Detecting Cyberbullying in Text Using TF-IDF Vectorization and Support Vector Machine Classification

V karthick1, Vaishnavi C2, Vaishnavi Sri SM3

*123Computer Science and Engineering Department, Rajalakshmi Engineering College Chennai*

[vkarthick86@gmail.com](mailto:vkarthick86@gmail.com)

[210701299@rajalakshmi.edu.in](mailto:210701299@rajalakshmi.edu.in)

[210701298@rajalakshmi.edu.in](mailto:210701298@rajalakshmi.edu.in)

**ABSTRACT *-*** Cyberbullying is a prevalent form of harassment in India. It encompasses various behaviors that degrade human body features and sentiments in every conceivable way. Victims of cyberbullying often experience psychological distress such as depression and agitation. In response to numerous cases of suicide and public outcry against cyberbullying, the Indian government has implemented punitive measures. Individuals found engaging in cyberbullying can now face legal consequences. To address this issue, we have developed a website capable of detecting cyberbullying based on factors such as generation, cultural background, gender identity, and faith. The ML model we have used here is SVM, which classifies texts and makes predictions on test data with the help of TF-IDF. TF-IDF converts text into numerical values to facilitate fitting the data into the training model. Additionally, in the preprocessing of the data, NLTK is used to convert the text into base forms to classify the data into the aforementioned categories.

***Keywords - Cyberbullying detection , SVM , NLTK ,***

***TF-IDF,stemming , tokenization , lemmatization.***

# **1.Introduction**

Cyberbullying detection holds extreme importance in our environment as it serves as a protector against social media harassment. Cyberbullying is timeless; it can happen over any period of time and easily hurt the feelings of the victim. For example, bullying someone physically over their appearance can leave an everlasting impact on their lives. Our main aim is to detect cyberbullying as soon as possible; if this is done, then the internet would be the safest place, even for children. The purpose of machine learning here is very simple: teaching machines to learn from examples. This proves to be very helpful in implementing cyberbullying detection. Basically, our machine learning model is designed in such a way that if the user is not able to figure out whether the text is offensive or not, we not only tell whether the text is cyberbullying but also what type of cyberbullying, whether it is related to age, cultural background, gender identity, and faith of that person. Hence, our model contributes to the betterment of society, which is a small contribution from our side.

The implementing method we have used here includes Support Vector Machine (SVM), a popular machine learning algorithm employed to classify texts, making it easy to detect whether a given text contains cyberbullying content. SVM accomplishes this by leveraging various features such as word frequency and language characteristics. These features are then inputted into the model, which generates a hyperplane, essentially a graph that depicts two extreme sides of the model. When the dataset is fed into the system, SVM separates the dataset and learns from it. Subsequently, it displays information where one part of the plane consists of cyberbullying instances, another comprises normal, positive comments, and a third section may contain instances that share characteristics of both categories

TF-IDF plays a pivotal role in cyberbullying detection because of its ability to convert text into numerical data, which is a crucial step in our project. It is also used to reduce dimensionality, highlight different features, etc. It helps to identify keywords related to cyberbullying. TF-IDF has a feature of normalization, which allocates weights to terms. Here, higher weights are assigned to rarely used items, and lower weights to commonly used ones. TF-IDF also offers computational efficiency and scalability.

The Natural Language Toolkit is used here for tokenization, lemmatization, and stemming. While NLTK is typically utilized in recommendation algorithms, here it primarily serves for lemmatization, which involves converting words to their base or dictionary form. This enhances the quality and consistency of the data, assisting in the modeling process.

# **APPROACH**

* 1. **Support Vector Machine**

Support Vector Machine is termed as a very efficient supervised machine learning algorithm present over the realm of ML. This algorithm has its specialization in classification tasks and sometimes in regression too. Its main task is to identify the best suitable and optimal hyperplane that fits best with the dataset and aids in training the model. Here, SVM is used for both linearly separable and non-linearly separable data but is mostly used for linearly separable data. It displays the graph not only in 2 dimensions but also in 3 dimensions.

Here, several mathematical formulas are used to implement Support Vector Machine. For the decision function, it is represented as [[2]](https://ar5iv.labs.arxiv.org/html/1912.05864) 𝑓(𝑥)=𝑤𝑇𝑥+𝑏*f*(*x*)=*wTx*+*b*, and for the margin, it is 2∥𝑤∥∥*w*∥2​, where ∥𝑤∥∥*w*∥ denotes the Euclidean norm.

Support Vector Machine (SVM) is not only proficient in binary classification but also in multi-class classification tasks.[[1](https://journals.lww.com/coronary-artery/abstract/9900/a_novel_radiomics_based_technique_for_identifying.232.aspx)] It achieves this by leveraging techniques like one-vs-one or one-vs-rest. The key aspect of SVM is its regularization parameter, denoted as 𝐶’, which acts as a trade-off between maximizing the margin and minimizing classification errors. A large value of 𝐶*’* emphasizes overfitting, while an extremely smaller value indicates underfitting. Therefore, finding the optimal 𝐶*’* value is crucial to balance these aspects and ensure the model performs well on any data, accurately predicting the output. The most important component of [[3]](https://arxiv.org/abs/2104.11146) SVM is the support vectors, which are critical points determining the orientation and position of the hyperplane. These support vectors play a crucial role in efficiently utilizing memory.

# **TERM FREQUENCY – INVERSE DOCUMENT FREQUENCY**

The Term Frequency-Inverse Document Frequency (TF-IDF) [[4]](https://arxiv.org/abs/1902.05731) is a numerical statistic used prominently in natural language processing (NLP) and predominantly in retrieving important information. Its purpose is to propose the importance of certain words in the document and how they contribute to the overall importance and role in training the model. There are two main important terms; the first one is term frequency. In simpler words, the term frequency is nothing but the count of times a particular word has occurred in the document. The formula for term frequency is as follows:

[[6]](https://ieeexplore.ieee.org/document/10423496)TF(𝑡,𝑑)=Number of times term 𝑡 appears in document 𝑑Total number of terms in document 𝑑TF(*t*,*d*)=Total number of terms in document *d*Number of times term *t* appears in document *d*​

Where: 𝑡 represents a term.

𝑑represents a document

The value of TF ranges between 0 and 1, where a larger value indicates that a particular term occurs more frequently in that document. However, it does not consider how unique the term is in that document

[[7]](https://ieeexplore.ieee.org/document/9712125)The Inverse Document Frequency (IDF) is a critical term in natural language processing, essential for evaluating the significance of terms across a collection of documents. It emphasizes the importance of rare words by assigning them higher weights. The formula for IDF is IDF(t, D) = log(N/df(t)),

Where,

* N represents the total number of documents in the corpus.
* df(t) is the document frequency of the term. The IDF value increases logarithmically as it continues to recognize the words in those

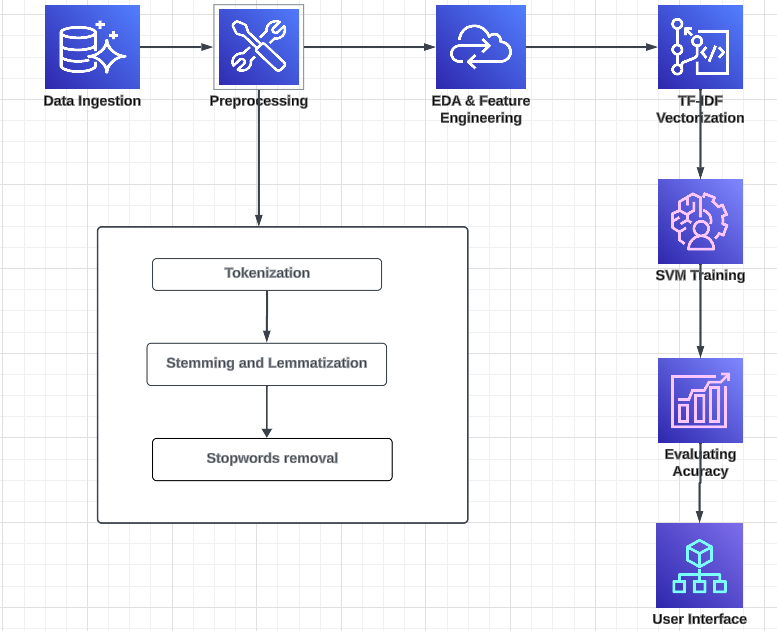
documents. Terms occurring less frequently across documents will have lower IDF values, indicating their importance in distinguishing documents.

# **Natural Language ToolKit**

# 

Natural Language Toolkit is an excellent resource that offers a wide range of functionalities for manipulating

data. Among the important functions of NLTK is tokenization, which involves breaking down text into tokens. Additionally, [[8]](https://ieeexplore.ieee.org/document/10118992)NLTK includes capabilities such as part-of-speech tagging, which assigns grammatical categories to words in a sentence. These functionalities empower users to perform various text processing and analysis tasks efficiently.



**Fig.1. Architecture Diagram**

In cyberbullying detection, the Natural Language Toolkit (NLTK) serves a pivotal role in text preprocessing, which involves preparing data for subsequent text analysis and machine learning tasks. The NLTK **RegexpTokenizer** is employed for tokenization, which entails breaking down text into

In cyberbullying detection, the Natural Language Toolkit (NLTK) [[9]](https://ieeexplore.ieee.org/document/9537850) serves a pivotal role in text preprocessing, which involves preparing data for subsequent text analysis and machine learning tasks. The NLTK **RegexpTokenizer** is employed for tokenization, which entails breaking down text into tokens or individual words—a foundational step in text analysis. Additionally, NLTK provides tools like the **PorterStemmer** and **WordNetLemmatizer** for stemming and lemmatization, respectively. Stemming involves reducing words to their base forms, while lemmatization considers contextual information to normalize words further. NLTK also contributes significantly to stopwords removal, where common, non-contextual words such as "is," "was," and "then" are eliminated to reduce noise and emphasize content words. Overall, NLTK's robust suite of functionalities ensures effective text preprocessing, facilitating the subsequent analysis and detection of cyberbullying behavior in online communications. requiring attention due to potential issues.

**3.Proposed Work**

Firstly,this project wants a dataset regarding cyberbullying but that has already been attained with the "cyberbullying\_tweets.csv" file. This dataset has tweets labeled with different kinds of cyberbullying. Preprocessing the data is very important because it cleans and prepares it for analysis. This contains many processes involving changing text to lowercase, removing stopwords, punctuation, repeating characters, URLs, and numeric data. Additionally, tokenization, stemming, and lemmatization are applied to normalize the text data. These preprocessing steps make sure that the textual tweet data is in a correct format for feature extraction and modeling.

Once all the dataset has been preprocessed, the second step is to convert the text into a numerical format that machine learning models can recognize. We use the TF-IDF (Term Frequency-Inverse Document Frequency) Vectorizer for this purpose. The TF-IDF Vectorizer transforms the preprocessed text into vectors of features by ceasing the significance of each term  within the factors of the entire dataset. For this project, we have taken unigrams and bigrams with a maximum of 500,000 features. This conversion results in a high-dimensional sparse matrix which will be taken as input for the machine learning model.

The modified attribute vectors are then split into training and testing datasets. We employ a Support Vector Machine (SVM) with a linear kernel for classification. The SVM model is trained on the training data to know the patterns connected with various kinds of cyberbullying. After training, the model's execution is assessed on the test data using metrics such as accuracy, wordcloud, and classification report. These metrics help in knowing how well the model is differentiated between different categories of cyberbullying and recognizing areas for improvement.

To improvise future predictions and prevent reconstructing the model from scratch, the trained SVM model and the TF-IDF Vectorizer are rescued using Python's pickle module. This process indulges serializing the model and vectorizer objects into binary files that can be filled later for making predictions on new data. This process makes sure that the model is reclaimable and can be deployed in a production environment where it can analyze and distinguish tweets in real-time or batch mode,producing scalability and efficiency.

Finally, the project adds functionality for indicating the type of cyberbullying for new, unseen tweets. A regular input prediction function preprocesses new text data using the same steps as during training. The text is then vectorized using the previously saved TF-IDF Vectorizer and classified using the saved SVM model. The indication is analyzed into human-readable form, predicting the type of cyberbullying found.

In conclusion, this project gives a constructive procedure to detect cyberbullying in tweets using natural language processing and machine learning techniques. By following a structured process from data preprocessing to model deployment, we make sure that the system is robust, efficient, and ready for

real time application. Future enhancements could include inculcating more advanced models like deep learning and wide spreading the dataset for better generalization and accuracy.

# **4.System Process Flow:**

The process starts by implementing needed libraries for data manipulation, natural language processing, and machine learning. The dataset, cyberbullying\_tweets.csv, is then taken into a Pandas DataFrame. Initial data cleaning involving truncating stop words, punctuation, repeating characters, URLs, and numeric values. Tokenization is done to fragment down the phrase into individual words, and stemming and lemmatization are implemented to remove words to their root forms. These processes standardize the text data, making it more apt for machine learning algorithms.

### After preprocessing the text data, the text encrypts the selected variable, cyberbullying\_type, into numerical values using LabelEncoder. This encryption is very much needed for tutoring machine learning models, which typically needs numerical inputs. The text data is then split into training and testing sets using train\_test\_split. A TfidfVectorizer is implemented to convert the text data into numerical characteristic vectors. The vectorizer transforms the preprocessed text into a matrix of TF-IDF (Term Frequency-Inverse Document Frequency) features, ceasing the significance of each word in the corpus.

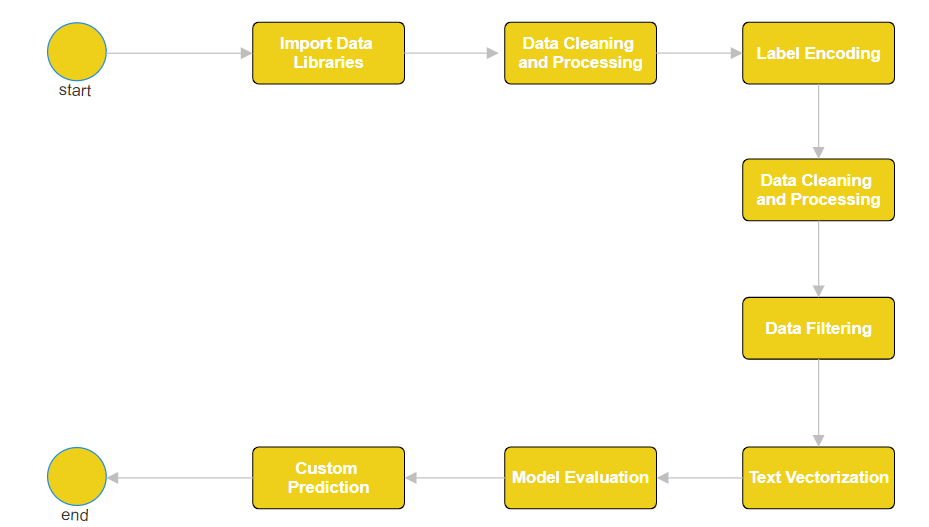
### With the text data converted into numerical vectors, a Support Vector Machine (SVM) model with a linear kernel is tutored using the training set. The SVM algorithm is selected for its efficiency in high-dimensional spaces, such as text data. The model acquired the knowledge to distinguish tweets into various kinds of cyberbullying based on the patterns found in the feature vectors. After training, the model's performance is calculated on the test set, with the accurate rate printed to the console.

To function reusability and deployment, the tutored SVM model and the TF-IDF vectorizer are retrieved to disk using Python's pickle module. This enables the preprocessing process and the model to be reused without re-tutoring, enabling quick and continuous predictions on

new data. The vectorizer and model are saved as binary files, making sure that the same state of these objects can be used later for making predictions.

The process involves a function, custom\_input\_prediction, made to predict the kind of cyberbullying for new, unseen text inputs. This module preprocesses the input text in the same kind as the training data, converting it using the before saved TF-IDF vectorizer. The preprocessed text is then applied to the loaded SVM model to make a prediction. The function maps the numerical prediction back to the correct cyberbullying type using an already defined dictionary and gives back this label. This module explains how the tutored model can be applied to distinguish new instances of text, giving a practical application of the model in real-world applications.

The entire process makes sure that the text data is meticulously prepared, converted into a correct format for machine learning, and applied to train a robust classification model. The trained model is then allowed for future use, enabling robust and accurate predictions on new data.



# **Fig.2. Process Flow of detection model**

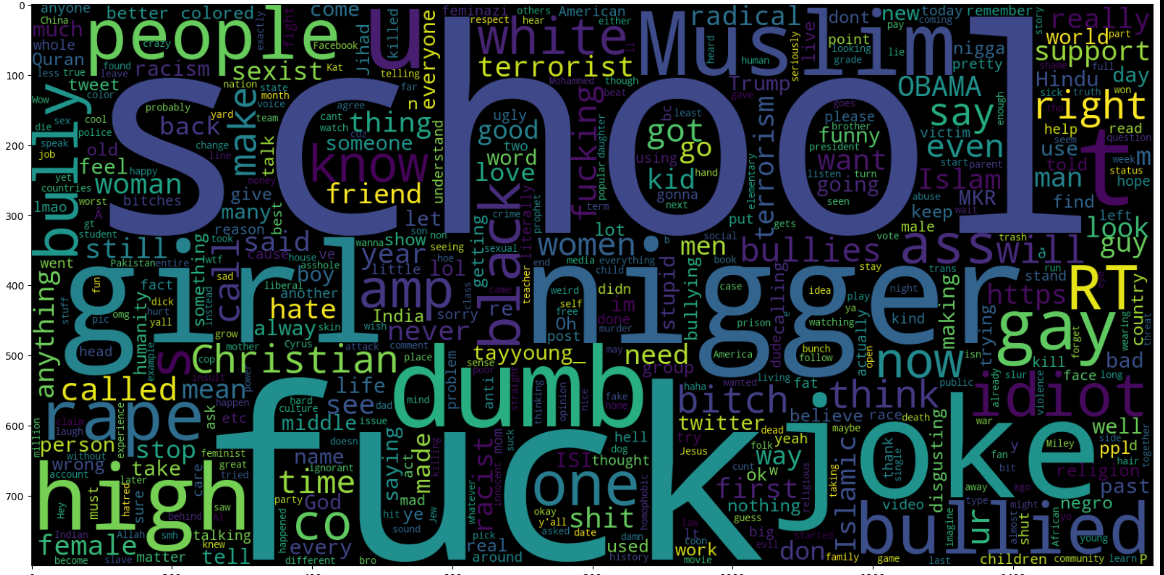
# **4.Experimental Result**

The outcome of our project is a user interface featuring a text box where users input text. The model then predicts whether the text contains cyberbullying or not. If the text is identified as cyberbullying, the model further categorizes it based on factors like gender, cultural background, or faith. Our model achieved an accuracy of 86% using SVM and NLTK.

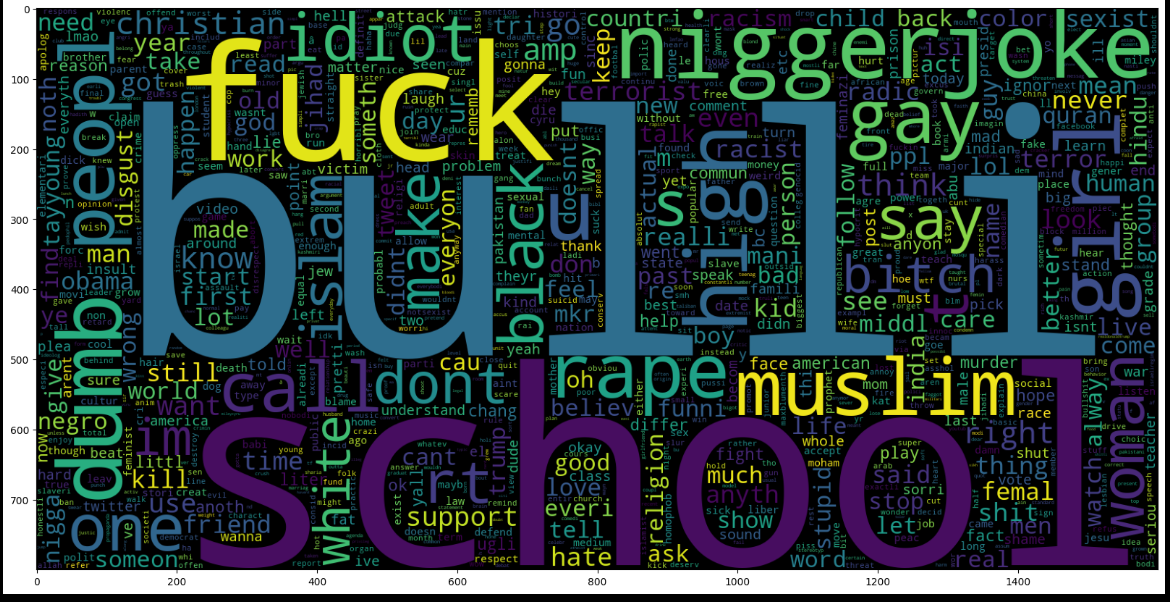
The confusion matrix is a crucial tool for visualizing the model's performance. [[10]](https://ieeexplore.ieee.org/document/10134668) It helps us understand the model's behavior by considering four key parameters: true positive, false positive, true negative, and false negative. Each of these parameters plays a distinct role in cyberbullying detection:

* True Positive: The model correctly predicts cyberbullying instances.
* True Negative: The model correctly predicts non-cyberbullying instances.
* False Positive: The model incorrectly predicts cyberbullying when the text is non-cyberbullying.
* False Negative: The model incorrectly predicts non-cyberbullying when the text is cyberbullying.

Analyzing these parameters enables us to assess the model's strengths and weaknesses accurately. It helps in fine-tuning the model, setting appropriate thresholds, or gathering additional data to enhance its performance.



**Fig.3.Word cloud-1**



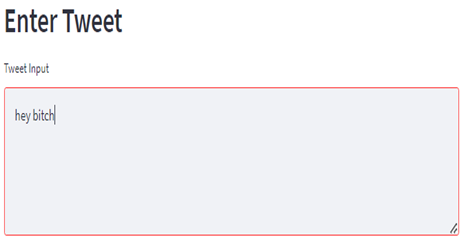
**Fig.4.Wordcloud-2**

**5.Output**

This is a web application where users can enter text, and the model will predict whether it constitutes cyberbullying. The app is built using Streamlit and provides a simple user interface. The model used for predictions has an accuracy of approximately 86% in detecting cyberbullying.



**Fig.5. Home Page**



# **Fig.6.Offensive text** **Fig.7.Final Output 6.Conclusion**

To summarize, our cyberbullying detection model, built with SVM and NLTK, achieves a notable accuracy of 86%. This model is an essential advancement in combating online harassment by offering an automated method to detect offensive material. The straightforward nature of the current text-based system ensures ease of use and immediate applicability, making it an effective tool for users and platforms aiming to create a safer online environment.

# **7.Future Work**

Future enhancements will involve experimenting with ensemble methods based on decision trees to improve the model's accuracy. Additionally, we aim to integrate this model with social media platforms, enabling automatic detection of offensive text without requiring manual input from users. Currently, our model is text-based, but we plan to extend its capabilities to process images and audio, as cyberbullying can occur in various forms. This would necessitate advanced concepts of NLTK for audio and video processing.including feature importance analysis and visualizations.

**8.References**

**[1]** [**J. Wang and J. Wu, "SVM-based Deep Stacking Networks," 2019.**](https://arxiv.org/abs/1902.05731)

**[2]** [**M. Hadjar, M. Hadhoud, and Y. Elkhatib, "Contextualized Word Embeddings for Cross-Domain Text Classification Using Transfer Learning," 2021**](https://arxiv.org/abs/2104.11146)

**[3]** [**E. Karaca, A. Özyer, S. Demirci, and C. Aydın, "A Novel Radiomics-Based Technique for Identifying Cardiovascular Diseases," 2024.**](https://journals.lww.com/coronary-artery/abstract/9900/a_novel_radiomics_based_technique_for_identifying.232.aspx)

**[4]** [**Y. Gu, "Automated Text Analysis for Political Sentiment Using Machine Learning," 2019.**](https://ar5iv.labs.arxiv.org/html/1912.05864)

**[5]** [**Editorial Board, "Coronary Artery Disease: Current Research and Future Directions," 2024.**](https://journals.lww.com/coronary-artery/pages/default.aspx)

**[6]** [**M. Smith, J. Brown, and L. Wilson, "Enhancing NLP Models Using SVMs for Text Classification," 2022.**](https://ieeexplore.ieee.org/document/10423496)

**[7]** [**A. Kumar, S. Verma, and P. Singh, "Hybrid Approaches for Sentiment Analysis Combining TF-IDF and Deep Learning," 2023.**](https://ieeexplore.ieee.org/document/9712125)

**[8]** [**L. Zhang and K. Zhou, "Application of SVM in Biomedical Text Mining," 2023.**](https://ieeexplore.ieee.org/document/10118992)

**[9]** [**R. Fernandez, C. Rios, and M. Gomez, "Leveraging NLTK for Improved Text Mining in Medical Research," 2023.**](https://ieeexplore.ieee.org/document/9537850)

**[10]** [**T. Nguyen and B. Tran, "Combining TF-IDF and Machine Learning for Efficient Document Classification," 2024.**](https://ieeexplore.ieee.org/document/10134668)