future investigations could explore more sophisticated algorithms or incorporate additional features to enhance the model's performance.

In summary, this research contributes insights into logistic regression for binary classification tasks, showcasing its practical application and suggesting avenues for further refinement.

VI. *Future scope*

Our research has overcome the problem with some comments and all the disadvantages were in the existing system proposed the system where spam comments will be detected based on finding out its features and also the problem where topic irrelevant comments which lead to misconception are also dealt with. Future enhancements can be made to this project as we are streaming the comments not just taking the static content which provides a great scope not only to remove the spam comments but to make this evaluation of the topic to be applicable in other areas of interest. The vast amount of user-generated content on the Internet over the last few years makes the manual moderation of these texts, images, and videos an unachievable task. There is a clear need for techniques to automate the detection of inappropriate content generated by users. In this work, we address the problem of detection of inappropriate comments or text posts on social media using machine learning techniques. The aim of this study is the assessment of several text encoders (either based on VSM or word embeddings) together with different classification models to detect erotic-sexual content in texts. The experimental results were conducted with a dataset extracted from public data on the Reddit Website. The best performance result is achieved with an SVM classifier with a linear kernel using the TF-IDF technique as a text encoder. The classification error is only 3% and precision and recall reach the values of 0.96 and 0.95, respectively. It is noteworthy that the simple TF-IDF feature weighting approach outperforms more complex ones based on word embeddings. The experimental results achieved in this work suggest that applying machine learning techniques to this problem is a reliable approach that enables automatic moderation of this form of inappropriate content. These models can be used to develop real filters for social networks. Minors are potential users as well as other users for whom these comments are not relevant, such as YouTube where the user can find a large number of comments without regulation.

VII. *Conclusion*

As the amount of online user-generated content quickly grows, it is necessary to use accurate, automated methods to flag abusive language is of paramount importance. Not addressing the problem can lead to users abandoning an online community due to harassment or companies pulling advertisements that are featured next to abusive comments. While there has been much work in this area in several different related fields, to date, there has not been a standard evaluation set with which researchers could compare their methods. Additionally, there have been several NLP methods used in prior work but these features have never been combined or evaluated against each other. In our work, we take a major step forward in the field by first providing a curated public dataset and also performing several evaluations of a range of NLP features. We experimented with several new features for this task: different syntactic features as well as different types of embedding features and found them to be very powerful when combined with the standard NLP features. Character n-grams alone fare very well in these noisy data sets. Our model also outperforms a deep learning-based model while avoiding the problem of having to retrain embeddings on every iteration. Next, we used our model to perform an analysis of hate speech over the course of one year, providing practical insight into how much data and what kind of data is necessary for this task. Most work has so far focused on abuse found in English, but it remains to be seen how our approach or any of the other prior approaches would fare in other languages. Given how powerful the two n-gram features were in English, these would probably fare well in other languages given enough training data. Another area of future work includes using the context of the comment as an additional feature. The context could include the article it references, any comments preceding or replied to, as well as information about the commenter's past behavior or comments.

VIII. *REFERENCES*

[1] Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., & Chang, Y. (2016, April). Abusive language detection in online user content. In Proceedings of the 25th International Conference on World Wide Web (pp. 145-153).

[2] Davis, D., Murali, R., & Babu, R. (2020). Abusive Language Detection and Characterization of Twitter Behavior. arXiv preprint arXiv:2009.14261.

[3] Vidgen, B., Harris, A., Nguyen, D., Tromble, R., Hale, S., & Margetts, H. (2019, August). Challenges and frontiers in abusive content detection. Association for Computational Linguistics.

[4] Rajamanickam, S., Mishra, P., Yannakoudakis, H., & Shutova, E. (2020). Joint modeling of emotion and abusive language detection. arXiv preprint arXiv:2005.14028

[5] Kanan, T., Aldaaja, A., & Hawashin, B. (2020). Cyber-bullying and cyber-harassment detection using supervised machine learning techniques in Arabic social media content. Journal of Internet Technology, 21(5), 1409-1421.

[6] Kaur, S., Singh, S., & Kaushal, S. (2021). Abusive Content Detection in Online User-Generated Data: A survey. Procedia Computer Science, 189, 274-281.

[7] Vidgen, B., Harris, A., Nguyen, D., Tromble, R., Hale, S., & Margetts, H. (2019, August). Challenges and frontiers in abusive content detection. Association for Computational Linguistics

[8] Urrutia Zubikarai, A. (2020). Applied NLP and ML for the detection of inappropriate text in a communications platform (Master's thesis, Universitat Politècnica de Catalunya).

[9] Althobaiti, M. J. (2022). BERT-based Approach to Arabic Hate Speech and Offensive Language Detection in Twitter: Exploiting Emojis and Sentiment Analysis. International Journal of Advanced Computer Science and Applications, 13(5).

[10] Balayn, A., Yang, J., Szlavik, Z., & Bozzon, A. (2021). Automatic Identification of Harmful, Aggressive, Abusive, and Offensive Language on the Web: A Survey of Technical Biases Informed by Psychology Literature. ACM Transactions on Social Computing (TSC), 4(3), 1-56.

[11] Ballinger, N. (2022). Using a BERT-based Ensemble Network for Abusive Language Detection.

[12] Biere, S., Bhulai, S., & Analytics, M. B. (2018). Hate speech detection using natural language processing techniques. Master Business Analytics Department of Mathematics Faculty of Science.

[13] Kshitiz, K., Singh, H., & Kukreja, P. (2017). Detecting hate speech and insults on social commentary using NLP and machine learning. Int J Eng Technol Sci Res, 4(12), 279-285.

[14] Srikantrh, P., & Behera, C. K. (2022, July). A Machine Learning Framework for Covid Detection Using Cough Sounds. In 2022 International Conference on Engineering & MIS (ICEMIS) (pp. 1-5). IEEE.

[15] Srikanth, P., & Behera, C. K. (2022, July). An Empirical Study and assessment of minority oversampling with Dynamic Ensemble Selection on COVID-19 utilizing Blood Sample. In 2022

International Conference on Engineering & MIS (ICEMIS) (pp. 1-7). IEEE.

[16] Srikanth, P. (2021). An efficient approach for clustering and classification for fraud detection using bankruptcy data in an IoT environment. International Journal of Information Technology, 13(6), 2497-2503.

[17] Panigrahi, S. (2020, April). Design and Analysis of Efficient Cluster Using Novel Dissimilarity Measure and Classification for High Dimensional Cancer Datasets. In Proceedings of the International Conference on Innovative Computing & Communications (ICICC).

[18] Panigrahi Srikanth, Kolla Saitejaswi and Dharmaiah Devarapalli, 'TEJU: Fraud Detection and Improving Classification Performance for Bankruptcy Datasets Using Machine Learning Techniques', international conference on sustainable computing in science, technology and management, ELSEVIER –SSRN,2019.

[19] Panigrahi Srikanth, Dharmaiah deverapalli and Narsinga Rao M . R "Identification of AIDS Disease Severity Based on Computational Intelligence Techniques Using Clonal Selection Algorithm", International Journal of Convergence Computing – INDERSCIENCE Publications, Volume 2, Issue 3-4, page No:193-207.

[20] Panigrahi Srikanth "Clustering Algorithm of Novel Distribution Function for Dimensionality Reduction Using Big Data of OMICS", 2016 IEEE International Conferences on Computational Intelligence and Computing Research (ICCIC 2016), 2016. IEEE-2016, PP-1-6.

[21] Panigrahi Srikanth and Dr.N. Rajasekhar, "A Novel Cluster Evolution for Gene-miRNA Interactions Documents using Improved Similarity Measure", International Conferences on Engineering &MIC-2016 (ICEMIS -2016), IEEE-Morocco Section, IEEE-2016, PP-1-7.

[22] Panigrahi Srikanth and Dharmaiah Devrapalli, "A Critical Study of Classification Algorithms Using Diabetes Diagnosis", 2016 IEEE 6th International Conferences on Advanced Computing (IACC 2016), IEEE 2016, (Google Scholar), 27-28 Feb 2016, PP-245-249.

[23] Panigrahi Srikanth and Dharmaiah deverapalli," CFTDISM: Clustering Financial Text Documents Using Improved Similarity Measure", 2017 IEEE International Conferences on Computational Intelligence and Computing Research (ICCIC 2017), 2017.