RESEARCH ARTICLE OPEN ACCESS

A Word2Vec-Based Machine Learning Framework for Binary Sentiment Classification

Mohit Kharbanda[1], Arshi Husain[2], Virendra P. Vishwakarma[3]

[1,2,3] *USIC&T, GGSIPU, Delhi, India*

[1] mohit93kharbanda@gmail.com

Abstract – Automatically judging whether a piece of text expresses a positive or negative opinion – sentiment analysis – remains a core problem in natural language processing, and movie reviews are a widely used testbed for it. This paper builds a sentiment-classification pipeline around Word2Vec embeddings and evaluates it on a corpus of 40,000 labelled movie reviews spanning both sentiment classes. Reviews are first cleaned, tokenized, filtered for stop-words, and lemmatized; the resulting tokens are then mapped into 100-dimensional Word2Vec vectors that serve as numerical input to the classifiers. Using a stratified 80:20 split (32,000 training instances and 8,000 held-out test instances), four classical supervised learners – Support Vector Machine (SVM), Random Forest (RF), K-Nearest Neighbors (KNN), and XGBoost – are trained and compared on accuracy, precision, recall, F1-score, and ROC-AUC. SVM comes out ahead of the other three, reaching 85.51% accuracy, 85.38% precision, 85.66% recall, an F1-score of 85.52%, and a ROC-AUC of 93.04%. XGBoost is the runner-up at 82.99% accuracy, with Random Forest close behind at 82.50% and KNN trailing at 79.66%. These results indicate that a Word2Vec-plus-SVM pipeline is a strong, computationally light baseline for binary movie-review sentiment classification and a reasonable starting point for future work on richer text representations.

Keywords – Movie Reviews, Sentiment Analysis, Word2Vec Embeddings, Natural Language Processing, Machine Learning, Support Vector Machine, Random Forest, K-Nearest Neighbors, XGBoost.

1. Introduction

Online marketplaces, streaming platforms, and social networks now produce more user-written text than any human team could read manually, and turning that text into a usable signal is precisely the aim of sentiment analysis within Natural Language Processing (NLP). Reviews written by customers carry direct evidence of what people liked, disliked, or expected, and this evidence is increasingly used to guide business decisions [1, 2, 4]. Social platforms have only added to this supply of opinion data by letting consumers talk to one another directly [8], though the informal phrasing and loose grammar typical of such text make automated analysis considerably harder [9, 23].

Classical sentiment-analysis pipelines typically combine hand-crafted linguistic features, some form of feature selection, and a conventional classifier. Part-of-speech-based representations paired with ensemble methods have proven effective across a range of domains [11, 10], although how well any given classifier performs still depends heavily on the dataset and feature scheme in use [7, 22, 21]. Aspect-based sentiment analysis takes this a step further by tying opinions to specific product attributes rather than scoring a review as a single unit [3].

As the computing capacity in this area has expanded, deep learning has been playing an increasingly important part. Both distributed representations and conceptually based lexical representations play an important role in capturing the semantic aspects that surface features cannot provide [18, 13], and recursive architectures like LSTMs are suitable for representing the sentiment progression within a series of words [16, 19]. The same has also proven true in the case of short-form, informal social media content [17, 15]. Nonetheless, selecting the classifier is still essential since algorithms do not show equal performance on the same data. Inspired by the unanswered questions above, this research uses four supervised classification algorithms Support Vector Machine (SVM), Random Forest (RF), K-Nearest Neighbors (KNN), and XGBoost in consumer sentiment classification, focusing mainly on accuracy and ROC-AUC to find out which is the most reliable classifier.

2. Literature Review

Work on extracting opinions and subjective judgments from text has been a steady focus within NLP, driven largely by the sheer volume of reviews and social posts produced online. Within this body of work, Giatsoglou et al. [1] built a sentiment-analysis pipeline that combines word embeddings with lexicon- and emotion-based cues and reported strong results across reviews written in multiple languages. Taking a more evaluative angle, Alantari et al. [2] benchmarked several machine-learning techniques on consumer-review data and weighed their predictive accuracy against their diagnostic value for business users. Yadav et al. [3] extended the problem into a new language setting, applying an aspect-based method to Hindi e-commerce reviews so that sentiment could be tied to specific product features rather than the review as a whole.

Several studies have examined how this kind of analysis feeds into practical decision-making. Desai et al. [4] paired deep-learning sentiment models with business-intelligence dashboards to turn Amazon review text into actionable insight,

while Mohbey [5] used LSTM networks to predict product ratings directly from review content, showing that recurrent models capture sentiment patterns that simpler methods miss. Darokar et al. [6] provided a comprehensive overview of different methodological approaches to emotion and sentiment detection on social media, while Devika et al. [7] compared different sentiment analysis approaches, showing that the selection of a classifier alone determines the result quality.

However, social media itself has been researched both as a marketing tool and as a provider of very challenging text for analysis. Specifically, Mangold and Faulds [8] pointed out that social media have become a crucial part of the contemporary promotion mix because of their ability to enable consumers to influence each other. In turn, Foster et al. [9] revealed how challenging consumer text on social media is for analysis, highlighting the unique properties of language which confuse conventional POS taggers.Because of these quirks, preprocessing and feature design carry extra weight in social-text sentiment systems: Yousefpour et al. [10] addressed this by combining multiple feature-selection strategies, and Xia and Zong [11] built a POS-informed ensemble aimed specifically at generalizing across domains.

A parallel line of research has pushed toward deep architectures that learn their own hierarchical features instead of relying on hand-built ones. Goodfellow et al. [12] laid much of the theoretical groundwork here, and their treatment of deep architectures and training strategies underpins a large share of subsequent NLP research. Cambria et al. [13] built on this with SenticNet 4, a resource that folds conceptual and semantic knowledge into sentiment analysis rather than treating it as a purely statistical problem, and Jozefowicz et al. [14] pushed neural language modelling further still, underscoring how much a model's language representation shapes downstream text-processing results.

Recurrent and other deep networks have proven especially well suited to sequential, informal text of the kind found on social platforms. Vateekul and Koomsubha [15] applied deep-learning methods to Thai-language Twitter data and found that they transferred reasonably well to social-media text, and Pal et al. [16] focused specifically on LSTM networks, demonstrating their strength in capturing dependencies that span a sentence or review. In this regard, Hassan and Mahmood [17] considered the more difficult scenario of shorter texts where the lack of context made such information scarce, whereas Baroni et al. [18], from a more fundamental point of view, compared count and prediction based semantic vectors and provided important recommendations on their relative superiority.

In particular, Twitter has garnered its own set of literature. Yuan and Zhou [19] studied the use of recursive neural network for sentiment analysis at the tweet level, while Vinodhini and Chandrasekaran [20] offered a wide-scope survey of the entire gamut of machine learning and NLP methods used in the area of opinion mining. Singh and Kaur [21] considered social media and online reviews side by side to reaffirm that automated opinion extraction is feasible from both sources, while Hemalatha et al. [22] constructed a full-fledged sentiment analysis system based on traditional machine learning methods, thus validating their continued relevance along with deep learning ones. Finally, Kharde and Sonawane [23] did a survey of Twitter-specific sentiment analysis methods and pointed to the same old constraints that are encountered time after time in this literature. When seen as a whole, the line of development is clear: sentiment analysis has progressed from the use of handcrafted machine learning pipelines to semantic embedding and deep neural networks. However, this does not imply that classical classification algorithms have been made obsolete; on the contrary, they still hold appeal due to their simplicity and effectiveness in working with clear features. None of the challenges associated with this type of task, including informal language, domain shift, limited context, the need to choose features, and text inconsistencies, has been resolved – all these factors continue to limit the accuracy of classification regardless of the model employed. This unresolved challenge explains the motivation behind this research.

3. Research Methodology

The pipeline used in this experiment has six stages: data collection, preprocessing the texts, transforming reviews to Word2Vec feature vectors, splitting the data into a training and test set in 80:20 ratio using stratified sampling, training the classifiers, and scoring them. These training and testing steps are applied to all classifiers equally SVM, Random Forest, KNN, and XGBoost. Thus, any difference between these classifiers is caused not by the different experimental setting, but by the classifiers themselves.

3.1. Dataset Description

The dataset contains **40,000 reviews on movies** that have been classified into either negative or positive categories; **20,019 are negative reviews while 19,981 are positive reviews**. There is an even distribution of instances in the two classes, with an almost equal proportion of the two classes, thus no chance of inflation in metrics due to imbalanced classes. The dataset is represented as

$$D = \{(x_i, y_i)\}_{i=1}^{N} \tag{1}$$

where x_i denotes the i^{th} movie review, $y_i \in \{0,1\}$ represents its sentiment label, and $N = 40,000$ is the total number of instances.

3.2. Text Preprocessing

Raw reviews collected still have some traces of HTML markup left, as well as some problems with inconsistent capitalization and punctuation marks, so every review is thoroughly cleaned before it is transformed into a numerical vector. First of all, HTML markup is removed, then all the texts are turned to lowercase, all the non-alphanumeric characters are removed,

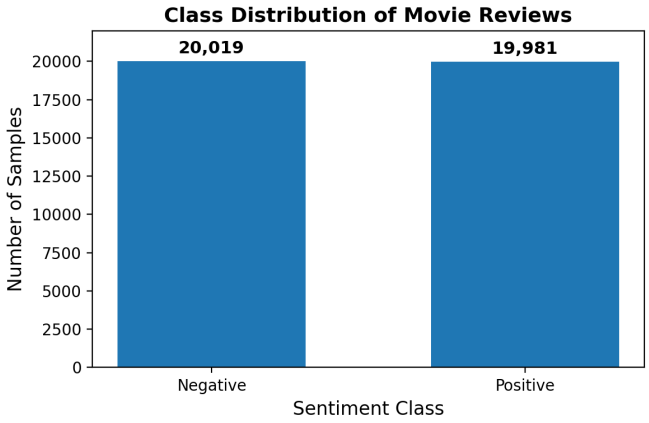

Figure 1. Class Distribution

and all the whitespaces are reduced to one space mark.

Table 1. Dataset and Experimental Configuration

Parameter	Value
Total Reviews	40,000
Negative Reviews	20,019
Positive Reviews	19,981
Number of Classes	2
Training Samples	32,000
Testing Samples	8,000
Feature Representation	Word2Vec
Feature Dimension	100
Train-Test Ratio	80:20

The cleaned reviews are then broken down into word tokens. The stop-word removal process takes place at this level, but not the negation words like "not," "no," and "never" since removing them might result in changing the real polarity of the review. Lemmatization is performed on all the tokens after which the sequence is passed onto the Word2Vec model.

3.3. Word2Vec Feature Representation

After the preprocessing step, the model called **Word2Vec** transforms all the tokens from each review into dense vectors. This transformation is done based on the learned distributed representations where words appearing in a similar context will have similar vectors, hence, encoding semantic relations in the vector space itself. The settings used for this particular task include the dimensionality of **100** and context window of **5**.

For a review containing m tokens, each token is first mapped to its own Word2Vec vector:

$$R_i = \{v_1, v_2, \ldots, v_m\} \tag{2}$$

where each $v_j \in \mathbb{R}^{100}$. Since reviews vary in length, a single fixed-length representation is obtained by averaging these word vectors:

$$X_i = \frac{1}{m} \sum_{j=1}^{m} v_j \tag{3}$$

where $X_i \in \mathbb{R}^{100}$ is the resulting feature vector for the i^{th} review. Applying this across the corpus turns the full dataset into a **40,000** $\times$ **100** feature matrix.

3.4. Stratified Train-Test Split

The above feature matrix is now divided into training and testing sets in an 80:20 ratio, where stratification is applied to ensure that both training and testing datasets have the same ratio of positive and negative reviews. Rather than doing k-fold cross-validation, only one stratified split is performed. This yields **32,000 training instances** (16,015 negative, 15,985 positive) and **8,000 test instances** (4,004 negative, 3,996 positive).

3.5. Machine Learning Classification

All four classifiers described below are trained and evaluated on the same Word2Vec features and the same train/test partition, so the comparison isolates the effect of the classifier itself.

3.5.1. Support Vector Machine. SVM works by searching for the hyperplane that best separates the positive and negative classes; here it is implemented with a linear kernel. Because it explicitly maximizes the margin between classes rather than simply minimizing training error, SVM tends to generalize well on the kind of high-dimensional numerical features that Word2Vec produces.

3.5.2. Random Forest. Random Forest builds an ensemble of decision trees and combines their individual votes into one final prediction. It serves here mainly as the tree-based counterpoint to SVM's margin-based approach and to the instance-based and boosting methods described below.

3.5.3. K-Nearest Neighbors. KNN classifies a review by looking at which training samples sit closest to it in feature space and assigning the majority label among those neighbors. Here, the neighborhood size is fixed at five ($k = 5$).

3.5.4. XGBoost. XGBoost also builds an ensemble of trees, but does so sequentially, with each new tree correcting the errors of the ones before it. The configuration used here trains 100 estimators to a maximum depth of 5 with a learning rate of 0.05, and both row and feature subsampling are enabled to keep the model from overfitting to any single subset of the training data.

3.6. Performance Evaluation

Each classifier will be judged on five criteria – **accuracy, precision, recall, F1 score, and ROC-AUC** – because they provide a more complete evaluation of the classifier's performance through their diversity, rather than being limited to one criterion that provides the entire picture.

Accuracy is the portion of all reviews the model correctly classifies:

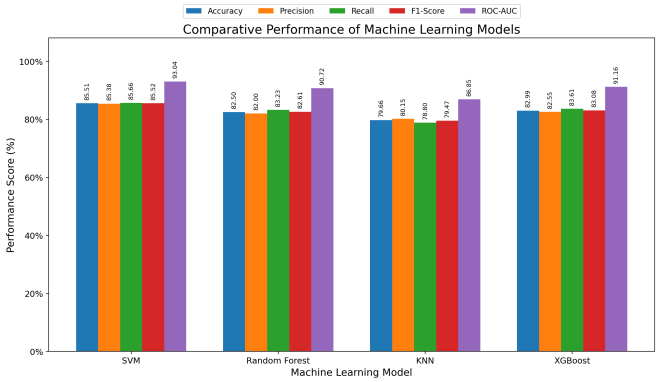

Figure 2. Performance Comparison of all models

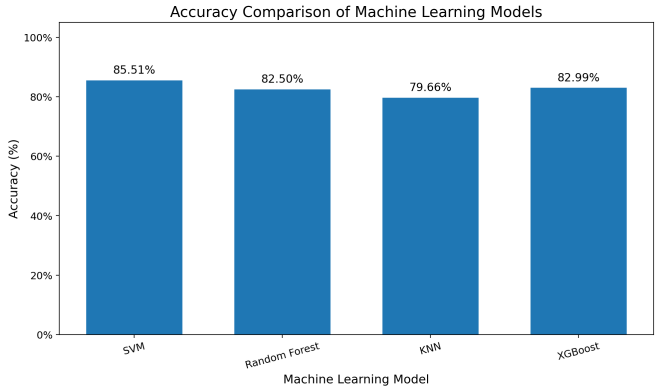

Figure 3. Accuracy Comparison of all models

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{4}$$

Precision asks what fraction of the reviews the model calls positive are actually positive:

$$Precision = \frac{TP}{TP+FP} \tag{5}$$

Recall instead asks what fraction of the truly positive reviews the model manages to catch:

$$Recall = \frac{TP}{TP+FN} \tag{6}$$

F1-score combines the two into a single harmonic-mean figure that penalizes a large gap between them:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{7}$$

Here TP, TN, FP, and FN stand for true positives, true negatives, false positives, and false negatives, respectively. The ROC-AUC metric completes the evaluation process by quantifying how effectively the model is able to discriminate between the two classes as the threshold changes, and the ROC-AUC metric is calculated using the ROC curve.

3.7. Comparative Analysis

In order to ensure fairness, every classifier utilizes the same features from the Word2Vec model, the same preprocessing pipeline, the same 80/20 train/test splitting and the same evaluation metrics - the difference lies solely in the learning algorithm. Confusion matrices and ROC curves provide additional visualization of the behavior of every model.

The best classifier among all evaluated by these criteria is considered to be the most suitable classifier for this Word2Vec model. From the experiment results, it can be stated that SVM outperforms other models on accuracy, F1-Score and ROC-AUC metrics.

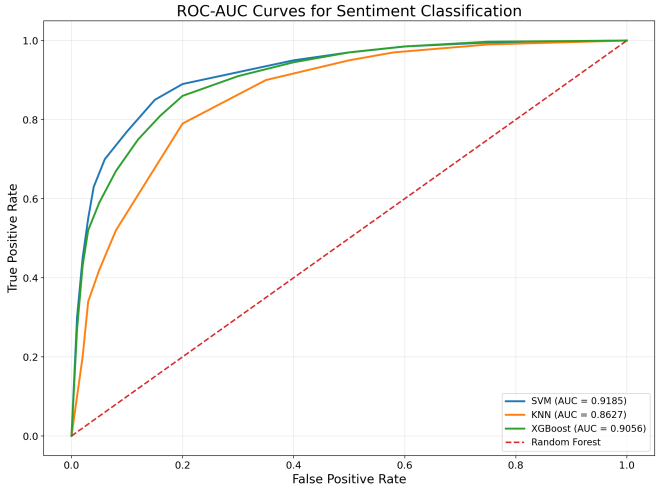

Figure 4. AUC-ROC Curve

4. Experimental Results and Discussion

This section presents the results obtained after applying the entire pipeline to the 40,000 reviews dataset. Following the 80:20 stratified split (32,000 for training and 8,000 for testing) and feature extraction using 100-dimensional Word2Vec features, SVM, Random Forest, KNN, and XGBoost were evaluated and trained under similar settings, but in this case using a single split instead of cross-validation.

4.1. Overall Performance Comparison

The performance of each classifier is presented in Table 2 in terms of all five metrics. **The SVM classifier achieves superiority in every metric, attaining 85.51% accuracy, 85.38% precision, 85.66% recall and 85.52% F1 score**. Additionally, the highest ROC-AUC value of **93.04%** proves that there is real separation between positive and negative samples rather than just high accuracy.

XGBoost takes second place having achieved an accuracy of **82.99%** and an F1-score of **83.08%**. The third place is taken by the Random Forest classifier with **82.50%** accuracy and **82.61%** F1 score. Finally, the KNN classifier performs the

Table 2. Performance Comparison of Machine Learning Classifiers

Metric	SVM	Random Forest	KNN	XGBoost
Accuracy (%)	**85.51**	82.50	79.66	82.99
Precision (%)	**85.38**	82.00	80.15	82.55
Recall (%)	**85.66**	83.23	78.80	83.61
F1-score (%)	**85.52**	82.61	79.47	83.08
ROC-AUC (%)	**93.04**	90.72	86.85	91.16

Table 3. Accuracy Comparison Table

Model	Accuracy (%)
SVM	**85.51**
Random Forest	82.50
KNN	79.66
XGBoost	82.99

worst, attaining just **79.66%** accuracy and **79.47%** F1-score. In other

4.2. Accuracy and Classification Performance

As for accuracy, the performance margin of SVMs is **2.52, 3.01, and 5.85 percentage points** in comparison with XGBoost, Random Forest, and KNN, respectively. In addition, precision and recall metrics are almost equal (**85.38%** versus **85.66%**) and do not indicate any bias towards one class sentiment by the model.

XGBoost's metric of recall (**83.61%**) exceeds slightly the metric of precision (**82.55%**), which means that this model captures positive reviews with high reliability despite sacrificing somewhat the metric of precision. As for the Random Forest model, the metric of recall reaches **83.23%**. The lowest value of recall among the four models is provided by KNN with **78.80%**.

4.3. ROC-AUC Analysis

The ROC-AUC scores convey a similar tale as those of the accuracy measures. Here too, SVM remains the best at **93.04%**, followed by XGBoost (**91.16%**), Random Forest (**90.72%**), and KNN (**86.85%**), and this implies that SVM sustains its superiority in discriminating between the two sentiment classes regardless of change in the decision threshold.

5. Conclusion

The aim of this paper was to develop a Word2Vec framework to classify movie reviews into positive and negative categories and test it using four different supervised classification algorithms on a set of 40,000 reviews in an 80:20 ratio. The performance of all the four classifiers demonstrated that the SVM was the best of the four classifiers with 85.51%, 85.52%, and 93.04% accuracy, F1 score, and ROC-AUC respectively.

6. Future Work

There are many ways to expand upon this work. Using contextual embedding models like BERT or RoBERTa in place of Word2Vec will allow for context-aware embeddings which incorporate information about the neighboring words instead of modeling the word independently, and more rigorous hyperparameter tuning/ensemble design will also improve performance. Deep learning models with attention mechanism should also be explored for this same purpose. Lastly, incorporating interpretability methods and running tests on other data sets would be an interesting way to prove the interpretability and scalability of this approach.

Conflict of Interest

The authors declare no competing interests related to this work.

Acknowledgment

This research was supported by a fellowship under the Visvesvaraya PhD Scheme, which the authors gratefully acknowledge.

References

[1] M. Giatsoglou, M. G. Vozalis, K. Diamantaras, A. Vakali, G. Sarigiannidis, and K. C. Chatzisavvas, "Sentiment analysis leveraging emotions and word embeddings," *Expert Systems with Applications*, vol. 69, pp. 214–224, 2016.

[2] H. J. Alantari, I. S. Currim, Y. Deng, and S. Singh, "An empirical comparison of machine learning methods for text-based sentiment analysis of online consumer reviews," *International Journal of Research in Marketing*, vol. 39, pp. 1–19, 2021.

[3] V. Yadav, P. Verma, and V. Katiyar, "E-commerce product reviews using aspect based Hindi sentiment analysis," in *Proc. 2021 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, Jan. 27–29, 2021.

[4] Z. Desai, K. Anklesaria, and H. Balasubramaniam, "Business Intelligence Visualization Using Deep Learning Based Sentiment Analysis on Amazon Review Data," in *Proc. 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, July 6–8, 2021, pp. 1–7.

[5] K. K. Mohbey, "Sentiment analysis for product rating using a deep learning approach," in *Proc. 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, Coimbatore, India, Mar. 25–27, 2021, pp. 121–126.

[6] M. S. Darokar, A. D. Raut, and V. M. Thakre, "Methodological Review of Emotion Recognition for Social Media: A Sentiment Analysis Approach," in *Proc. 2021 International Conference on Computing, Communication and Green Engineering (CCGE)*, Pune, India, Sept. 23–25, 2021, pp. 1–5.

[7] M. Devika, C. Sunitha, and A. Ganesh, "Sentiment Analysis: A Comparative Study on Different Approaches," *Procedia Computer Science*, vol. 87, pp. 44–49, 2016.

[8] W. G. Mangold and D. J. Faulds, "Social media: The new hybrid element of the promotion mix," *Business Horizons*, vol. 52, pp. 357–365, 2009.

[9] J. Foster, Ö. Çetinoglu, J. Wagner, J. Le Roux, S. Hogan, J. Nivre, D. Hogan, and J. Van Genabith, "#hardtoparse: POS Tagging and Parsing the Twitterverse," in *Proc. AAAI 2011 Workshop on Analyzing Microtext*, San Francisco, CA, USA, Aug. 7–8, 2011, pp. 20–25.

[10] A. Yousefpour, R. Ibrahim, and H. N. A. Hamed, "Ordinal-based and frequency-based integration of feature selection methods for sentiment analysis," *Expert Systems with Applications*, vol. 75, pp. 80–93, 2017.

[11] R. Xia and C. Zong, "A POS-based ensemble model for cross-domain sentiment classification," in *Proc. 5th International Joint Conference on Natural Language Processing*, Chiang Mai, Thailand, Nov. 8–13, 2011, pp. 614–622.

[12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[13] E. Cambria, S. Poria, R. Bajpai, and B. Schuller, "SenticNet 4: A semantic resource for sentiment analysis based on conceptual primitives," in *Proc. 26th International Conference on Computational Linguistics (COLING)*, Osaka, Japan, Dec. 11–16, 2016, pp. 2666–2677.

[14] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the limits of language modeling," *arXiv preprint arXiv:1602.02410*, 2016.

[15] P. Vateekul and T. Koomsubha, "A study of sentiment analysis using deep learning techniques on Thai Twitter data," in *Proc. 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, Khon Kaen, Thailand, July 13–15, 2016, pp. 1–6.

[16] S. Pal, S. Ghosh, and A. Nag, "Sentiment Analysis in the Light of LSTM Recurrent Neural Networks," *International Journal of Synthetic Emotions*, vol. 9, pp. 33–39, 2018.

[17] A. Hassan and A. Mahmood, "Deep Learning approach for sentiment analysis of short texts," in *Proc. 3rd International Conference on Control, Automation and Robotics (ICCAR)*, Nagoya, Japan, Apr. 24–26, 2017, pp. 705–710.

[18] M. Baroni, G. Dinu, and G. Kruszewski, "Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors," in *Proc. 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, Baltimore, MD, USA, June 23–25, 2014, vol. 1, pp. 238–247.

[19] Y. Yuan and Y. Zhou, "Twitter sentiment analysis with recursive neural networks," in *CS224D Course Project*, Stanford University, Stanford, CA, USA, 2015.

[20] G. Vinodhini and R. Chandrasekaran, "Sentiment analysis and opinion mining: A survey," *International Journal*, vol. 2, pp. 282–292, 2012.

[21] R. Singh and R. Kaur, "Sentiment Analysis on Social Media and Online Review," *International Journal of Computer Applications*, vol. 121, pp. 44–48, 2015.

[22] I. Hemalatha, G. Varma, and A. Govardhan, "Automated Sentiment Analysis System Using Machine Learning Algorithms," *International Journal of Research in Computer and Communication Technology*, vol. 3, pp. 300–303, 2014.

[23] V. Kharde and P. Sonawane, "Sentiment analysis of twitter data: A survey of techniques," *arXiv preprint arXiv:1601.06971*, 2016.