[1] L. Sudha Rani
[2] Dr. S. Zahoor-Ul Huq
[3] Dr. C. Shoba Bindu

Comprehensive Analysis of Twitter Sentiment Using Machine Learning Algorithms for Enhanced Sentiment Prediction

Journal of Electrical Systems

Abstract: - The principal aim of e-commerce systems is to improve the consumer experience, with customer input being essential to this objective. This study introduces an innovative method for analysing tweet data to enhance customer happiness. Key tweet features were retrieved to capture the sentiment expressed in customer tweets by utilising a combination of n-gram models and word embeddings. The features were subsequently employed to construct classification models utilising Support Vector Machines (SVM) and Artificial Neural Networks (ANN), both of which are frequently applied in sentiment analysis. A model based on Convolutional Neural Networks (CNN) was proposed to enhance the accuracy of classifying sentiment by categorizing tweets as either positive or negative. The study's results indicated that the CNN model surpassed both the SVM and ANN models, exhibiting more accuracy in sentiment classification of tweets. In addition to its precision, the CNN model offered significant insights into the interrelations among different tweet categories, enhancing comprehension of client mood and views. This extensive study underscores the need of understanding the complexities of customer feedback to improve e-commerce strategies and consumer delight as well as the efficiency of the CNN model for sentiment categorisation. By means of exact classification and analysis of sentiment in real-time twitter data, e-commerce systems may more successfully evaluate customer impressions and react dynamically to enhance their whole experience.

Keywords: Sentiment Analysis, Tweet Analysis, CNN, ANN, SVM

I. INTRODUCTION

Driven by ongoing developments in mobile technology, the worldwide use of smart phones has exploded with over 2.71 billion users actively engaged across several platforms. Many of these people often connect on social media, which produces a great amount of material reflecting a wide range of emotions and ideas. Usually falling into three categories—negative, neutral, and positive—these emotional expressions are triggered by travel, dining, movie watching, and political event participation. Therefore, social media is a great source of data since it provides perceptive knowledge on user satisfaction and discontent, which makes it a necessary tool for companies aiming on improving client experiences.

Opinion mining, or sentiment analysis, is a powerful technique for detecting and interpreting emotions or opinions expressed in textual data. For companies, it is now a vital instrument since it enables them to grasp consumer attitude on many different subjects. Though sentiment analysis has many applications, it also presents various difficulties such domain reliance, false or misleading data, and processing complexity of natural language including bipolar terminology. Improving sentiment analysis system accuracy and efficiency depends on addressing these difficulties. Researchers have made great progress in creating sentiment analysis models in recent years; deep learning especially shows great promise. Particularly with regard to big datasets, deep learning methods can automatically extract features and learn representations more suited for sentiment classification.

Focussing especially on sorting consumer reviews into positive and negative categories, this work attempts to leverage deep learning to improve sentiment analysis. This study makes use of a dataset comprising Twitter data as well as Amazon product reviews, therefore offering a variety of user comments. Advanced pre-processing techniques such as stemming, tokenization, and stop words removal were employed to optimize the dataset for sentiment classification. To achieve accurate sentiment categorization, we applied a deep learning-based approach using CNN with both conventional ML models—including ANN and SVM. Our comparison analysis showed that the CNN model beat the conventional models by means of its potential to grasp the semantic structure of the text, therefore offering superior accuracy and less computational complexity. Moreover, the CNN-based model exposed significant correlations across many Twitter categories, therefore providing more complete knowledge of consumer mood patterns.

[1]* Research Scholar, Dept. of CSE, JNTUA Anantapur, Asst.Professor,G.Pulla Reddy Engineering college, Andhra Pradesh. Mail Id:sudha10211@gmail.com.
[2]Professor, Dept. of CSE, G. Pulla Reddy Engineering College, Kurnool
[3]Professor, Dept. of CSE, JNTUA Anantapur, Andhra Pradesh, India

This work provides a strategy that lowers computing complexity while raising classification accuracy, therefore supporting the continuous evolution of sentiment analysis methods. This study supports the development of more strong and effective apps in e-commerce and other sectors that mostly depend on knowledge of client sentiment by proving the possibilities of deep learning to advance sentiment analysis.

II. Literature Review

Techniques to improve sentiment analysis and customer review mining performance have been thoroughly investigated by researchers. Extraction of sentiment from textual data is among the most often used techniques. Using deep learning methods to assess textual content, the authors in [6] offer a flexible and effective approach for analysing such data. Consumer reviews have surged in response to the growing popularity of digital channels. Researchers have so been looking at several approaches for handling this enormous volume of data. Based on [7], in sentiment analysis, ML approaches offers some precise prediction outcomes. However, they noted that certain topic modeling methods may not be well-suited for generating accurate predictions. Sentiment analysis aims ultimately to give a thorough knowledge of consumer thoughts about a product, thereby pointing out important elements influencing its success. Examining the evolution of the Aspect-Based Sentiment Analysis (ABSA) model within Hindi literature [8] helps this article also extend the scope.

In another work [9] the sentiment in different reviews was evaluated using NLP combined with ML. The authors also improved client satisfaction and simplified company processes using Microsoft's Power BI tool. For businesses and prospective consumers involved in the creation of new items, the above-described studies are intimately tied to knowledge of consumer feedback on past bought products. The research presented in [10] how sentiment analysis and business intelligence can benefit both consumers and producers alike. It also looks at several pragmatic uses for these instruments [11] and presents a long-term forecasting model to project consumer review results based on ML and NLP approaches.

Many studies have looked at several approaches for sentiment classification of social media data, particularly from sites like Twitter. Experimenting with public Twitter datasets using machine learning techniques including SVM, RF,NB, and Logistic Regression to enhance accuracy, Nadia et al. [12] suggested a sentiment classification method that employs lexicons and classifier ensembles. Medhat et al. [13] compiled a thorough review of 54 papers on sentiment analysis methods, therefore offering a summary of the generally applied sentiment classification and feature selection methods in this discipline. Their results show that sentiment classification challenges routinely use SVM and Naive Bayes.

In sentiment analysis, Ramasamy et al. [14] emphasized the importance of pre-processing, showing that utilizing SVM with proper feature selection leads to improved classification outcomes. Showing the value of integrating dimensionality reduction and lexicon-based techniques, Nurulhuda et al. [15] used Principal Component Analysis (PCA) with a SentiWordNet lexicon and SVM for sentiment classification.

On a polarity-based movie dataset, Abinash et al. [16] contrasted Naive Bayes with SVM. Their investigation shown the competitive character of these algorithms and categorised the data depending on favourable and negative reviews. In a similar vein, Shiyang [17] suggested a neural network-based method to examine Twitter sentiment, therefore highlighting the ability of deep learning to more profoundly grasp human emotions.

In order to assess time-changing data streams and so improve the accuracy of real-time sentiment analysis, Albert and Eibe [18] finally examined the difficulties in sentiment categorisation of Twitter data and suggested a sliding window-based kappa value statistic.

With ML methods (for example SVM, NB, and NN) routinely used, often complemented by sophisticated techniques such text pre-processing, feature selection, and dimensionality reduction to improve classification accuracy, these studies underline the evolving character of analysis.

III. Methodology

A. ***Data Collection***

The first part of data collecting is identifying appropriate sources from which to get the required information. This research made use of information gathered from multiple sites, including reviews, comments, and social media entries. Search criteria were set prior to data collecting to focus on customer evaluations and relevant themes. News feeds, product reviews, and tweets are among the often-used sentiment analysis datasets that fit mining and analysis in this field.

When the relevant data is found, it is further handled by a system designed to examine and classify sentiment from the data. Table 1 table shows the datasets utilised for the sentiment analysis method, therefore illustrating the harmony between positive and negative examples in several fields. Four separate datasets used for sentiment analysis are compiled in the table, which also highlights the distribution of positive and negative instances throughout many fields. With 14,000 reviews in total, the Amazon Products dataset shows a generally positive attitude towards Amazon products with most of 13,252 classed as positive and 750 as unfavourable. With 100,000 reviews, the biggest dataset available, the Cell Phones and Accessories one likewise indicates a predominance of favourable comments, with 87,516 positive evaluations against 12,484 negative ones. Comprising 25,000 evaluations, the Food evaluations dataset shows a similar tendency with 21,783 positive and 3,217 negative cases implying consumers generally have good opinions of food items and services. Finally, the Yelp dataset—the smallest with 4,000 reviews—also shows more positive (3,267) than negative (735) reviews, therefore indicating widespread contentment with local companies. All four datasets show positive sentiment predominance; hence they are suitable for teaching sentiment analysis models in different sectors.

Table 1: Dataset Description

S. No.	**Dataset**	**Positive-Instance**	**Negative Instance**	**Total**
1	Amazon Products	13252	750	14000
2	Cell Phones and Accessories	87516	12484	100000
3	Reviews of Food	21782	32178	25000
4	Yelp	3268	734	4000

B. Data Preprocessing

The process of text analysis begins with data preparation, which is particularly crucial in sentiment classification tasks. Raw data gathered from sites like Twitter sometimes includes noise and pointless bits that might hide important information. Data cleaning is necessary for extracting essential information and improving the model's performance. The preparation actions taken on the dataset are described in this part, with particular attention to deleting extraneous tweets.

Algorithm 1: Tweet Pre-processing
Input: Twitter Data obtained from Amazon Customer reviews Output: Tweet pre-processing **Initialize an Empty List**: Create an empty list PTF that will store the raw text data from each element in TC. **Loop Through TC (a Collection of Data Files)**: Start a loop to iterate over each element TW in TC. Inside the loop: Load the data from each TW using json.load(TW). This assumes that each element TW is in JSON format. Extract the value corresponding to the key 'Raw Text' from the loaded data. Append this raw text to the PTF list. **Iterate Over PTF (List of Raw Text)**: Once the PTF list is populated with all the raw text data, start another loop to iterate through each element TW in PTF. **Extract Specific Information from Each TW**: Use regular expressions to find and extract specific patterns from each text element: **Retweets (DRT)**: Use re.findall(r"RT@\w+", TW) to find and store any retweets that start with "RT@". **Hashtags (DHT)**: Use re.findall(r"#\w+", TW) to find and store any hashtags that start with #. **User Mentions (DU)**: Use re.findall(r"@\w+", TW) to find and store any user mentions that start with @. **URLs (DUL)**: Use re.findall(r"http(s)?://\S+", TW) to find and store any URLs starting with

http or https. **Clear Extracted Values**: After extracting the values from TW, set all extracted data (DRT, DHT, DU, DUL) to [None] to reset or clear the variables. **End the Process**: The process concludes once all elements in PTF have been processed.

The twitter preparation procedure comprises several essential phases, outlined below. These processes guarantee the raw data is primed for subsequent analysis by eliminating extraneous elements such as retweets, hash tags, usernames, and URLs. These elements generally do not enhance sentiment analysis and may impair model accuracy.

Step 1: Initialization

The pre-processing begins with initializing an empty structure that will hold the parsed tweet data. In this context, the raw tweet data is loaded from the dataset, and this step ensures that the text of each tweet is extracted and stored in a suitable format for further processing. This involves reading and loading the data from various JSON files, where each tweet is represented as structured text.

Step 2: Loading the Tweets

Once the data structure is set up, the next step is to load the individual tweets from the dataset, which in this case, is a combination of Twitter data and Amazon customer reviews. Each tweet is extracted from the corpus and its raw text is parsed. This raw text is then stored in a list or table that will be used in subsequent steps for cleaning and analysis. The goal here is to capture the unprocessed text content from each tweet, as this will form the core of the sentiment analysis process.

Step 3: Cleaning and Preprocessing

With the raw tweets now stored in a structured format, the most important phase of pre-processing begins cleaning the text data. This step involves identifying and removing irrelevant components from the tweets that do not contribute to sentiment detection but instead add noise to the dataset. Several specific elements are targeted for removal

Retweets: Tweets that are shared by other users, typically identified by the "RT@" pattern, are removed. Retweets are often repetitive and may skew the analysis since they reflect the same sentiment multiple times.

Hashtags: Hash tags, such as #example, are commonly used in social media to tag content, but they often do not contribute directly to sentiment and can vary widely across domains. By removing them, we reduce noise in the data.

Usernames: Twitter handles (e.g., @username) are primarily used for mentions and interactions but hold little value in determining the sentiment of a tweet. These are removed to avoid unnecessary distractions in the text.

URLs: Links (e.g., http://example.com) that point to external content are frequently present in tweets but do not provide any direct contribution to the text's sentiment. These are also identified and stripped from the data to clean the tweet further.

Regular expressions, effective tools for pattern recognition and text extraction, are used to identify each component. Once identified, they are removed from the tweet text, leaving only the core content that is likely to contain meaningful sentiment information.

Step 4: Finalizing the Pre-processed Data

After the tweets are cleaned and unnecessary elements have been removed, the processed data is stored in a refined structure. This cleaned dataset is now ready for further analysis and is suitable for feeding into machine learning models. At this point, the tweets consist purely of the text that expresses the user's sentiment, free from distractions like retweets, hashtags, usernames, and URLs.

Additional Pre-processing Techniques

Though not explicitly described in the algorithm, several other preprocessing techniques are often applied to further improve the dataset:

Stop-Word Removal: Common words that appear frequently but carry little meaning in terms of sentiment (e.g., "the", "is", "and") are removed. This reduces noise and helps the model focus on more meaningful words.

Tokenization: Each tweet is split into individual words or tokens, making it easier to analyze and model.

Stemming or Lemmatization: Words are converted to their base form (for instance, "eating" is simplified to "eat") to unify the vocabulary and enhance dataset consistency.

Pre-processing plays a critical role in sentiment analysis, particularly when working with social media data like tweets. Tweets are inherently noisy, often containing irrelevant information that could detract from meaningful analysis. By cleaning the data and removing non-contributory elements, the pre-processing pipeline ensures that the model will focus on the actual text that carries sentiment.

For example, retweets and hash tags can lead to biased or duplicated entries, while usernames and URLs provide no sentiment value. Removing these items helps streamline the data, making it more representative of the actual sentiment being expressed. This results in a more accurate and reliable model, with improved generalization and performance. Moreover, advanced techniques like stop-word removal and tokenization enhance the data further by filtering out redundant information and breaking down the text into manageable components. These steps ensure that the data is in the best possible state for sentiment classification, ultimately leading to better insights and predictions.

Algorithm 2: Glove Method for Feature Extraction
Input: TD -> Tweet Data Vc -> Vocabulary (all unique words in the dataset) Th -> Frequency Threshold (minimum frequency for a word to be considered) Ws -> Window Size (the context window size for word co-occurrence) Output: A Sparse Matrix representing the tweet data as word vectors. Begin Initialize the word embedding matrix Wc, with dimensions determined by multiplying the vocabulary size Vc by the vector dimensions X. Initialize the bias term B with random values within the range of -0.5 to +0.5. Repeat the following steps until convergence: Calculate the cost function. For each iteration: Compute the gradient of the cost function using the current parameters. Update the parameters based on the learning rate. End

C. Feature Extraction

Feature extraction is a crucial step in sentiment classification because it involves converting raw text into a numerical format that ML models can process. We used two main approaches for feature extraction in this work: n-gram models and word embeddings derived from GloVe. Accurate sentiment categorization depends on the semantic and contextual interactions between words, which these techniques assist to capture. With the Twitter dataset, every tweet is limited to 140 characters, therefore restricting the total information per entry. We turned the words into vector representations using word embeddings in order to adequately depict the text data. Because they preserve the semantic ties between words by encoding them into dense vectors, word embeddings are rather strong. This helps the model to grasp the setting in which a word appears instead of considering every word as an autonomous entity.

Using the GloVe (Global Vectors for Word Representation) algorithm—which considers co-occurrences of terms across the dataset—we turned the tweets into vectorised forms for this work. By means of global statistical information analysis of a corpus and hence efficient reduction of the dimensionality of the word space, the GloVe method generates word embeddings This approach generates vectors representing each word in a high-dimensional space by helping to capture associations between words depending on their frequency of of appearing together in different contexts.

The GloVe algorithm for feature extraction begins by initializing two key components: the word embedding matrix and the bias terms (B). The word embedding matrix is initialized with dimensions based on the vocabulary size.which represents all unique words in the dataset, and a predefined vector dimension, which determines the size of the word vectors. Each word in the vocabulary will have a corresponding vector of length X. This matrix will be used to store the vector representations of the words.

Next, bias terms (B) are initialized for each word. These bias values are set to a random range between -0.5 and +0.5 and play a crucial role in fine-tuning the model during training. The bias terms allow the model to account for specific word co-occurrence tendencies that may not be captured by the word vectors alone.

Once the word embedding matrix and bias terms are initialized, the algorithm proceeds by calculating the cost function. The cost function measures the difference between the actual word co-occurrences (how often pairs of words appear together in the dataset) and the predicted co-occurrences based on the current word vectors. The algorithm aims to minimize this cost function by adjusting the word vectors and biases iteratively.

During each iteration, the GC function is computed to determine the direction and magnitude of adjustments needed for the word vectors and bias terms, with these adjustments applied according to a predefined learning rate that controls the update size. The learning rate ensures that the parameters are updated gradually, allowing the model to converge toward an optimal set of word embeddings. This process of computing the cost function, calculating the gradient, and updating the parameters is repeated until the cost function converges, meaning the model has found the best representation of the words in the dataset.

D. Sentiment Classification Approach

In this study, we utilized ML algorithm for sentiment classification SVM, CNN, and ANN, each offering distinct advantages. We provide a thorough explanation of how these techniques are applied to tweet analysis

1. Support Vector Machine (SVM)

SVM are popular machine learning methods that find widespread use in sentiment analysis and other classification problems. Using SVMs, we can find the optimal hyper plane that divides the data into two groups, like positive and negative emotions. Locating the hyper plane that widens the chasm between the two sets of data is where its real power resides. The hyper plane is separated from the closest data points from each class by this space, called the margin. These points are called support vectors. The optimal hyper plane is the one that maximizes this margin, making the classification model more dependable and less susceptible to errors when new data is introduced. In this research, we evaluate the effectiveness of SVM for sentiment classification and compare it with other methods.

2. Artificial Neural Networks (ANN)

ANN offer a flexible approach to classification, effectively handling both numerical and categorical data.ANN is structured as a series of interconnected layers of neurons, where each neuron processes input values and applies weights to produce an output. For sentiment classification, the design of the network is influenced by the dataset's complexity and the features being assessed.

During training, the weights in each layer are modified to reduce prediction errors, thereby enhancing the accuracy of the classification model. A specific variation of ANN, known as Back propagation ANN (BPANN), is particularly useful in fine-tuning the model. In BPANN, the error predicted is transmitted backward through the network, enabling the model to refine the weights of earlier layers and thereby reduce the error. This iterative process of adjusting the weights continues until the model reaches a higher level of accuracy. In our study, we utilized ANN to assess its ability to classify sentiment and compare its results with other machine learning techniques.

3. Convolutional Neural Networks (CNN)

CNN previously developed for image processing, have evolved into a powerful tool for text classification, including sentiment analysis. CNN is applied to classify opinions based on affirmative or negative tweets. CNNs excel in this task because they can capture spatial relationships among words and extract advanced features through convolutional process.

The first step in using a CNN for tweet classification is to represent the tweets in a high-dimensional space. Specifically, if each word in the tweet is represented by a word embedding of dimension d and the tweet contains l words, then the tweet can be represented as a matrix of size $l \times d$. This matrix serves as the input to the CNN, where convolution filters are applied to capture sentiment-related features.

CNNs use different filters to scan the tweet, detecting patterns or features that indicate sentiment. Following the convolution layer, a Max Pooling layer is utilized to decrease the output's dimensionality and emphasize the most significant features, those with the highest values or "scores." This step helps the network to retain the most relevant information while discarding less important details.

The network employs an activation function known as the Rectified Linear Unit (ReLU) to further improve feature extraction. This operation makes the model non-linear, which improves its ability to spot complex patterns in the data. The ReLU activation layer helps the model learn to generate accurate sentiment predictions by making underlying tweet properties more visible. Lastly, the results from the convolution and pooling layers are fed into a Softmax layer, which generates a probability distribution across the sentiment categories (positive or negative). The Softmax layer determines the final prediction by choosing the class with the highest probability.

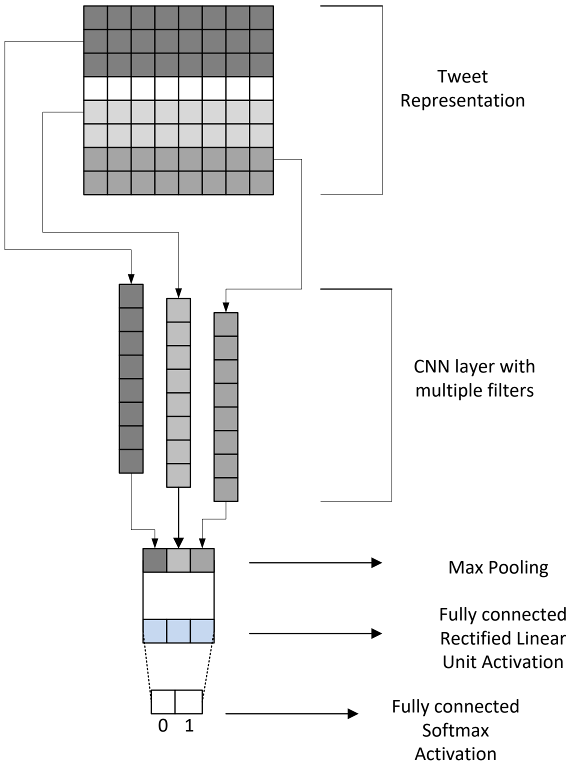

Figure 1: CNN architecture for tweet classification

Sentiment classification, more especially, tweet classification makes use of the CNN architecture shown in the diagram. By means of the extraction and processing of textual elements, every architectural component is meant to progressively transform the unprocessed input—a tweet—into a meaningful sentiment classification (positive or negative).

Showed at the top of the diagram, the tweet representation shows how the input tweet is converted into a numerical form the CNN can grasp. Every tweet is seen as a matrix with word embedding dimensions by each column and words expressed by each row. These embeddings quite faithfully capture the semantic meaning of the words. The size of the matrix is determined by the dimensionality of the word embeddings—typically d, which may be 100, 200, or more, depending on the embedding technique used, such as GloVe or Word2Vec—and the length l of the tweet. This matrix comes from CNN.

The next phase in the design is the CNN layers with multiple filters, which are required for the convolutional operation. These convolution filters search for local patterns in the word embeddings—that is, relevant phrases or n-grams—as they move throughout the Twitter matrix. Every filter generates several feature maps, which are meant to extract a certain type of feature from the tweet. These feature maps highlight important tweet elements necessary for sentiment analysis. Using many filters ranging from small meaningful patterns (like word pairings) to larger semantic structures (like sentiment-loaded sentences), the CNN can capture many levels of abstraction in the tweet.

After the convolutional layers, a max pooling layer is applied to decrease the dimensionality of the feature maps. Max pooling achieves this by choosing the most significant feature from small segments of the data, effectively reducing the size of the feature maps while retaining only the most critical values from each region. This technique reduces the computational load on the next layers and guarantees that the most crucial features of the tweet remain for further processing.

The collected and aggregated salient features of the tweet find application in the ReLU function in a complete connected layer. Every neuron is interconnected to every other, hence the fully connected layer acts as a dense layer. Combining the data discovered by the previous pooling and convolutional layers helps the network to find complex, non-linear relationships between the features. Popular activation function ReLU gives the model non-linearity—something the network needs to learn complex patterns and differentiate between positive and negative emotions.

Fully linked layer output passes through softmax activation layer. This function creates a probability distribution for positive and negative sentiment. After calculating probabilities, the class with the highest likelihood is predicted. Based on the highest softmax score, tweets are classified as positive or negative in sentiment analysis

IV EXPERIMENTAL ANALYSIS

Python 3.6 was the tool utilized for the tests; machine learning models were created using Keras and Scikit-learn modules. The dataset featured two distinct moods: good and negative. Mood data was examined and classified utilizing a linear SVM. The training value for the SVM model was 1 as the sentiment classes were either yes or no. This made it most suited for this employment.

Word Embedding (WE) transformed tweets into vector models in order to improve the SVM performance. Different approaches were utilized to include words into hybrid embeddings, pre-trained embeddings, and trained embeddings in order to prepare the dataset. These several embedding techniques aimed to identify varying degrees of semantic links in the Twitter data. Apart from the SVM model, the research examined the optimal configuration of an ANN to group the data. Various ANN setups were tested using varying word dictionaries and embeddings in search of the optimum one for sentiment analysis. Table 2 shows six of the ten ANN configurations tried having good performance. With SVM and ANN both showing promise in many contexts, this blend of word embeddings and machine learning models helped us identify the best approaches to categorize mood.

Table 2: Parameters used in the experiments

Configuration	Batch Size	Size of the Hidden layers	No. of Neurons in Layer1	No. of Neurons in Layer2
Config-1	256	1	16	-
Config-2	256	1	32	-
Config-3	128	1	64	-
Config-4	256	2	16	4
Config-5	64	2	32	8
Config-6	128	2	64	16

In order to find the best configuration for sentiment classification, the tests used six different ANN configurations, as shown in Table 2. The batch size, hidden layer count, and neuron per layer vary among configurations. A model's batch size determines how many training data are handled prior to updating the model's internal parameters. Batch sizes varied between 64 and 256 in the combinations. Large batch sizes (256, for example) usually result in more consistent updates but may take more time to converge, whereas smaller batch sizes (64, for example) might lead to more frequent updates and possibly better convergence. The hidden-layer size parameter determines how many hidden layers make up the neural network. The number of hidden layers in a network might be anything from one to two; adding more layers allows it to pick up on more complicated patterns in the input. Config-4, Config-5, and Config-6 are examples of multi-hidden-layer models that can model more complicated relationships in the input data.

The number of neurons in both Layer 1 and Layer 2 is a crucial element affecting the model's learning capability. Configurations with a single hidden layer (Config-1, Config-2, and Config-3) have between 16 and 64 neurons. Configurations with two hidden layers (Config-4, Config-5, and Config-6) have more complex structures, with fewer neurons in the second layer (Layer 2) compared to the first. For example, in Config-6, the first hidden layer contains 64 neurons, whereas the second hidden layer consists of 16 neurons.

Figure 2 shows across several setups the classification accuracy attained with the GloVe word embedding approach. Especially when utilizing word embeddings derived from the Twitter dataset, the trial results consistently demonstrate that Config-4 outperformed other configurations in terms of accuracy.

Two hidden levels comprise the Config-4 architecture. There are sixteen neurons in the 1st layer and sixteen in the 2nd layer, therefore offering a balanced and effective framework for learning. More successfully than the other configurations evaluated, this configuration effectively captured the core sentiment patterns in the tweet data. Additionally, Config-4 outperformed the SVM method, especially in the context of back propagation. Back propagation minimized the error between anticipated and real sentiment labels therefore enabling the neural network to iteratively change its weights. Over time, this produced rising categorization accuracy.

The success of Config-4 emphasizes the need of both network architecture and high-quality word representations in sentiment classification tasks especially when trained on pre-trained GloVe embeddings. Pre-trained embeddings help the model to employ past knowledge about word semantics, hence improving its capacity to generalize across several sets of tweets and hence increase the total accuracy.

As Config-4 shows, this mix of GloVe embeddings and a well-tuned ANN architecture presents a strong method for tweet sentiment analysis that beats conventional models like SVM.

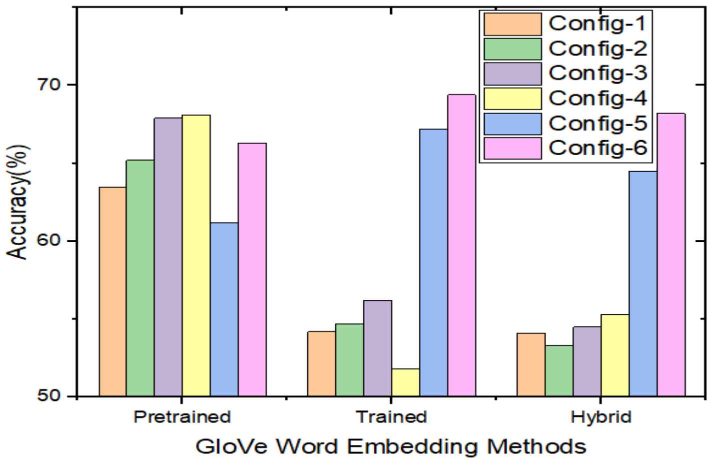

Figure 2: Accuracy comparison of the different methods

Figure 3 displays the results of six n-gram models using twitter data used to assess Config-4 and SVM using different word embedding formats. Each n-gram model encapsulates text word sequences differently, where "n" represents the total word count. The 3-gram model—where n = 3—showed the highest performance for both the SVM and Config-4 architectures according to the experiment results. SVM specifically obtained an accuracy of 74.5% using 3-gram representations; Config-4 exceeded SVM with an accuracy of 78.1%.

Because the 3-gram model can capture richer context than lower-order n-grams (like unigrams or bigrams), it exhibits better performance. Particularly in sentiment analysis where phrases often contain more sentiment information than individual words, the model can better grasp the links between nearby words by evaluating sequences of three words. This outcome emphasizes how well n-gram characteristics combined with word embedding representations work for both deep learning architectures like Config-4 and conventional machine learning models like SVM. By enabling the models to more efficiently employ contextual information, 3-grams help them to improve sentiment analysis task classification performance. Especially when enhanced with advanced word embeddings and multi-word sequences like n-grams, the better performance of Config-4 over SVM again emphasizes the value of deep learning models in capturing complicated patterns inside the data.

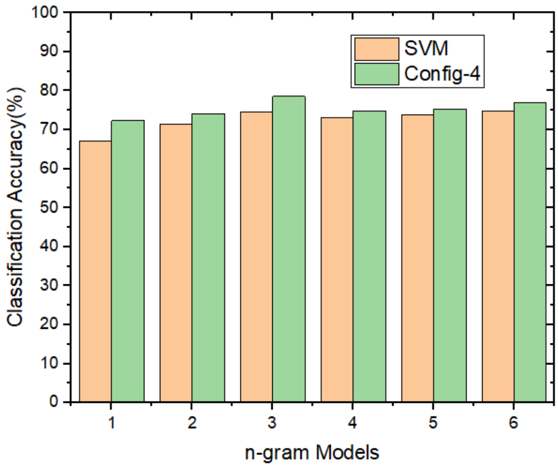

Figure 3: Plot of Classification Accuracy of the SVM and Config-4

V CONCLUSION

This study concentrated on using Twitter data on the Amazon Corporation to, based on consumer tweets, identify major rivals and industry leaders. Pre-processing, feature extraction using n-grams, and word embeddings created from the GloVe lexicon were among the advanced approaches for data extraction and analysis that the suggested method combined. Using pre-trained word embedding datasets, several ANN configurations were trained and compared following the transformation of the tweets into feature vectors alongside the SVM classifier. Outperformance of the Config-4 architecture among the tested models exceeded that of the other ANN designs and the SVM model. Particularly the combination of GloVe embeddings and the multi-layer structure of Config-4 for sentiment analysis and data classification tasks, this emphasises the efficiency of the hybrid strategy applied in the research.

REFERENCES

[1] Number of smart phone users: https ://www.stati sta.com/stati stics /33069 5/numbe r-of-smart phone -users – world wide/. Accessed 06 Sep 2022.

[2] Flight is best mode of travel: https ://www.fligh tnetw ork.com/blog/10-reaso ns-flyin g-still -best-way-trave l/.s Accessed 06 Sep 2022.

[3] Twitter https ://en.wikip edia.org/wiki/Twitt er. Accessed 23 Sep 2022.

[4] Abel F, Gao Q, Houben G-J, Tao K (2013) Twitter-based user modeling for news recommendations. In: Rossi F (ed) IJCAI 2013, proceedings of the 23rd international joint conference on artificial intelligence, Beijing, China, August 3–9, 2013.IJCAI/AAAI.

[5] T. U. Haque, N. N. Saber and F. M. Shah, "Sentiment analysis on large scale Amazon product reviews," 2018 IEEE International Conference on Innovative Research and Development (ICIRD), 2018, pp. 1-6, doi: 10.1109/ICIRD.2018.8376299.

[6] Onan, Aytuğ. "Mining opinions from instructor evaluation reviews: a deep learning approach." Computer Applications in Engineering Education 28, no. 1 (2020): 117-138.

[7] Zomaya AY, Sakr S. Handbook of big data technologies. Berlin: Springer; 2017. https ://doi.org/10.1007/978-3-319-49340-4.

[8] Wilson T, Hoffmann P, Somasundaran S, Kessler J, Wiebe J, Choi Y, Cardie C, Riloff E, Patwardhan S. OpinionFinder. In: Proceedings of HLT/EMNLP on interactive demonstrations, association for computational linguistics. 2005. p. 34–5.

[9] Kontopoulos E, Berberidis C, Dergiades T, Bassiliades N. Ontology-based sentiment analysis of twitter posts. Expert Syst Appl. 2013;40(10):4065–74.

[10] Alantari, H.J.; Currim, I.S.; Deng, Y.; Singh, S. An empirical comparison of machine learning methods for text-based sentiment analysis of online consumer reviews. Int. J. Res. Mark. 2021, 39, 1–19.

[11] Yadav, V.; Verma, P.; Katiyar, V. E-commerce product reviews using aspect based Hindi sentiment analysis. In Proceedings of the 2021 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 27–29 January 2021; pp. 1–8.

[12] Nadia F. F., da Silva, Eduardo R. Hruschka, Estevam R. Hruschka Jr., “Tweet sentiment analysis with classifierensembles,” Decision Support Systems, vol. 66, pp. 170-179, 2014.

[13] Walaa Medhat, Ahmed Hassan, Hoda Korashy “Sentiment analysis algorithms and applications: A survey,” AinShams Engineering Journal, vol. 5, no. 4, pp. 1093-1113, 2014.

[14] L. K. Ramasamy, S. Kadry, S. Lim, “Selection of optimal hyper-parameter values of support vector machine forsentiment analysis tasks using nature-inspired optimization methods,” Bulletin of Electrical Engineering andInformatics (BEEI), vol. 10, no. 1, pp. 290-298, 2020.

[15] Nurulhuda Zainuddin, Ali Selamat, Roliana Ibrahim, “Twitter Feature Selection and Classification Using SupportVector Machine for Aspect-Based Sentiment Analysis,” International Conference on Industrial, Engineering andOther Applications of Applied Intelligent Systems, vol. 9799, pp. 269-279, 2016.

[16] Abinash Tripathy, Ankit Agrawal, Santanu, Kumar Rath, “Classification of Sentimental Reviews Using MachineLearning Techniques,” Procedia Computer Science, vol. 57, pp. 821-829, 2015

[17] Tan, Kian Long, Chin Poo Lee, and Kian Ming Lim. "A survey of sentiment analysis: Approaches, datasets, and future research." *Applied Sciences* 13.7 (2023): 4550.

[18] Rodríguez-Ibánez, Margarita, et al. "A review on sentiment analysis from social media platforms." *Expert Systems with Applications* 223 (2023): 119862.

[19] Cui, Jingfeng, et al. "Survey on sentiment analysis: evolution of research methods and topics." *Artificial Intelligence Review* 56.8 (2023): 8469-8510.

[20] Zhu, Linan, et al. "Multimodal sentiment analysis based on fusion methods: A survey." *Information Fusion* 95 (2023): 306-325.