Proceedings of the 4th International Conference on Computing Innovation and Applied Physics
DOI: 10.54254/2753-8818/2025.22733

Sentiment Analysis Applied on Tweets

Chen Yuan[1*], Xianglong Wang[2], Xingda Jiang[3], Jiale Wang[4], Boxuan Huang[5]

[1]*Department of Mathematics, University of Connecticut, Storrs, USA*
[2]*Management and Business Department, Skidmore College, Saratoga Springs, USA*
[3]*Faculty of Applied Science and Engineering, University of Toronto, Toronto, Canada*
[4]*Arts and Science, University of Toronto, Mississauga, Canada*
[5]*Wuhan No.11 Middle School, Wuhan, China*
**Corresponding Author. Email: chen.yuan@uconn.edu*

Abstract: As digitization continues to expand nowadays, the accurate capture and comprehension of public sentiment on social media has become vital for various stakeholders such as government, businesses, and researchers. In this paper, we aim to perform sentiment analysis on Tweets and explore effective methods to classify sentiment into joy, sadness, anger, and fear from textual content. We utilized datasets from Kaggle containing textual tweets and employed models such as RNN, LSTM, Transformer, and SVM to compare their performance in sentiment analysis. The findings of this paper may serve as a valuable reference for the selection of models in sentiment analysis, particularly when working with medium-sized datasets. Additionally, they may offer guidance on selecting universally applicable model choices for conducting sentiment analysis across social media platforms.

Keywords: Sentiment Analysis, social media, Kaggle.

1. Introduction

In the era of digital communication, where ideas and thoughts are openly exchanged across numerous platforms such as Twitter, Reddit, and Facebook. The ability to accurately collect and comprehend public emotions has become crucial. Sentiment analysis, also known as opinion mining, has become an important area of study within Natural Language Processing (NLP), attracting the interest of both academics and business sectors with its wealth of useful applications. Some of the applications include analyzing customer satisfaction with goods and services, public sentiment on political events, and predicting financial market trends. The information uncovered by sentiment analysis has significant ramifications for social studies, business strategies, and governmental regulations.

Twitter, among various platforms, stands out as a focal point for sentiment analysis due to its immediacy, diversity, and volume. Twitter's huge user base, with millions of users actively tweeting (making a post on Twitter), commenting, and interacting with other users every day, touching upon everything from consumer goods to international political views, provides an unfiltered and real-time glimpse into the public sentiment on a broad spectrum of topics. In consequence, many studies have utilized Twitter to perform sentiment analysis. Classifying sentiment into positive, negative, and neutral based on the given text is one of the methods most frequently used in those studies. However, our study aims to develop a more precise and nuanced understanding of these emotions expressed on Twitter by classifying them into four dimensions: joy, sadness, anger, and fear. These emotions play

DOI: 10.54254/2753-8818/2025.22733

significant roles in various contexts such as consumer behavior, political engagement, and social interactions. We used four datasets from Kaggle containing Twitter content, which includes numerous tweets, along with the sentiment corresponding to each of them. Advanced NLP techniques and deep learning algorithms such as RNN, LSTM, Transformer, and SVM were used in the analysis.

Generally, in our study, by exploring effective methods to analyze sentiment and emotions on Twitter, we hope to enhance our understanding of public sentiment on social media and facilitate more informed and responsive decision-making across a multitude of sectors, offering valuable insights for researchers, businesses, policymakers, and the broader community.

2. Literature review

With the increasing digitalization nowadays, sentiment analysis and opinion mining on social media platforms have drawn substantial academic and practical interest. Over the previous years, many studies have shown that it can provide valuable insights into various aspects such as customer satisfaction with goods and services [1,2,3], public opinions on specific topics like NFT, and trends in the financial market [4,5]. Notably, Twitter, a popular social media platform and microblogging service with almost 238 million daily active users [6], has been suggested to be a reliable source for sentiment analysis of public opinions including emotional dynamics on social issues, electoral prediction, and the stock market [7,8,9,10].

From a methodological perspective, one of many techniques that have been employed to conduct sentiment analysis is the Support Vector Machine (SVM) classifier. It is a supervised machine-learning algorithm that can classify data by utilizing predetermined features [11]. For instance, Pang et al. [12] and Mullen & Collier. [13] have conducted studies utilizing SVM to categorize the polarity of movie reviews into positive, negative, or neutral. Additionally, SVM was also used in other applications, one of which is to summarize opinions expressed on microblogs [14].

On the other hand, the emergence of deep learning techniques also has a significant impact on the research direction in sentiment analysis. Recurrent Neural Network (RNN), compared to SVM, has the unique ability to process sequential data and temporal dependencies in text [15]. This capability enables it to forecast future trends by utilizing past and current data. Therefore, RNN has been utilized to tackle challenging real-world scenarios such as predicting financial trends [16] and gesture recognition [17]. Comprehensive investigations have been carried out regarding the theoretical underpinnings, design, and practical applications of recurrent neural networks, as documented in the work by Haykin et al.[18], Kolen and Kremer [19], and Medsker and Jain [20]. However, traditional RNNs face challenges when processing longer sequences due to the problem known as "vanishing gradients," leading to difficulties in learning long-term dependencies in the data [21].

To address this, a variety of RNN variants have been proposed, including Long Short-Term Memory (LSTM), which was designed to better remember information from earlier in the sequence by using a unique mechanism known as "gates" while maintaining the ability to process sequential data like raw text and capture temporal dependencies in texts [22]. Such ability allows it to comprehend the context and relationships between words in a sentence [23,24]. A recent study by Mahadevaswamy & Swathi [25] suggested their LSTM model achieved an amazing accuracy of 91.4% in classifying users' attitudes from product reviews on Amazon into positive and negative. On top of that, LSTM is also promisingly capable of identifying emotional tones such as happy, sad, and angry from the text. Islam et al. [26] conducted a study using LSTM to classify text from Twitter into four emotions (Happy, Sad, Angry, and Love), and the result showed that their method has outperformed other existing models such as Naive Bayes and the convolutional neural network (CNN) with an accuracy of 88.5%.

What's more, the Transformer provides the researchers with another valuable tool in sentiment analysis. Introduced by Vaswani et al. [27], the Transformer was designed to address the limitation

DOI: 10.54254/2753-8818/2025.22733

of the traditional RNN and LSTM models in processing long-range dependencies in sequential data. Its attention mechanism allows the model to selectively focus on different parts of the input sequence when processing each element [27]. It has demonstrated state-of-art performance in various NLP tasks including sentiment analysis and detecting hate speech on social media. For instance, [28] explored the use of Transformer-based models for hateful speech and showed that their models achieved significant improvement compared to models like 1-dimensional convolutional neural network (1D-CNN) and long-term memory (LSTM). Furthermore, the Transformer has been extended and adapted for specific sentiment analysis tasks. Bidirectional Encoder Representations from Transformers (BERT), introduced by Devlin et al.[29], is a pre-trained transformer-based language model that has been fine-tuned for sentiment analysis tasks. More specifically, it's shown that BERT can also effectively identify sentiments associated with specific aspects of a text [30].

The combination of these studies underscores the importance of user-generated content, particularly tweets, as a resource for sentiment analysis. Utilizing techniques including SVM, LSTM, RNN, and Transformer makes it possible to understand and analyze public sentiment more effectively. The results of these analyses can subsequently be applied in various sectors, from shaping political strategies to driving business decisions. By understanding these factors that significantly influence public opinion, we can anticipate more effective communication strategies and better public engagement.

3. Data

In our study, we pick and combine 4 datasets of tweets from Kaggle. Kaggle is one of the largest platforms that offer a wide array of open datasets. Therefore, we decided to utilize Kaggle to collect relevant datasets for our research paper and subsequently integrate them. We opted for Twitter as the platform for sentiment analysis due to its status as one of the world's largest social media platforms. Furthermore, tweets often contain evident or strong emotional expressions. Hence, we believe that choosing tweets for sentiment analysis serves as an ideal data source.

The integrated dataset consists of 16,747 training samples and 14,221 testing samples, each representing a short text tweet. On average, the word count of each tweet ranges from 16 to 18 words, as illustrated in the accompanying Tables 1 and 2.

Table 1: Training data statistics

Dimension	Anger	Joy	Fear	Sadness	Total
Train Average Word Comt	18.2	18.7	17.4	18.1	18.1
Train Raw Number	4026	4184	4451	4085	16747
Train %	24.0%	25.0%	24.4%	26.6%	100%

Table 2: Testing dataset statistics

Dimension	Anger	Joy	Fear	Sadness	Total
Train Average Word Comt	16.4	16.3	16.3	16.9	16.5
Train Raw Number	3580	3575	3531	3534	14221
Train %	25.2%	24.8%	25.1%	24.9%	100%

Apart from the above, while organizing the data, we didn't merely divide it into positive and negative sentiments. Instead, we classified it into four dimensions, representing distinct emotions: "Anger," "Joy," "Fear," and "Sadness." We took care to distribute the data evenly among these four dimensions. By introducing additional dimensions to sentiment analysis, we aim to compare the performance of RNN models, LSTM models, SVM models, and Transformer models.

DOI: 10.54254/2753-8818/2025.22733

In addition, we took a different approach compared to other studies, which often use imbalanced training and testing data ratios like 8:2 or 7:3. Instead, we intentionally balanced the dataset by setting a 1:1 ratio for training and testing data (Figure 1). We have slightly increased the amount of training data compared to testing data to facilitate the model's learning process and enhance its feature extraction capabilities. At the same time, this adjustment ensures that the generalizability of these models remains unaffected during testing. This choice allows us to evaluate the overall performance and generalizability of the four models in sentiment analysis.

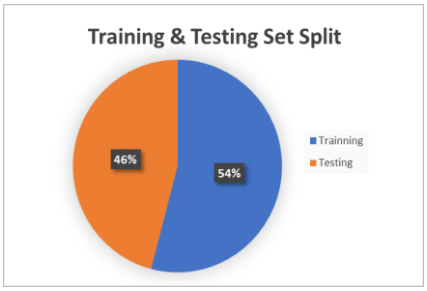

Figure 1: Training and testing set ratio

4. Methodology

Our research methodology has 5 steps. One step of preprocessing data and four steps of building different models and testing with our datasets.

4.1. Data Preprocessing

In the data preprocessing part, we did two procedures to clean the data. Figure 2 is the research flowchart.

1. We Removed all the spaces, names, emojis, and websites in our dataset. Websites like forwarded links and advertisements are not useful in sentiment analysis. For the same reason, names, spaces, and emojis were also removed.

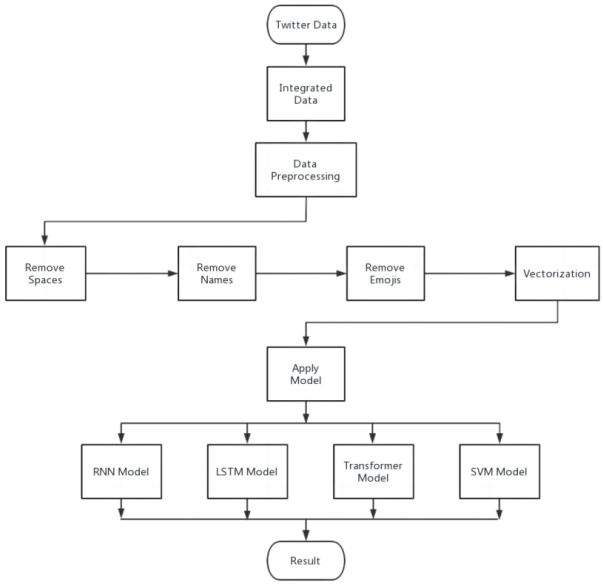

Figure 2: Research flowchart

DOI: 10.54254/2753-8818/2025.22733

2. After removing all the useless terms in the dataset, we performed the vectorization of our data. This step is crucial as it allows computers to understand and process text data by converting it into numerical form, a process known as text vectorization. By vectorizing text data, we obtain a numerical representation, enabling the use of statistical and machine learning methods for sentiment classification and analysis. Additionally, data vectorization facilitates the extraction of useful features from the text data, such as word frequency and word embeddings, which provide vital information to enhance the sentiment analysis model's understanding of the sentiment tendencies or categories in the text. Most importantly, through text vectorization, we seamlessly integrate text data with other data types and apply various powerful algorithms for conducting sentiment analysis.

4.2. RNN

Recurrent Neural Networks, known as RNN models, are a type of neural network designed to work with sequential data, such as time series or natural language. The key idea behind RNNs is the introduction of recurrent connections, which allow information to be persistently passed from one time step to another within the network [31].

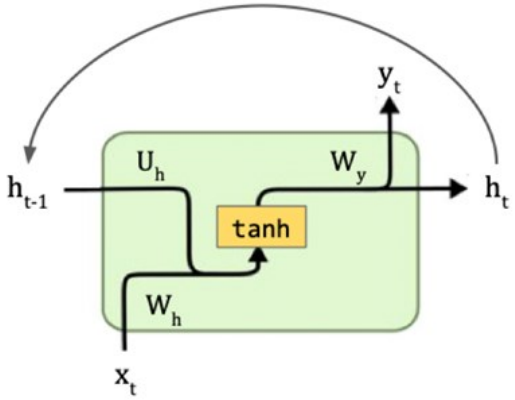

Figure 3: One RNN layer

Unlike traditional feedforward neural networks, which process data in a one-way manner, RNNs have an internal loop that enables them to maintain hidden states or memory of previous inputs as they process each element in a sequence [32]. In Figure 3, it shows one layer of RNN, if we unroll this, we can see multiple layers, and each layer is responsible for a token. The fundamental building block of an RNN is the RNN cell. It takes an input at each time step, processes it along with the hidden state from the previous time step, and produces an output and a new hidden state. The output at each time step can be used for prediction or fed back into the RNN for the next time step. This temporal processing capability allows RNNs to recognize patterns and dependencies across time, making them well-suited for tasks involving sequential data.

To use the RNN model for sentiment analysis of Twitter text. We use the gradient descent method to implement the RNN model. With gradient descent, we hope to find a local minimum [33]. Unfortunately, during the process of dealing with gradients, one bad thing about RNN is that it has faced challenges with long-range dependencies due to the vanishing gradient problem[31]. As the network processes data over many time steps, the gradients may become extremely small, leading to difficulties in learning and capturing long-term dependencies. The model may also suffer from gradient explosion or gradient vanishing, which makes it harder to learn from inputs.

DOI: 10.54254/2753-8818/2025.22733

4.3. LSTM

LSTM is considered one of the most useful models in performing sentiment analysis (Figure 4). Compared to the RNN model, the most notable difference in the LSTM model is that the output from the previous step influences three locations in the next step. [34] Each action is controlled by a sigmoid unit, which determines whether to suppress or enhance. The three cylindrical units on the far right in the diagram are the core of the LSTM, namely the input gate, the forget gate, and the output gate. Firstly, the forget gate acts on the position of the linear self-loop, which is the location of the black square on the right side of the path. This self-loop partially addresses the issue of long-term dependencies.

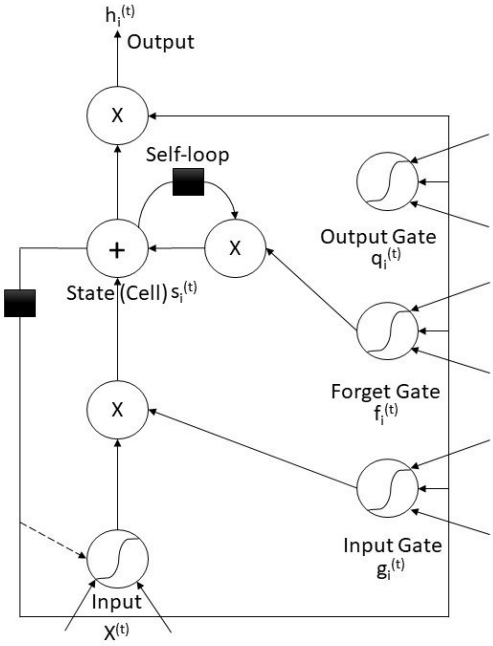

Figure 4: LSTM model illustration

The forward propagation function of the forget gate is represented by the following equation.

$$f_i^{(t)} = \sigma(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{i-1})$$

"$x^{(t)}$" represents the current output, "h" denotes the current hidden layer vector. "b^f", "U^f", and "W^f" are the bias, input weights, and forget gate loop weights, respectively. The internal state update within the LSTM unit is shown in the following.

$$s_i^{(t)} = f_i^{(t)} S_I^{(t-1)} + g_i^{(t)} \sigma(b_i + \sum_j U_{i,j} x_j^t + \sum_j W_{i,j} h_j^{(t-1)})$$

The LSTM cell is where the "State" is in figure 4. Follow the cell down until you get to the bottom input gate on the right-hand side, whose formula is very similar to the forget gate.

$$g_i^{(t)} = \sigma(b_i^f + \sum_j U_{i,J}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{i-1})$$

DOI: 10.54254/2753-8818/2025.22733

After input State, the outputs are multiplied by the inverse hyperbolic function to obtain the final output. The outputs are the top gate on the right-hand side, and this function is like the forget gate, which is the function below.

$$q_i^{(t)} = \sigma(b_i^o + \sum_j U_{i,J}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{i-1})$$

The last function is the formula for the final output, and this concludes the LSTM forward propagation.

$$h_i^{(t)} = \tanh(s_i^{(t)}) q_i^{(t)}$$

4.4. Transformer

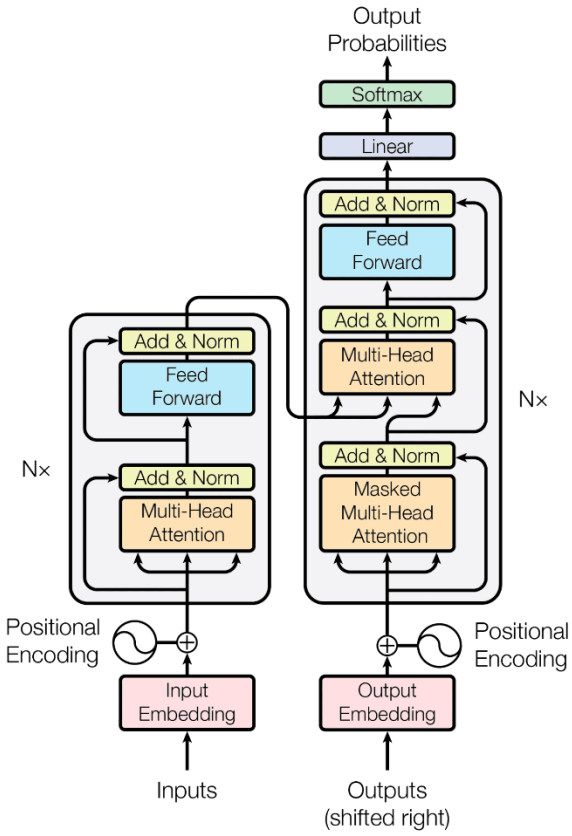

Figure 5: Transformer model illustration

As shown in Figure 5, the Transformer model [27] marked a paradigm shift in NLP research, replacing recurrent neural networks (RNNs) as the primary sequence-to-sequence model. The traditional sequential models were constrained by the sequential nature of processing, making it difficult to handle long-range dependencies effectively. The Transformer addressed this limitation by employing self-attention mechanisms, enabling parallel processing, and capturing global contextual information.

The core of the Transformer is the self-attention mechanism, which allows the model to weigh the importance of different words in a sentence concerning each other. It computes the attention scores between all pairs of words in the input sequence and derives the attention weights for each word based on its relevance to other words. This attention mechanism empowers the model to focus on the most

DOI: 10.54254/2753-8818/2025.22733

informative tokens, allowing it to capture complex linguistic patterns and dependencies across the entire input sequence.

To enhance the modeling capability further, the Transformer employs multi-head attention. It involves multiple parallel self-attention layers, each responsible for capturing different types of relationships within the input sequence. By attending to various semantic aspects, multi-head attention enables the model to effectively process and encode diverse information, contributing to better representation learning.

The Transformer model comprises two main components: the encoder and the decoder. The encoder receives the input sequence and generates a series of hidden representations that preserve essential information. The decoder, on the other hand, uses the encoder's outputs and self-attention mechanisms to generate the output sequence. The encoder-decoder architecture allows the model to learn the relationships between the input and output sequences effectively, making it highly suitable for machine translation tasks.

Since the Transformer lacks the inherent sequential order present in RNNs, it requires a mechanism to preserve positional information. To achieve this, the Transformer uses positional encodings, which are added to the input embeddings. These positional encodings convey the relative positions of words in the input sequence, enabling the model to distinguish between tokens based on their positions.

The Transformer model is trained using supervised learning on large-scale datasets. During training, it minimizes the cross-entropy loss between predicted outputs and ground-truth labels. To optimize the model's parameters, advanced optimization techniques such as Adaptive Moment Estimation [35] are commonly employed, facilitating faster convergence and better generalization.

The transformer model we used is BERT (Bidirectional Encoder Representations from Transformers) [29], a groundbreaking pre-trained language representation model.

The key innovation of BERT lies in its pre-training objectives. It employs two unsupervised tasks: masked language modeling (MLM) [29] and next sentence prediction (NSP) [29]. In MLM, a certain percentage of tokens in an input sentence are randomly masked, and the model is trained to predict the original masked tokens based on the context. This task encourages the model to understand the bidirectional relationships between words, facilitating deeper contextual understanding. In NSP, BERT is trained to determine whether two input sentences appear consecutively in a text corpus, aiding the model in capturing sentence-level relationships.

BERT adopts the Transformer architecture as its backbone, consisting of multiple stacked encoder layers. During pre-training, BERT utilizes a massive amount of raw text data to learn representations that capture the intricacies of the language. The unsupervised pre-training process enables BERT to build a strong language model without any task-specific labels.

After pre-training, BERT is fine-tuned on downstream NLP tasks using supervised learning with task-specific datasets. The model is further refined using task-specific objectives and labeled data while leveraging the contextual embeddings learned during pre-training. This transfer learning approach allows BERT to adapt to various NLP tasks effectively, yielding state-of-the-art results with minimal task-specific training data.

4.5. SVM

Support Vector Machine, known as SVM, is an algorithm used for pattern recognition[36]. SVM is a machine learning classification technique that uses functions called kernels to map a space of data points into a new space where the data is not linearly separable, it allows for false classifications [13].

SVM is implemented mainly through classification. By classifying them, he placed different patterns on different points, assuming they were on a large flat surface. The goal of SVM is to maximize the distance of the large plane to each point, which means that each region has the most

DOI: 10.54254/2753-8818/2025.22733

special eigenvalue. This allows the SVM to process a given input, and the greater the distance, the more accurate the classification [37].

Since our goal is to perform sentiment analysis on Tweets, the SVM model fits our requirements well. SVM involves extracting eigenvalues from the training text and using them to analyze emotions in the test text. To achieve our goal, Initially, we preprocess the training text by obtaining each word from each sentence and filtering out common words, among other steps. Through this analysis, we determine the word frequency of each word, which serves as the weight. Words in each sentence are then reorganized based on their word frequency weight. As we all know, in general frequency keywords will express certain emotions more effectively.

$$w_{ik} = \frac{tf_{ik} * log(N/n_k + 0.01)}{\sqrt{\sum_{i=k}[tf_{ik} * log(N/n_k + 0.01)]^2}}$$

The formula includes variables such as "tf" for word occurrences, "N" for the number of sentences, and "nk" for word occurrences across all sentences.

By converting each sentence into a set of numbers and comparing them with the training data, we calculate their similarity to each class. Continuously adjusting parameters based on the training and test data results helps us achieve a relatively accurate classification standard [38]. Finally, we used Score, precision, recall, and accuracy to evaluate the SVM model.

5. Results

To analyze the performance of all models, we use four indicators as comparative standards: Accuracy, F-1 Score, Precision, and Recall.

To calculate these four indicators, four elements are used: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). Because we have divided the data into four dimensions, we calculate each indicator using these four elements in each dimension. Then, the average value of each indicator obtained from the four dimensions serves as the result for these four indicators.

Accuracy serves as a fundamental metric for evaluating the performance of a model. It denotes the proportion of correct predictions out of the entire set of predictions. High accuracy means that the model is capable of correctly classifying the text into their respective sentiment categories.

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}}$$

Precision is a performance metric that quantifies the model's ability to accurately identify samples belonging to a particular class among those samples predicted to be in that class. High precision implies that the model minimizes the possibility of misclassifying samples when assigning them to specific sentiment categories.

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}}$$

Recall is used to measure the ratio between the number of true positive predictions and the total number of true positive and false negative predictions. Recall focuses on the proportion of true positive predictions captured by the model. High recall signifies that the model captures fewer samples belonging to a specific class, resulting in more comprehensive coverage of the target category with fewer omissions.

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}}$$

The f-1 score combines precision and recall, and its values range between 0 and 1. A higher value indicates that the model considers both the classifier's accuracy and coverage simultaneously.

DOI: 10.54254/2753-8818/2025.22733

$$F1 = \frac{2PR}{P+R}$$

As shown in Table 3, our RNN model showed an accuracy of 67.62%, an F1 score of 64.08%, a precision of 61.94%, and a recall of 66.39%. Despite employing techniques like dropout to mitigate the issues of vanishing or exploding gradients, the RNN model still falls short with an accuracy of only 0.6762, which indicates that it is not a suitable choice for sentiment analysis when divided into four dimensions.

Table 3: RNN testing results

Model	Accuracy	F1 Score	Precision	Recall
RNN	67.62%	64.08%	61.94%	66.39%

Since our RNN model did not work out as we expected, we pushed our research into the next stage by performing the LSTM model. As shown in Table 4, our LSTM model achieved an accuracy of 70.98%, a precision of 81.56%, a recall of 71.21%, and an F1 score of 72.07%. Even though the LSTM model's results did not reach our expectations, it still showed better outcomes than the RNN model. The four indicators are around 90% with the training dataset, but the testing results dropped to about 70%. The reason for this result was that the training dataset we used was not big enough. Because testing the model's generalization in sentiment analysis is also one of our research objectives, we do not intend to increase the training data quantity to pursue high accuracy with the LSTM model.

Table 4: LSTM training and testing results

Model	Data Type	Accuracy	F1 Score	Precision	Recall
LSTM	Train	94.27%	88.49%	88.21%	94.27%
	Test	70.98%	72.07%	81.56%	70.98%

Since the LSTM model did not reach our goal, we used the Transformer model. As shown in Table 5, our Transformer model demonstrated an accuracy of 94.87%, an F1 score of 95.00%, a precision of 94.89%, and a recall of 94.88%. The Transformer model offered excellent results, but the only disadvantage is that the Transformer consumed a lot of memory at runtime. We could not run all the testing datasets at one time, so we separately tested our dataset several times. Then we calculated the average results of all results.

Table 5: Transformer testing results

Model	Accuracy	F1 Score	Precision	Recall
Transformer	94.87%	95.00%	94.89%	994.88%

We also built another traditional model that is frequently used in sentiment analysis, the SVM model. As shown in Table 6, our SVM model gave an accuracy of 85.43%, an F1 score of 85.39%, a precision of 86.16%, and a recall of 85.43%.

Table 6: SVM testing results

Model	Accuracy	F1 Score	Precision	Recall
SVM	85.43	85.39%	86.16%	85.43%

DOI: 10.54254/2753-8818/2025.22733

The comparison of the four models' results can be observed in the following Figure 6. We can easily conclude that the Transformer model provides the best results in sentiment analysis of Tweets. In general, LSTM models perform better in sentiment analysis compared to SVM models. However, when facing datasets with more dimensions and a comparable amount of training and testing data, SVM exhibits better generalization.

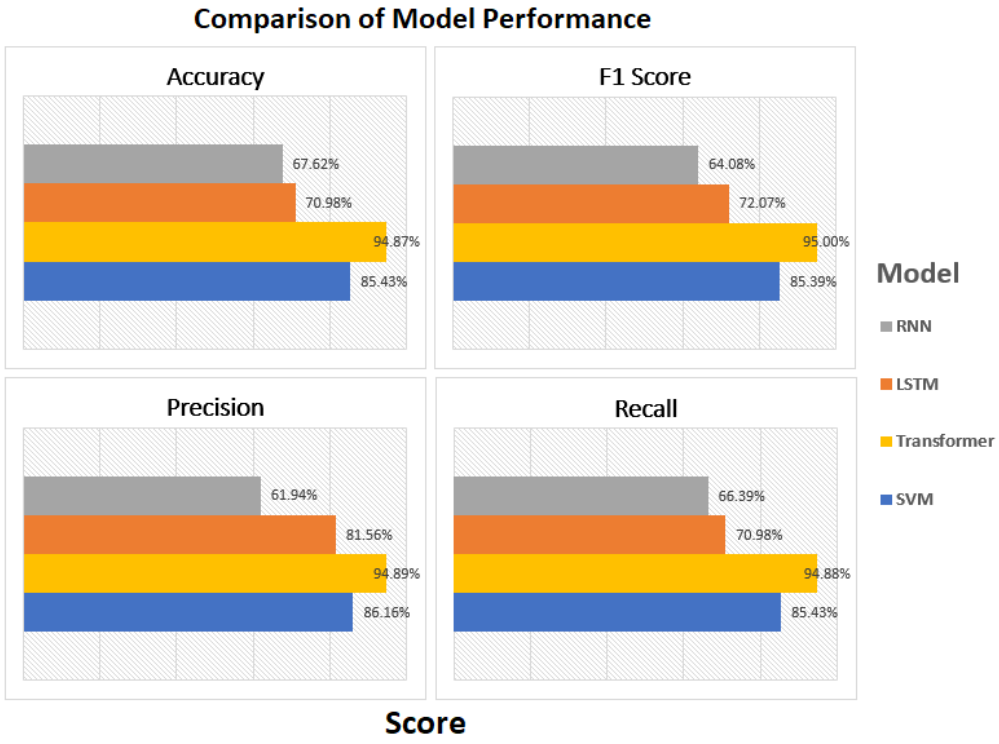

Figure 6: Model comparison

6. Conclusion

In our research, we explored sentiment analysis and classification of tweets using four machine learning and deep learning models. We divided the dataset containing pure text tweets into four emotions and performed data preprocessing and model construction.

Through the RNN model, we found that the sentiment analysis results were not satisfactory in the four dimensions. Furthermore, due to the high dimensionality, RNN was susceptible to gradient explosion.

The LSTM model also did not perform well when facing datasets with similar proportions in training and testing. Although it achieved high training accuracy, the final accuracy was only around 70%. Hence, LSTM's generalization in sentiment analysis was limited, especially when the training data was limited.

Surprisingly, the SVM model yielded decent results compared to the previous two models. As a traditional model commonly used in sentiment analysis, SVM demonstrated good generalization and accuracy.

The Transformer model, being a relatively new deep learning model in the context of sentiment analysis, outperformed all the other models. However, it demands substantial computational power. With Transformer, social media platforms can grasp user sentiment types and trends more quickly and accurately. Considering the one-to-one proportion of the training and testing data we selected,

DOI: 10.54254/2753-8818/2025.22733

the Transformer model showed not only higher accuracy but also good generalization. Thus, even when there is insufficient training data available on the platform, the Transformer model can be employed for sentiment analysis.

In summary, our research has thoroughly examined the generalization and accuracy of different sentiment analysis models across multiple dimensions. Through rigorous reasoning and scientific comparisons, we have convincingly established that the Transformer model exhibits superior performance. While the current studies on the application of Transformer models in sentiment analysis may be limited, we remain optimistic that their potential in analyzing sentiment from social media tweets will be more fully realized in future research.

References

[1] *Al-Otaibi, Shaha, et al. "Customer satisfaction measurement using sentiment analysis." International Journal of Advanced Computer Science and Applications 9.2 (2018).*

[2] *Gang, Zhou, and Liao Chenglin. "Dynamic measurement and evaluation of hotel customer satisfaction through sentiment analysis on online reviews." Journal of Organizational and End User Computing (JOEUC) 33.6 (2021): 1-27.*

[3] *Sinha, Raj. "Data Analysis and Sentiment Analysis on Amazon Reviews." International Journal for Research in Applied Science and Engineering Technology, vol. 9, no. 12, 2021, pp. 2200–2206*

[4] *Qian, Cheng, et al. "Understanding public opinions on social media for financial sentiment analysis using AI-based techniques." Information Processing & Management 59.6 (2022): 103098.*

[5] *Houlihan, Patrick, and Germán G. Creamer. "Leveraging social media to predict continuation and reversal in asset prices." Computational Economics 57.2 (2021): 433-453.*

[6] *Dixon, Stacy Jo. "Twitter Global Mdau 2022." Statista, 11 Nov. 2022, www.statista.com/statistics/970920/monetizable-daily-active-twitter-users-worldwide/.*

[7] *Naskar, Debashis, et al. "Emotion dynamics of public opinions on Twitter." ACM Transactions on Information Systems (TOIS) 38.2 (2020): 1-24.*

[8] *Gayo-Avello, Daniel. "A meta-analysis of state-of-the-art electoral prediction from Twitter data." Social Science Computer Review 31.6 (2013): 649-679.*

[9] *Singh, Sanjam, and Amandeep Kaur. "Twitter sentiment analysis for stock prediction." Available at SSRN 4157658 (2022).*

[10] *Pagolu, Venkata Sasank, et al. "Sentiment analysis of Twitter data for predicting stock market movements." 2016 International Conference on signal processing, communication, power, and Embedded systems (SCOPES). IEEE, 2016.*

[11] *Cortes, Corinna and Vladimir Naumovich Vapnik. "Support-Vector Networks." Machine Learning 20 (1995): 273-297.*

[12] *Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up? Sentiment classification using machine learning techniques." arXiv preprint cs/0205070 (2002).*

[13] *Mullen, Tony, and Nigel Collier. "Sentiment analysis using support vector machines with diverse information sources." Proceedings of the 2004 conference on empirical methods in natural language processing. 2004.*

[14] *Wu, Peng, et al. "Social media opinion summarization using emotion cognition and convolutional neural networks." International Journal of Information Management 51 (2020): 101978.*

[15] *Mikolov, Tomas, et al. "Recurrent neural network based language model." Interspeech. Vol. 2. No. 3. 2010.*

[16] *Giles, C. Lee, Steve Lawrence, and Ah Chung Tsoi. "Rule inference for financial prediction using recurrent neural networks." Proceedings of the IEEE/IAFE 1997 Computational Intelligence for Financial Engineering (CIFEr). IEEE, 1997.*

[17] *Murakami, Kouichi, and Hitomi Taguchi. "Gesture recognition using recurrent neural networks." Proceedings of the SIGCHI conference on Human factors in computing systems. 1991.*

[18] *Haykin, Simon. Neural networks and learning machines, 3/E. Pearson Education India, 2009.*

[19] *Kolen, John F., and Stefan C. Kremer, eds. A field guide to dynamical recurrent networks. John Wiley & Sons, 2001.*

[20] *Medsker, Larry, and Lakhmi C. Jain, eds. Recurrent neural networks: design and applications. CRC Press, 1999.*

[21] *Bengio, Yoshua, Patrice Simard, and Paolo Frasconi. "Learning long-term dependencies with gradient descent is difficult." IEEE transactions on neural networks 5.2 (1994): 157-166.*

[22] *Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural Computation 9.8 (1997): 1735-1780.*

DOI: 10.54254/2753-8818/2025.22733

[23] Wang, Yequan, et al. "Attention-based LSTM for aspect-level sentiment classification." Proceedings of the 2016 conference on empirical methods in natural language processing. 2016.
[24] Tai, Kai Sheng, Richard Socher, and Christopher D. Manning. "Improved semantic representations from tree-structured long short-term memory networks." arXiv preprint arXiv:1503.00075 (2015).
[25] Mahadevaswamy, U. B., and P. Swathi. "Sentiment analysis using bidirectional LSTM network." Procedia Computer Science 218 (2023): 45-56.
[26] Islam, Juyana, et al. "Recognition of Emotion from Emoticon with Text in Microblog Using LSTM." Advances in Science, Technology and Engineering Systems Journal 6.3 (2021): 347-354.
[27] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).
[28] Mukherjee, Swapnanil, and Sujit Das. "Application of transformer-based language models to detect hate speech in social media." Journal of Computational and Cognitive Engineering (2022).
[29] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).
[30] Hoang, Mickel, Oskar Alija Bihorac, and Jacobo Rouces. "Aspect-based sentiment analysis using bert." Proceedings of the 22nd Nordic conference on computational linguistics. 2019.
[31] Wang, Jingyang, et al. "NGCU: A new RNN model for time-series data prediction." Big Data Research 27 (2022): 100296.
[32] Zargar, S. "Introduction to sequence learning models: RNN, LSTM, GRU." Department of Mechanical and Aerospace Engineering, North Carolina State University, Raleigh, North Carolina 27606 (2021).
[33] Andrychowicz, Marcin, et al. "Learning to learn by gradient descent by gradient descent." Advances in neural information processing systems 29 (2016).
[34] Ko, Ching-Ru, and Hsien-Tsung Chang. "LSTM-based sentiment analysis for stock price forecast." PeerJ Computer Science 7 (2021): e408.
[35] Kingma, D. P., & Ba, J. (2017, January 30). Adam: A method for stochastic optimization. arXiv.org. https://arxiv.org/abs/1412.6980
[36] Joachims, Thorsten. "Svmlight: Support vector machine." SVM-Light Support Vector Machine http://svmlight.joachims.org/, University of Dortmund 19.4 (1999): 25.
[37] Pradhan, Ashis. "Support vector machine-a survey." International Journal of Emerging Technology and Advanced Engineering 2.8 (2012): 82-85.
[38] Ahmad, Munir, et al. "SVM optimization for sentiment analysis." International Journal of Advanced Computer Science and Applications 9.4 (2018).