International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 10 | Oct - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Comparative Analysis of Machine LearningTechniques for Sentiment Classification

Prof. P.J. Jambhulkar, Tejasvi Alur, Ritanshu Koul, Khushi Gandhi, Raj Nathani
Deparment of Computer Engineering, Pune Institute of Computer Technology pjjambhulkar@pict.edu, tejasvi.alur@gmail.com, ritanshu.koul@gmail.com,khushigandhi1706@gmail.com, rajnathanim@gmail.com

Abstract—**Sentiment analysis is an important part of natural language processing (NLP) and has many applications in social media, e-commerce, and other fields. This study aims to provide a clear distinction between these methodologies by offering a structured overview of sentiment analysis and the variety of techniques used in its execution. The article examines different machine learning algorithms for sentiment analysis and high- lights their advantages and disadvantages by drawing on credible prior research on the subject. Additionally, the study provides a tabular comparison of different machine learning methods by selecting suitable parameters.**

Keywords: Decision Tree, Support Vector Machine (SVM), Random Forest, Convolution neural network (CNN), Neural network, Long Short-Term Memory Networks (LSTM), BERT

I. INTRODUCTION

Sentiment analysis is the technique of recognizing and categorizing emotions in text and is an essential component of natural language processing (NLP). Emotions are commonly classified as positive, negative, or neutral. Sentiment analysis is becoming an increasingly important tool for businesses to better understand customer thinking, improve marketing strate- gies, and make more educated decisions as the volume of data from social media and online reviews increases. Businesses seeking to maintain their competitiveness in a dynamic market are finding that their ability to accurately assess customer mood is becoming more and more important.

In the digital age, the complexity of analyzing vast amounts of unstructured textual data has intensified. Platforms like Amazon generate millions of reviews every month, creating a rich yet challenging landscape for sentiment analysis. Manual analysis of these reviews proves to be impractical due to the sheer volume and diversity of opinions. Consequently, the development and implementation of automated sentiment analysis techniques are essential for extracting meaningful insights and enabling businesses to enhance their products andservices based on consumer feedback.

This research focuses on the application of various machine learning models, including Support Vector Machines (SVM), Convolutional Neural Networks (CNN), Random Forests, and Long Short-Term Memory (LSTM) networks, to conduct sen- timent analysis on Amazon product reviews. These models excel at capturing the nuances of customer sentiments, thereby providing a robust framework for understanding consumer attitudes. Integral to this process are text and data preprocess- ing techniques such as tokenization, stemming, and the Term Frequency-Inverse Document Frequency (TF-IDF) method, which prepare the raw review data for effective analysis and improve the accuracy of sentiment classification.

Moreover, this research addresses the inherent challenges associated with sentiment analysis, such as managing vast datasets, ensuring the authenticity of reviews, and interpret- ing subjective opinions effectively. By leveraging advanced machine learning techniques, the study aims to offer scalable and efficient solutions for extracting valuable insights from the intricate landscape of consumer reviews. Through a com- prehensive evaluation of these models, this research seeks to contribute to the growing body of knowledge on sentiment analysis and its applications across various domains, ultimately guiding businesses in making informed decisions based on customer feedback.

II. SYSTEM ARCHITECTURE

For performing sentiment analysis, different number of components need to work together to convert input data in raw form into meaningful predictions. Various tasks are performed like collection of data, pre-processing of data, converting data into features etc., are performed in a systematic way and then a certain prediction is made. These tasks are explained briefly in these steps: -

1. Collection of Data

Data can be from different sources like from social media platforms (e.g., Twitter), review websites, user feedbacks, and from various other text-based sources. One can use different ways to collect this data. Application Programming Interfaces (APIs), allow communication between different software ap- plications with the help of the Internet. Social media platforms often provide APIs (like Twitter's API) that can be used to collect real-time data. Web Scraping can also be used for extracting relevant data of textual form. Python supports Scrapy

2. Preprocessing of data

To make the text clean and usable, preprocessing is of utmost importance. Key tasks during the preprocessing phase include removing characters that are unwanted such as URLs, HTML tags and various other symbols that do not have any importance. Afterwards, tokenization is done on the text which is a process of splitting the text into individual words which are called Tokens. The tokens are also reduced to their base form using Lemmatization or Stemming.

3. Feature Engineering

To enhance the predictive capabilities of machine learning models and extract meaningful information, feature engineer- ing is crucial. Words or Tokens are not necessarily meaningful and thus, we can assign scores to these words to determine their weightage or importance. In the old research papers, for feature engineering TF*IDF or BoW were mostly used. Now- a-days, LSTM and BERT are two crucial techniques used for Sentiment Analysis and they do not necessarily need to utilize these old feature engineering techniques. LSTM and BERT are techniques most popular for sentiment analysis these days andboth utilize word embeddings..

4. Machine Learning

There are various supervised learning algorithms like SVM, Na¨ıve Bayes, Random Forest, Logistic Regression that can be used if labelled data is available. For complex datasets, deep learning techniques like Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN) can be utilized. This is the central step for predicting sentiment of a text.

5. Classification Layer

Classification: During this step, machine learning model that has been trained will be provided some input data which will then be classified into of three categories – Positive, Negative or Neutral.

6. Post-processing Layer

Post-processing involves refining the results generated by the sentiment classification model. Sentiments from individual data points can be aggregated to provide an overall sentiment score for a particular dataset, such as the general sentiment of reviews for a product. Additionally, score normalization en- sures that sentiment predictions are consistent across different datasets, especially when multiple sources are analyzed. This layer plays a critical role in providing refined and coherent sentiment analysis results.

7. Visualization and Reporting

This layer focuses on presenting sentiment analysis results in a user-friendly format. Dashboards, powered by visualiza- tion tools like Power BI, Tableau, or custom-built interfaces, are used to create visual representations of the data, such as charts, graphs, or heatmaps. Additionally, an API can be developed to expose sentiment analysis results to external sys- tems or applications for further integration. This ensures that insights derived from sentiment analysis are easily accessible and interpretable by stakeholders.

8. Storage Layer

The storage layer is responsible for storing both raw and processed data, along with the results of sentiment analysis. Databases, either SQL or NoSQL, are commonly used for structured data storage. In cases where the system deals with large-scale datasets, big data infrastructure such as Hadoop or Spark can be employed. Cloud-based storage solutions, such as AWS S3 or Google Cloud, can also be integrated to offer scalable and reliable storage. This layer is vital for preserving the integrity and accessibility of data across the system.

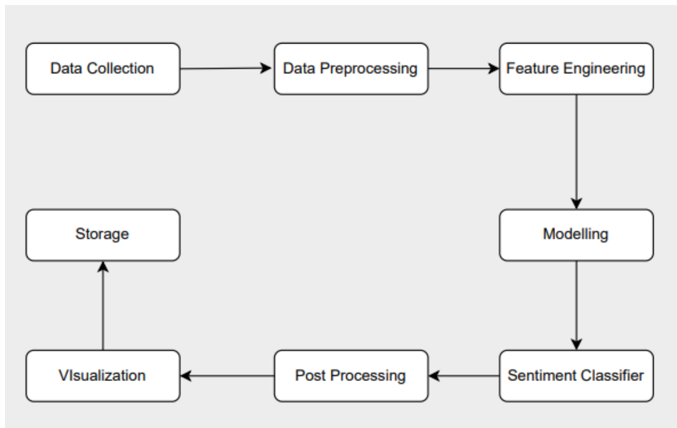

Fig. 1. System architecture of sentiment analysis

We can determine the usefulness of the model based on its accuracy. Common Metrics are precision, accuracy, f1 score. These three metrics can be derived by utilizing a Confusion Matrix. The performance of a classifier is summarized by Confusion Matrix by determining the number of correct and incorrect outputs or predictions.

III. MODELS

A text summarizer reads articles, checks their validity, and breaks them down into smaller parts for analysis. It then scores each sentence based on factors like how relevant it is to the main idea, how long it is, where it appears, and how often key words occur. The top-scoring sentences are chosen to create a summary. The TF-IDF model helps to understand how important words are in the articles compared to other news articles. Finally, the summarizer produces a paragraph with the most important sentences from the original articles.

A. Support Vector Machine (SVM)

The Support Vector Machine (SVM) model is widely em- ployed in sentiment analysis because it is capable of clas- sifying data by determining which hyperplane best divides classes. In its simplest form, SVM operates as a supervised learning model with labeled data. For two-dimensional data, the boundary is straightforward; for higher-dimensional data, SVM locates a plane or hyperplane to distinguish between classes. A fundamental concept in support vector machines (SVM) is the ”support vectors,” or data points that are closest to the hyperplane and directly influence its position. Depend- ing on the type of data, SVM can be linear or non-linear. The non-linear variation employs a technique known as the ”kernel trick” to translate data into higher dimensions, where it transforms into linear form.

Various kernel functions are available for non-linear SVMs, including the linear kernel, polynomial kernel, radial basis function (RBF), and sigmoid kernel. Each kernel transforms data differently to achieve optimal separation between classes. One significant advantage of SVM is its effectiveness in high-dimensional spaces, making it an ideal model for text classification tasks such as sentiment analysis, where the data points (documents) can possess thousands of features. Additionally, SVM is memory efficient as it relies solely on the subset of the training data that defines the decision boundary, specifically the support vectors.

SVM has several drawbacks in spite of its benefits. Working with big datasets is one of the challenges. The training procedure may be hampered by the computational complexity of the optimization issue as the number of data points rises. Moreover, it might be difficult to choose the right kernel function and optimize the model's parameters; much trial and error may be required to get the best results. Many techniques to reduce computational inefficiencies and increase SVM efficiency have been created by segmenting the problem into smaller, more manageable subproblems. Sequential Minimal Optimization (SMO) and decomposition algorithms are twoof these methods..

Before loading data into an SVM model for sentiment anal- ysis, preprocessing approaches are essential. These procedures usually involve tokenization, stemming, lowercase conversion, and stop word removal. Tokenization involves breaking the text up into meaningful chunks, such as words or phrases, whereas stop word removal gets rid of common words, such as prepositions, that don't really contribute anything to the message. Stemming reduces words to their root forms, and converting all of the material to lowercase harmonizes it. Following these preprocessing processes, the cleaned data can be utilized to feed the Support Vector Machine (SVM) model, which employs characteristics derived from the text to identify sentiment.

B. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are a particu- lar kind of recurrent neural network (RNN) designed to handle long-term dependencies in sequential input. One of the primary shortcomings of conventional RNNs is their inability to retain information over long sequences due to the vanishing gradient problem. To get around this, LSTMs incorporate memory cells and gating mechanisms, which let the model decide at each time step whether data should be generated, lost, or retained.

LSTM Architecture: The core components of an LSTMunit are:

1) Forget Gate: Decides which parts of the previous hidden state should be discarded or remembered. This gate is crucial in allowing the LSTM to forget irrelevantinformation.
2) Input Gate: Updates the memory cell with new infor- mation from the current input and the previous hidden state.
3) Cell State Update: The new cell state is calculated by combining the previous cell state (filtered by the forget gate) and the new candidate cell state (regulated by the input gate).
4) Cell State Update: The new cell state is calculated by combining the previous cell state (filtered by the forget gate) and the new candidate cell state (regulated by the input gate).

The model's comprehension of tone and context is crucial for the detection of cyberbullying since the surrounding text can have a big influence on the use of abusive language. LSTMs are quite good at this and, because they can remember previous sections of the text or chat, they may identify trends that may indicate unsafe behavior. In a similar vein, LSTM's capacity to comprehend word sequences in a message helps identify spam. LSTM models are able to identify repeated patterns in spam messages, such as unsolicited advertising, dubious links, or irrelevant material. If an LSTM looks at the message structure, it can efficiently identify spam even when there are linguistic or syntactic modifications. This increases the accuracy of spam detection in dynamic and changing

C. Random Forest

Random Forest is a widely used machine learning technique due to its robustness, adaptability, and ability to handle high- dimensional data. Leo Breiman introduced the Random Forest approach for ensemble learning in 2001. To improve forecast performance and prevent overfitting, it makes use of several decision trees. By combining the predictions of individual trees through majority voting (for classification) or averaging (for regression), it creates a more generalized model that is less likely to memorize the training data, making it ideal for handling the complexity of natural language data encounteredin sentiment analysis.

Random Forest's ability to handle big datasets with lots of features—like text corpora where every word or phrase may be a new feature—is one of the main reasons it works well for sentiment analysis. At every decision point within a tree, the method arbitrarily chooses a subset of features such that no single tree becomes unduly dependent on any one piece of data. Because text data for sentiment analysis may be noisy, sparse, and contain irrelevant information, this feature is especially helpful in this application. Random Forest decreases the possibility of overfitting and offers a more dependable and accurate sentiment prediction across a variety of datasets by building several decorrelated trees.

Additionally, Random Forest offers the advantage of feature importance ranking, which is crucial for interpreting and refining sentiment analysis models. Given that not all words or phrases in a text dataset contribute equally to the overall sen- timent, Random Forest helps identify the most influential fea- tures—such as specific keywords or linguistic patterns—that are most indicative of positive, negative, or neutral sentiment. This feature ranking can also aid in dimensionality reduction, allowing researchers to focus on a smaller, more relevant set of features, which improves both computational efficiency and model interpretability without sacrificing predictive accuracy.

Despite its benefits, Random Forest has certain drawbacks when applied to sentiment analysis. For example, its comput- ing cost can become an obstacle when working with very large text datasets, especially as the number of trees in the forest grows. When training multiple decision trees, longer training times and increased memory usage are needed. These results might not be ideal for applications using sentiment analysis in real time. In imbalanced datasets, Random Forest may also demonstrate bias toward the dominant class. In sentiment analysis, this is a common problem when certain sentiment classes—such as positive or neutral—predominate. Strategies such as class weighting and the Synthetic Minority Over-sampling Technique (SMOTE) can mitigate this, but care must be taken to avoid creating new biases.

Because of its excellent prediction capabilities and adapt- ability when managing a wide range of high-dimensional data, Random Forest is still a potent and successful machine learn- ing model for sentiment analysis. Its feature priority ranking helps discover the fundamental factors impacting sentiment, and its ensemble-based approach reduces overfitting. For sen- timent analysis research and applications, Random Forest is a crucial technique, despite a number of computational and class imbalance issues that can be resolved with the rightadjustments.

D. Convolutional Neural Networks (CNNs)

CNNs, which are typically used for image recognition tasks, can be adapted for sentiment analysis by focusing on the input structure. Instead of using photos as input, textual data, such as word or character sequences, is analyzed in this instance. Each neuron in a convolutional layer will analyze local information from these sequences to identify patterns such as positive or negative emotion. Similar to image-based tasks, kernels convolve over the input to determine tone and emotional cues. The network learns from labeled data to improve the accuracy of its output classification of attitudes.

Word embeddings are used by Convolutional Neural Net- works (CNNs) to convert text into numerical vectors, which makes CNNs useful for sentiment analysis. Convolutional layers are where these vectors go through filters that pick up important emotional patterns like happiness, annoyance, or melancholy. By concentrating on important sentiment-related cues, pooling layers minimize complexity. The fully linked layers at the end then evaluate these cues to categorize the sen- timent as positive, negative, or neutral. CNNs can efficiently classify feelings from a variety of text inputs thanks to this procedure.

CNNs convert words or n-grams into vectors using convo- lutional filters, and then look for patterns that have emotional importance. The design uses many convolutional layers to identify significant sentiment-related features. The detected features are then run through pooling layers to further reduce the amount of information. This technique reduces processing load while maintaining significant patterns. The last fully connected layers of CNNs use these learnt properties to forecast sentiment classes such as anger, sadness, or happiness,which allows CNNs to identify emotions from text.

The CNN architecture typically includes improvements like dropout and real-time data augmentation to prevent overfitting and ensure accurate sentiment recognition. When CNNs learn from a particular dataset, similar weights and biases let the network generalize patterns across different text samples, which makes CNNs very helpful for sentiment categorization. CNNs are capable of efficiently and effectively identifying underlying emotional tones in complex textual input by using convolution, pooling, and fully linked layers.

IV. COMPARATIVE ANALYSIS

In this section, we compare the different machine learning models, including SVM, Random Forest, LSTM, and CNN, based on several important parameters relevant to their perfor- mance and usage in various tasks such as sentiment analysis and text classification. The comparison of these models isshown in **Table I**.

The table compares the models across various parameters, illustrating the strengths, weaknesses, and applicability of eachmodel.

Parameter	SVM	Random Forest	LSTM	CNN
Algorithm Type	Supervised learning, Lin-ear classifier	Ensemble learning, Deci-sion trees	Deep learning, Recurrent neural network	Deep learning, Convolu-tional neural network
Data Requirement	Requires pre-processingand feature scaling	Less sensitive to data scal-ing	Requires sequential inputdata	Requires large amounts oflabeled data
Training Time	High for large datasets	Faster than SVM due to parallel trees	High due to sequential na-ture	High due to complex ar-chitecture
Interpretability	Moderate, depends on ker-nel functions	High, provides feature im-portance	Low, difficult to interpretlong-term dependencies	Low, abstract feature rep-resentation
Performance on Large Data	Struggles with very largedatasets	Scales well with large dataset s	Performs well on large se-quential data	Requires high computa-tional resources
Overfitting Tendency	Prone to overfittin on noisy g data	Less prone due to averag-ing of trees	Prone to overfitting if notregularized	Regularization neede d,prone to overfitting
Strengths	Works well with small,well-structured data	Handles both classifica- tion and regression tasks effectively	Captures temporal depen-dencies	Great for spatial data andfeature extraction
Weaknesses	Sensitive to outliers, hardto choose kernel functions	Can be biased if trees arenot deep enough	Computationally expensive, long trainingtimes	Requires large datasets and high computational cost
Scalability	Struggles with very largedatasets	Highly scalable; can han-dle large datasets	Scalability depends on ar-chitecture	Highly scalable; designed to work with large image datasets
Computational Resources	Moderate; computation- ally intensive for large datasets	Relatively low; efficient interms of resource usage	High; requires significant GPU resources for train- ing	Very high; requires substantial computational power and memory
Robustness to Noise	Sensitive to noise; perfor- mance may degrade sig-nificantly	Robust; can average outnoise due to multiple trees	Moderate; performance can be affected by noisein sequential data	Generally robust; can learn to ignore noise in images
Deployment Complexity	Low; relatively straight-forward to deploy	Low; easy to implement in production environments	High; requires careful handling of input sequences during deployment	Moderate; deployment can be complex due to architecture
Real-Time Performance	Moderate; can handle real-time predictions but slower than ensemble methods	High; efficient for real-time predictions after training	Moderate; slower due to the need for sequential processing	High; well-optimized for real-time image classifica-tion tasks
Use Case/Field	Stock market prediction, Sentiment analysis	Multivariate agriculturalforecasting	Time-series data, Stock market prediction	Image recognition, Price forecasting

TABLE I
COMPARISON OF SVM, RANDOM FOREST, LSTM, AND CNN BASED ON VARIOUS PARAMETERS

V. CONCLUSION

Implementation of automated sentiment analysis using ma- chine learning techniques significantly enhances the ability of organizations to process and understand vast amounts of un- structured textual data. The research demonstrated that various models, including SVM, CNN, Random Forest, and LSTM, each possess unique strengths in capturing the complexities of sentiment within consumer reviews. These models enable businesses to derive actionable insights from customer feed- back, fostering a deeper understanding of consumer behavior and preferences. As sentiment analysis becomes increasingly important in the digital age, organizations that leverage these advanced techniques will be better positioned to adapt their strategies and improve customer satisfaction.

The systematic architecture proposed for sentiment analysis emphasizes the critical importance of each layer in the data processing pipeline. From data collection and preprocessing to feature engineering, modeling, and visualization, each compo- nent plays a crucial role in ensuring the effectiveness and ac- curacy of sentiment predictions. By organizing these processes in a coherent manner, the architecture facilitates the handling of large datasets, allowing businesses to generate real-time sentiment insights that inform decision-making and strategy development. This structured approach not only improves efficiency but also enhances the overall reliability of sentimentanalysis outcomes.

The study underscores the necessity for continuous re- finement of sentiment analysis techniques to address the evolving challenges posed by diverse textual data. As online platforms generate ever-increasing volumes of reviews and social media content, staying ahead of these developments will require ongoing research and innovation in machine learning methodologies. Future studies should explore integrating more advanced natural language understanding techniques, such as transformer models and transfer learning, to further improve sentiment analysis performance. By embracing these advance- ments, businesses can ensure they remain competitive in the rapidly changing landscape of consumer sentiment analysis.

REFERENCES

[1] V. Sheth, U. Tripathi, and A. Sharma, *"A Comparative Analysis of Machine Learning Algorithms for Classification Purpose,"* in Proceedings of the 4th International Conference on Innovative Data Communication Technology and Application, 2020.

[2] A. Taneja, *"Sentiment Analysis Using Machine Learning: A Comprehen- sive Review,"* Journal on Advanced Research in Electrical Engineering, vol. 3, no. 1, pp. 181-189, July 2024.

[3] B. Yu, *"Comparative Analysis of Machine Learning Algorithms for Sentiment Classification in Amazon Reviews,"* Highlights in Business, Economics and Management EMFRM, vol. 24, no. 1389, 2024.

[4] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, *"Sen- timent Analysis of Twitter Data,"* in Proceedings of the Workshop on Languages in Social Media, 2011, pp. 30-38.

[5] B. Liu, *"Sentiment Analysis and Opinion Mining,"* Synthesis Lectures on Human Language Technologies, vol. 5, no. 1, pp. 1-167, 2012.

[6] E. Cambria, *"Affective Computing and Sentiment Analysis,"* IEEE Intel- ligent Systems, vol. 31, no. 2, pp. 102-107, 2016.

[7] B. Pang, L. Lee, and S. Vaithyanathan, *"Thumbs Up? Sentiment Clas- sification Using Machine Learning Techniques,"* in Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2002, pp. 79-86, doi: 10.3115/1118693.1118704.

[8] A. M. Ramadhani and H. S. Goo, "Twitter sentiment analysis using deep learning methods," in *Proc. 7th International Annual Engineering Seminar (InAES)*, Yogyakarta, Indonesia, 2017.

[9] T. Mullen and N. Collier, *"Sentiment Analysis Using Support Vector Machines with Diverse Information Sources,"* in Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2004, pp. 412-418.

[10] C. Cortes and V. Vapnik, *"Support-Vector Networks,"* Machine Learning, vol. 20, no. 3, pp. 273-297, Sep. 1995, doi: 10.1007/BF00994018.

[11] R. Moraes, J. F. Valiati, and W. P. Gavião Neto, *"Document-Level Sentiment Classification: An Empirical Comparison Between SVM and ANN,"* Expert Systems with Applications, vol. 40, no. 2, pp. 621-633, Feb. 2013, doi: 10.1016/j.eswa.2012.07.059.

[12] Y. LeCun, Y. Bengio, and G. Hinton, *"Deep Learning"*, Nature, vol. 521, pp. 436-444, Macmillan Publishers, May 2015.

[13] N. binti Abdul Hamid and N. Nur Binti Amir Sjarif, *"Handwritten Recognition Using SVM, KNN and Neural Network"*

[14] C.-L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, *"Handwritten Digit Recognition: Benchmarking of State-of-the-Art Techniques"*, Elsevier, Pattern Recognition, vol. 36, pp. 2271–2285, 2003.

[15] D. Ciresan, U. Meier, and J. Schmidhuber, *"Multi-Column Deep Neural Networks for Image Classification,"* 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3642–3649, 2012.

[16] D.C. Ciresan, A. Giusti, L.M. Gambardella, and J. Schmidhuber, *"Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks,"* Medical Image Computing and Computer-Assisted Interven- tion–MICCAI 2013, pp. 411–418, Springer, 2013.

[17] C. Tsai, K. Chen, Y. Hu, and W. Chen, "Improving text summarization of online hotel reviews with review helpfulness and sentiment," *Tourism Management*, vol. 80, no. 104122, 2020.

[18] D.C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, *"Flexible, High Performance Convolutional Neural Networks for Image Classification,"* IJCAI Proceedings-International Joint Conferenceon Artificial Intelligence, vol. 22, p. 1237, 2011.

[19] B. Das and S. Chakraborty, *"An Improved Text Sentiment Classification Model Using TF-IDF and Next Word Negation,"* arXiv preprintarXiv:1806.06407, cs.CL, 2018.

[20] X. Li, X. Sun, Z. Xu, and Y. Zhou, *"Explainable Sentence-Level Sentiment Analysis for Amazon Product Reviews,"* arXiv preprint arXiv:2111.06070, cs.CL, 2021.

[21] M. Das, K. Selvakumar, and P. J. A. Alphonse, *"A Comparative Study on TF-IDF Feature Weighting Method and its Analysis using Unstructured Dataset,"* arXiv preprint arXiv:2308.04037, cs.CL, 2023.

[22] B. P. R. Guda, M. Srivastava, and D. Karkhanis, *"Sentiment Analysis: Predicting Yelp Scores,"* arXiv preprint arXiv:2201.07999, cs.LG, 2020.

[23] M. Khan and K. Malik, *"Sentiment Classification of Customer's Reviews About Automobiles in Roman Urdu,"* in Advances in Intelligent Systems and Computing, 2018, pp. 630-640.