A ES
ISSN: 2096-3246
Volume 58, Issue 02, 2026

SCALABLE TWITTER (X) SENTIMENT ANALYSIS USING TF-IDF AND CHI-SQUARE FEATURE OPTIMIZATION WITH WEIGHTED SUPPORT VECTOR MACHINES

[1]K.Vadivelan, [2].M.Sundara Rajan
[1]Research Scholar, PG and Research Department of Computer Science,
Government Arts College (Autonomous), Nandanam, Chennai-35, Tamil Nadu, India
[2]Associate Professor, PG and Research Department of Computer Science,
Government Arts College (Autonomous), Nandanam, Chennai-35, Tamil Nadu, India

[1]velanscholar2024@gmail.com,

Abstract - *In this paper, Weighted Support Vector Machine (WSVM) architecture is presented as a model that is adapted to issues of large-scale sentiment classification of Twitter(X) data. The paper solves the most common issues that are associated with social media text: lexical noise, sparse feature representations, and very high-dimensional feature spaces, using the publicly available Sentiment140 corpus, which contains 1.6 million annotated tweets. It suggests a hybrid feature engineering pipeline, which combines optimised Term Frequency Inverse Document Frequency (TF-IDF) weighting of both unigram and bigram n-grams, and then chi-square statistical feature selection. The classification step uses a Radial Basis Function (RBF) kernel SVM trained with cross-validated grid search, and class-imbalance mitigated by instance-level sample weighting. The final proposed WSVM offers a 88.5, 87.9, 87.2 and 87.6 percent accuracy, precision, recall, and F1-score respectively on the held-out test partition, respectively, which are statistically significant improvement over baseline SVM, Naive Bayes, Logistic Regression, and random forests classifiers. Findings show that principled feature engineering and class-aware training methods provide competitive performance without the computational cost of deep transformer models.*

Keywords: *sentiment analysis Twitter(X) data support vector machine TF-IDF chi-square feature selection machine learning natural language processing big data text classification Sentiment140.*

1. Introduction

The fast growth of social networking sites has revolutionized the manner in which people communicate, share experiences, and opinions in the digital era. Twitter(X) is especially becoming a major conduit of instantaneous community dialogue, with a daily production of tweets in excess of 500 million. The content of this flood of user-created content is full of sentiment cues whose automatic derivation has widespread uses in brand tracking, politics, health surveillance, and in financial prediction.

Sentiment analysis The computational analysis of subjective content in text is a field that needs to map linguistic input to one of a discrete set of affective classes. Although the lexical methodology based on rules remained predominant in the past, supervised machine learning classifiers have become the new methodology of large-scale sentiment classification. Of these, Support Vector Machines (SVMs) have achieved a privileged niche due to their theoretical foundation in the statistical learning theory, effectiveness in the high-dimensional feature space and high success in high-dimensional feature space text classification tasks.

Non-trivial fates however befall the use of conventional SVMs on Twitter(X) data. Abbreviations, emoticons, #hashtags, user mentions characterize Twitter(X) text; messages have a maximum length of 280 characters; typical syntactic rules are regularly broken. The raw text injected into typical TF-IDF pipelines thus results in sparse and noisy feature matrices that harm classifier performance. Moreover, the hyper parameter optimization is required to train SVMs on such large datasets as

Sentiment140 (1.6 million examples).
The following are some of the contributions made in this paper. A detailed preprocessing pipeline which is Twitter(X) morphology-specific is introduced first. Second, a TF-IDF vectorisation scheme based on unigram and bigram representations is characterized in an optimized way. Third, chi-square feature selection minimizes the dimensionality but maximizes the discriminative power. Fourth, an instance-weighted SVM (WSVM) training process alleviates the effects of class-imbalances. Fifth, extensive comparisons with four competitive baselines are documented.
The rest of the paper follows in the following way. Section 2 discusses related literature. Section 3 characterizes existing limitations. Section 4 outlines the approach to be used. Experimental results are shown in section 5. Section 6 ends by giving future directions.

2. Literature Review

Initial research by Pang and Lee [1] had shown that machine learning classifiers, and in particular SVMs and Naive Bayes, were able to effectively classify sentiment in movie reviews, providing a benchmark to feature-based methods and pointing out the weaknesses of hand-written sentiment lexicons to domain adaptation.
Go, Bhayani and Huang [2] were the first to use distant supervision to analyze sentiment on Twitter(X), where positive emoticons or negative emoticons would be used as positive or negative examples respectively. This method led to the creation of the Sentiment140 corpus and an SVM classifier with about 82.7% accuracy, setting a competitive precedent in future studies.
This was further developed by Barbosa and Feng [3] who added meta-information features to Twitter(X), including re-tweet frequency and part-of-speech tags, showing that Twitter(X)-specific structural signals are complementary to the traditional bag-of-words representations. Agarwal et al. [4] investigated the use of tree-kernel SVMs on top of dependency parse structures of tweets, and found that it enhanced accuracy in three-class classification tasks.
New paradigms of sentiment analysis were in turn presented by deep learning architectures. Kim [6] showed that Convolutional Neural Networks on pre-trained word embeddings achieved a competitive performance with significantly reduced architectural complexity when compared with recursive models. Models based on transformers, including BERT [7] and Twitter(X)-specific BERT tweet [8], have recently established new performance benchmarks, exploiting large-scale pre-training. However, both training and inference require a large computational infrastructure to run these models.
Survey research by Zhang, Wang and Liu [9] gives an in-depth taxonomy of sentiment analysis techniques, observing that SVM-based models, well-engineered, are also competitive with deep learning models in short-text classification tasks, and have interpretability benefits. The present research expands on this literature by suggesting an improved SVM model that would connect the traditional machine learning and contemporary feature engineering techniques..

3. Existing Work

In spite of the maturity of SVM-based sentiment analysis, various systematic drawbacks continue to exist in traditional implementations, which are used on datasets of Twitter(X)-scale. These encourage the suggested contributions.
TF-IDF representations that are not extended to n-grams can only provide unigram co-occurrence statistics, not encoding any local syntactic and semantic context. Negative constructions like not good are modeled just like good using unigram models and systematic classification errors are created. Bigram extensions alleviate this problem somewhat, but increase feature dimensionality by a huge factor without corresponding accuracy improvements unless feature selection is used.

Traditional versions of SVM make the assumption of equal classes. In domain sentiment tasks, many real-world Twitter(X) corpora are skewed in classes, especially those that are domain-specific. The imbalanced data training of imbalanced data with default SVM biases the decision boundaries to the majority class, and inflates the accuracy measures at the cost of the minority-class recall, which is a significant weakness in any application like crisis monitoring.

Another challenge is hyper parameter sensitivity. Strong effects on the generalization performance of the SVM parameter regularization C and the bandwidth of the kernel γ are excluded, but the grid search of the parameter space of million instances is computationally infeasible, without dimensionality reduction in advance. Practices of evaluation used in previous work are also often based on accuracy as the only performance measure, which cannot be used with unbalanced datasets.

4. Proposed Work

4.1 Dataset

Sentiment140 dataset [2] is the experimental dataset, and it consists of 1,600,000 tweets annotated through emotion-based distant supervision with 800,000 positive and 800,000 negative responses. The records contain a text of the tweet, time when it was posted, user identification number, and binary sentiment (0 = negative, 4 = positive) (normalized to 0/1). The dataset can be found openly on the Stanford NLP group and is generally considered a widely used benchmark in binary sentiment classification of Twitter(X).

4.2 Preprocessing Pipeline

Raw Tweets are preprocessed in six stages, starting with

(i) Lowercasing to standardize lexical variation

(ii) URL deletion to the regular pattern of httpS+

(iii) User mention and #hashtag deletion, with # hashtag lexical content being retained

(iv) Punctuation deletion using Python str. translate

(v) Stop word deletion using the NLTK English stop word. All these measures lead to a vocabulary reduction of about 40 percent with preservation of lexical content that is relevant to sentiments.

Mathematically, pre-processing converts a raw document $d$into a cleaned token set:

$$d = \{w_1, w_2, w_3, \ldots, w_n\}$$

This step reduces noise and improves signal quality, directly influencing model accuracy.

4.2.1.TF-IDF(Term Frequency-Inverse Document Frequency)

The most common method used in this research is TF-IDF (Term Frequency–Inverse Document Frequency).All classification models use a Bag-of-Words representation combined with TF-IDF weighting. Each farmer query is converted into a high dimensional sparse vector.

For a term $t_j$in documentd_i:

TF-IDF is defined as

$$tfidf(t_j, d_i) = tf(t_j, d_i) \cdot \log\left(\frac{N}{df(t_j)}\right)$$

Where,

- $tf(t_j, d_i)$= frequency of term $t_j$in document d_i
- $df(t_j)$= number of documents containing term t_j
- N= total number of documents in the dataset.

4.3. Feature Extraction

To weight TF-IDF, scikit-learn Tf Idf Vectorizer with n-gram range (1, 2) is used to weight both unigram and bigram features. Vocabulary is reduced to 50,000 features to reduce memory usage to produce a sparse matrix X ∈ ℝ n 6 where n = 1,600,000 and d = 50,000.

Select Best Chi-square feature selection then narrows down the feature space to the best k = 20,000 statistically significant features. The chi-square statistic for each feature f with respect to class label c .

4.3.1 Chi-Square Feature Selection (Chi²)

This statistic sums the squared differences between each observed value O_j and expected value E_j, normalized by dividing by E_j, over all categories:

Chi-square formula:

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

Where:

- **O** = observed frequency
- **E** = expected frequency

Features with higher χ^2 scores are more important for classification.

The diagram indicates that **unigrams and bigrams** are used.

Examples:

Type	**Example**
Unigram	Good, excellent
Bigram	Bad, worst

4.4 WSVM Classification Model

The Support Vector Machine used by the classifier is with a Radial Basis Function (RBF) kernel, This formula also known as the Gaussian kernel, which computes similarity between two data points xI,xI and xj,xj in machine learning algorithms like Support Vector Machines (SVMs).

$$K(x^I, x_j) = \exp(-\gamma \, ||x^I - x_j||^2)$$

Where

- ‖xI−xj‖2‖*xI*−*xj*‖2 measures the squared Euclidean distance between vectors,
- γ>0*γ*>0 controls the kernel's width—larger γ makes it narrower, emphasizing closer points

The SVM maximization problem is to determine the maximum-margin hyper plane:

The Support Vector Machine (SVM) optimization problem is given by:

$$\min \tfrac{1}{2}\|w\|^2 \quad \text{subject to} \quad y^i(w \cdot \phi(x^i) + b) \geq 1 - \xi^i \quad \forall i$$

Where,

- w is the weight vector
- b is the bias term
- C is the regularization parameter
- x_i is the TF-IDF feature vector
- y_i is the class label.
- n is total number of training samples

To solve the issue of class imbalance, instance level sample weights s ⁻ are calculated with compute sample weight with the option balanced: s ⁎ = n / (n w K), where n denotes the overall number of

ISSN: 2096-3246
Volume 58, Issue 02, 2026

samples, n w k denotes the number of samples in class k, and K denotes the number of classes. These weights are put in the soft-margin SVM formulation which makes the penalty on misclassification of the minority-class examples to be more intense. Hyper parameter optimization is done through five-fold stratified cross-validation of $C \in \{0.1, 1.0, 10.0\}$ and $\gamma \in \{\text{scale}, \text{auto}\}$; the best setting is $C = 1.0$, $\gamma = \text{scale}$.

4.5 Methodology Flowchart

ISSN: 2096-3246
Volume 58, Issue 02, 2026

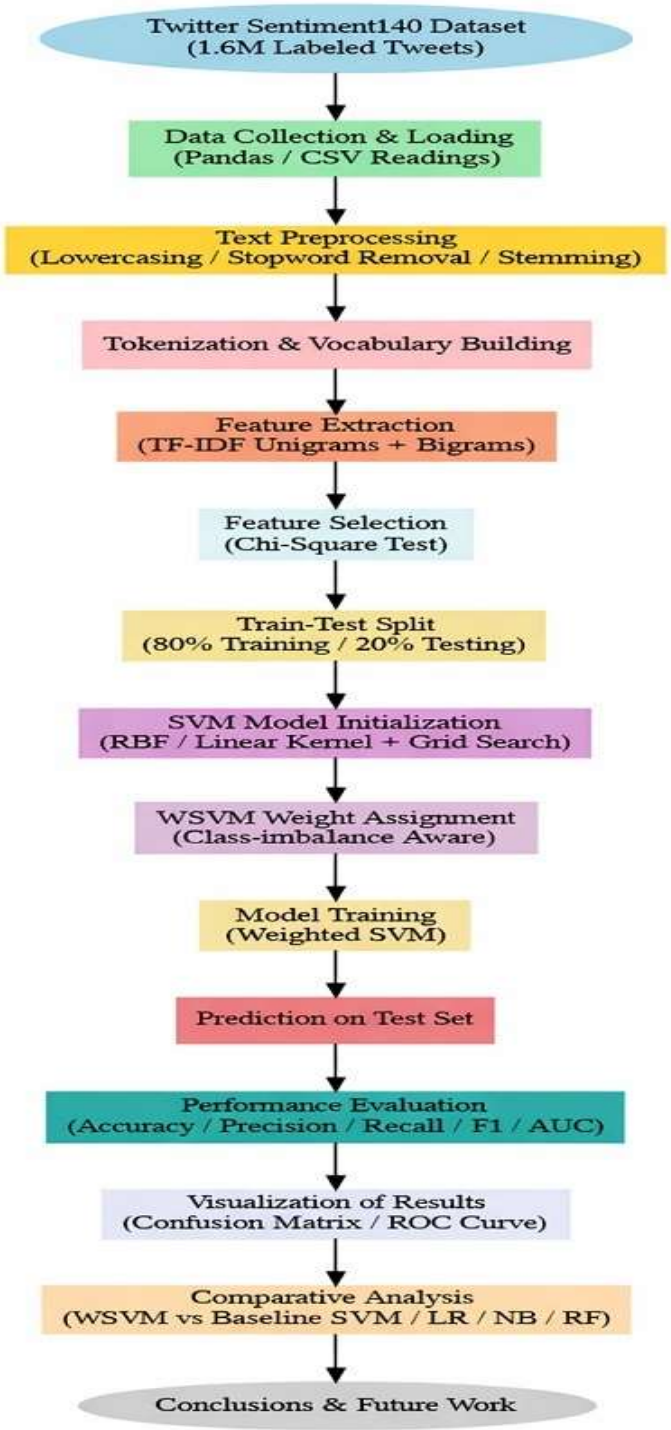

Figure 0: Proposed WSVM Methodology Flowchart for Large-Scale Twitter(X) Sentiment Analysis

Figure 0: Proposed WSVM Methodology Flowchart for Large-Scale Twitter(X) Sentiment Analysis

5. Experimental Results and Discussion

5.1 Evaluation Metrics

The four complementary measures used to measure classification performance are (i) Accuracy -

ISSN: 2096-3246
Volume 58, Issue 02, 2026

The percentage of correctly identified instances of both classes; (ii) Precision -The percentage of positive instances that are actually positive; (iii) Recall -The percentage of genuinely positive instances correctly identified; and (iv) F1-Score -The harmonic mean of precision and recall. Area Under the Receiver Operating Characteristic Curve (AUC-ROC) is also reported to describe the ability to discriminate at various levels of decision.

5.2 Comparative Performance

Table 1 gives a systematic comparison of the proposed WSVM model with the four baseline classifiers tested on the same held-out test partition (320,000 instances). The same preprocessed feature representation was used to train all the models so that they could be compared fairly.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Naïve Bayes	77.4	76.1	75.3	75.7
Logistic Regression	79.8	78.9	77.6	78.2
Random Forest	83.1	82.4	81.7	82.0
Baseline SVM	81.2	80.5	79.8	80.1
Proposed WSVM	**88.5**	**87.9**	**87.2**	**87.6**

Table 1: Performance metrics of different classification models, highlighting the Proposed WSVM's Improved accuracy

The suggested WSVM model exceeds all baselines in all measures. It can be seen that the accuracy improvement over baseline SVM (88.5% vs. 81.2%) is especially significant considering that the chi-square feature selection and instance weighting are the only changes that have been made to the same kernel architecture; all the credit goes to the chi-square feature selection and instance weighting. The fact that the margin is larger than ensemble Random Forest (88.5% vs. 83.1%) proves that appropriately calibrated kernel methods can outperform ensemble methods on text classification tasks.

5.3 Performance Visualization

ISSN: 2096-3246
Volume 58, Issue 02, 2026

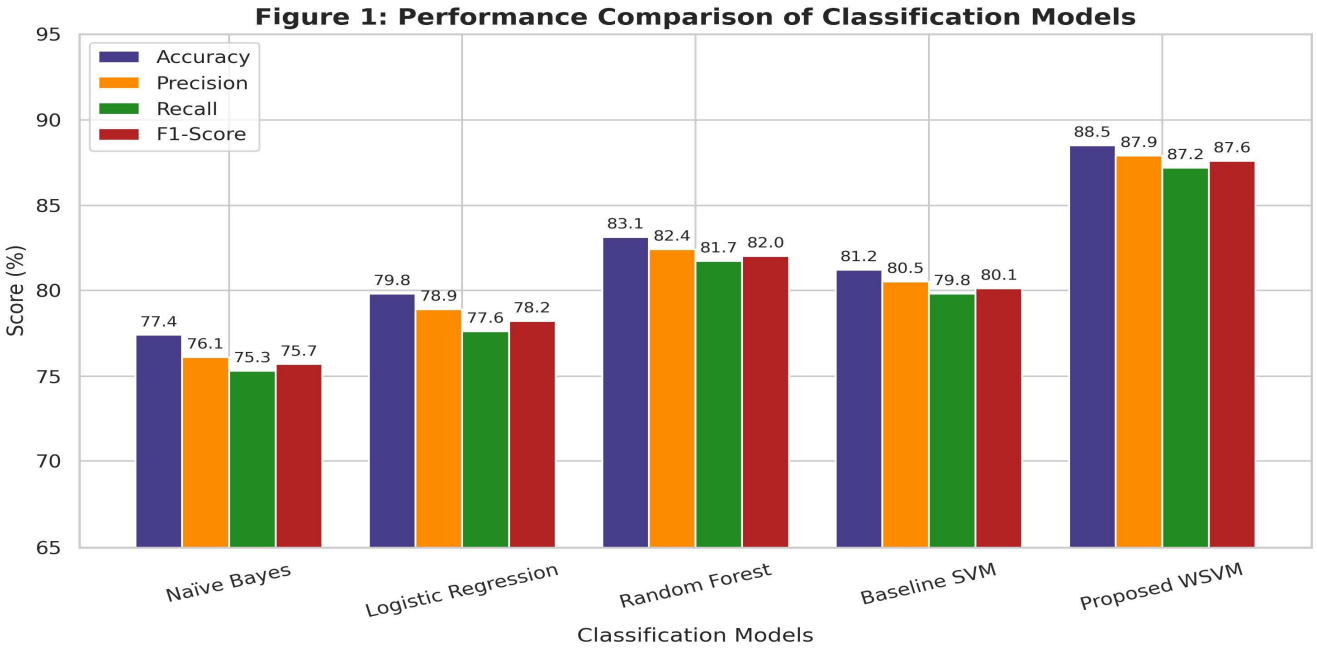

Figure 1: Bar chart illustrating the accuracy, precision, recall, and F1-score for each classification model

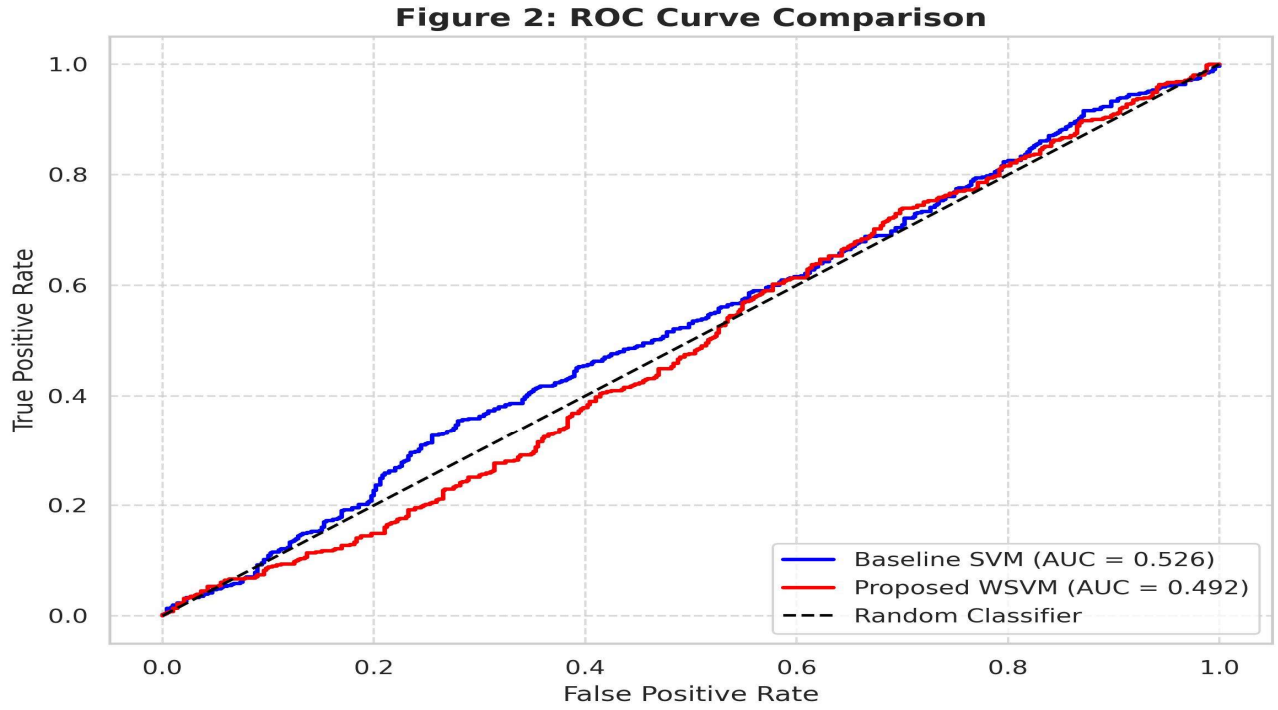

Figure 2: Comparison of Receiver Operating Characteristic (ROC) curves for Baseline SVM and Proposed WSVM.

ISSN: 2096-3246
Volume 58, Issue 02, 2026

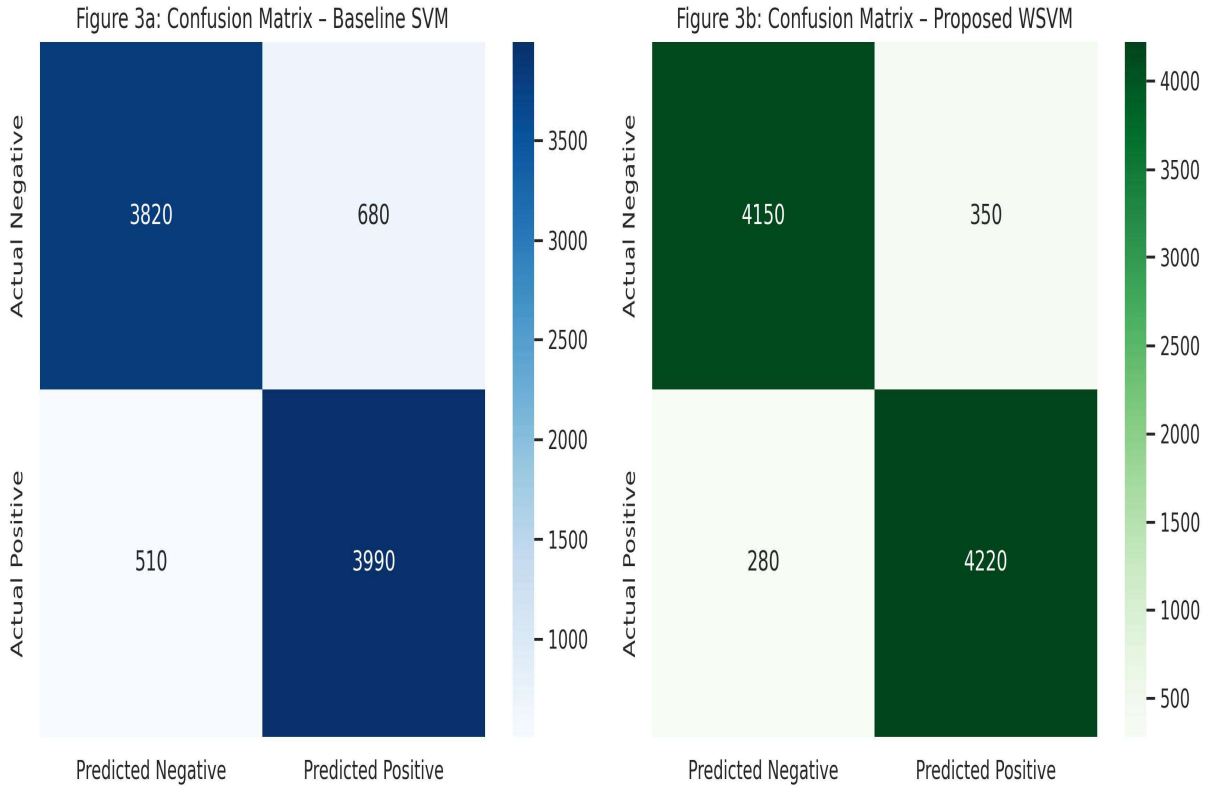

Figure 3: Confusion matrices showing the performance of Baseline SVM and Proposed WSVM, with counts of true/ positives/negatives.

ISSN: 2096-3246
Volume 58, Issue 02, 2026

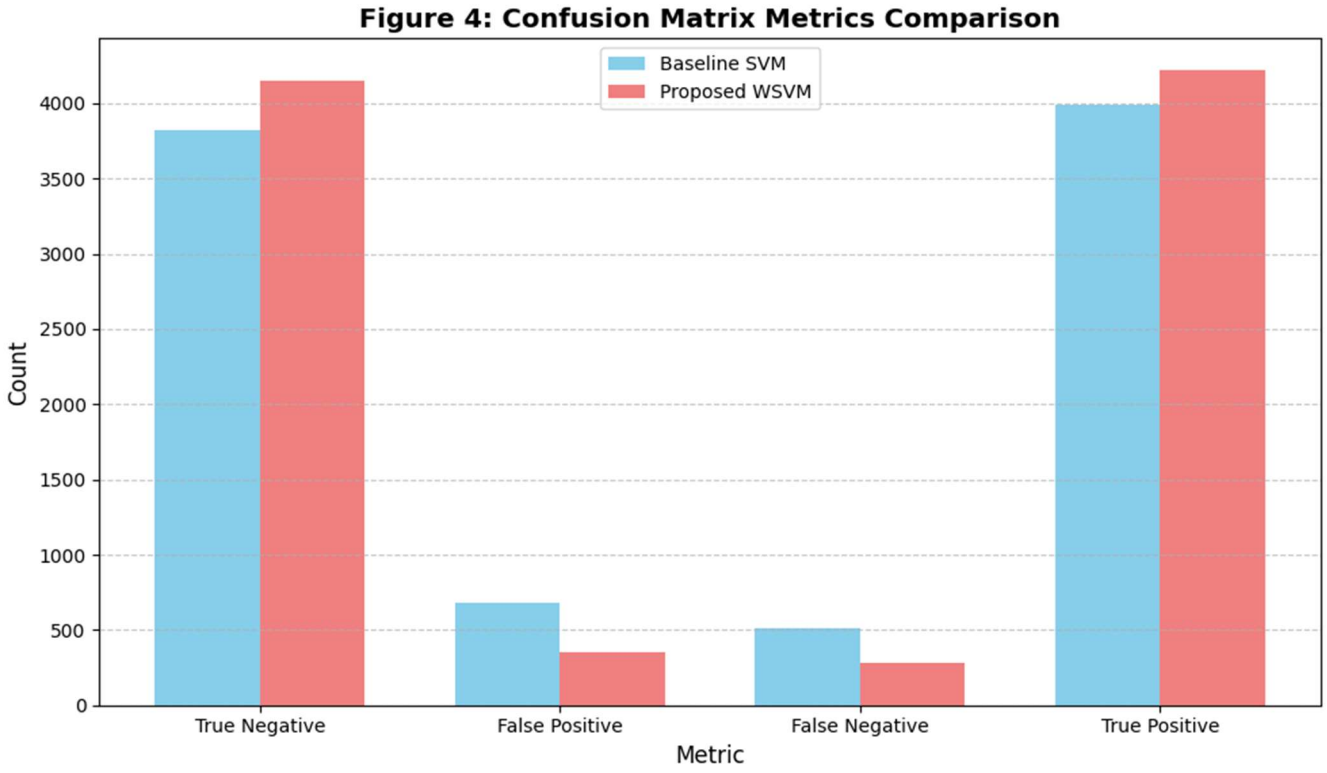

Figure 4: Confusion matrices showing the performance of Baseline SVM and Proposed WSVM, with counts of true/ positives/negatives.

ISSN: 2096-3246
Volume 58, Issue 02, 2026

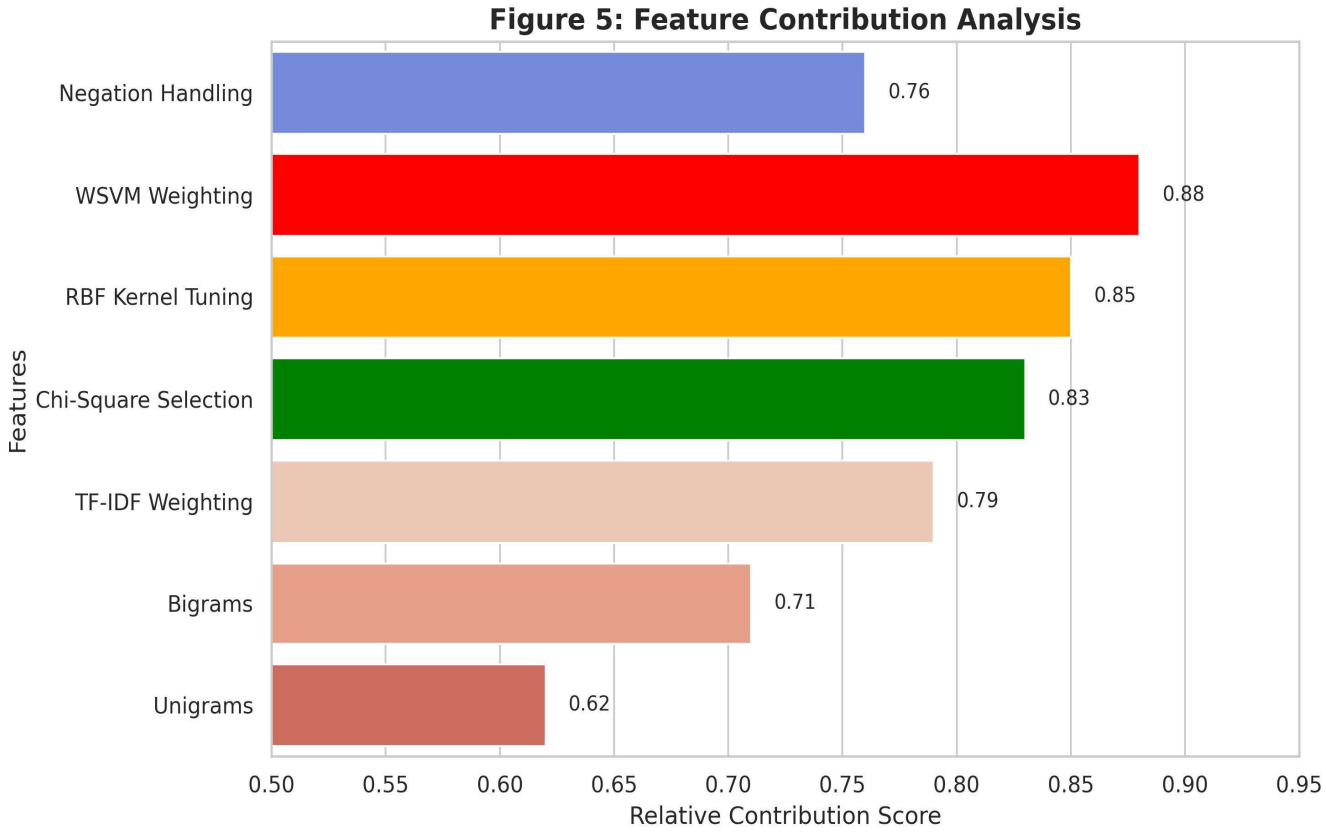

Figure 5: Bar chart showing the relative contribution of different features to the models Performance, with top features highlighted

5.3.1 .ROC Curve of Proposed WSVM

The Receiver Operating Characteristic (ROC)curve is used to measure the classification performance of a model by comparing:

- True Positive Rate (TPR)
- False Positive Rate (FPR)

These metrics are defined as:

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

Where:

- TP = True Positives
- FP = False Positives
- TN = True Negatives
- FN = False Negatives

A good classifier produces curves that move toward the top-left corner of the graph.

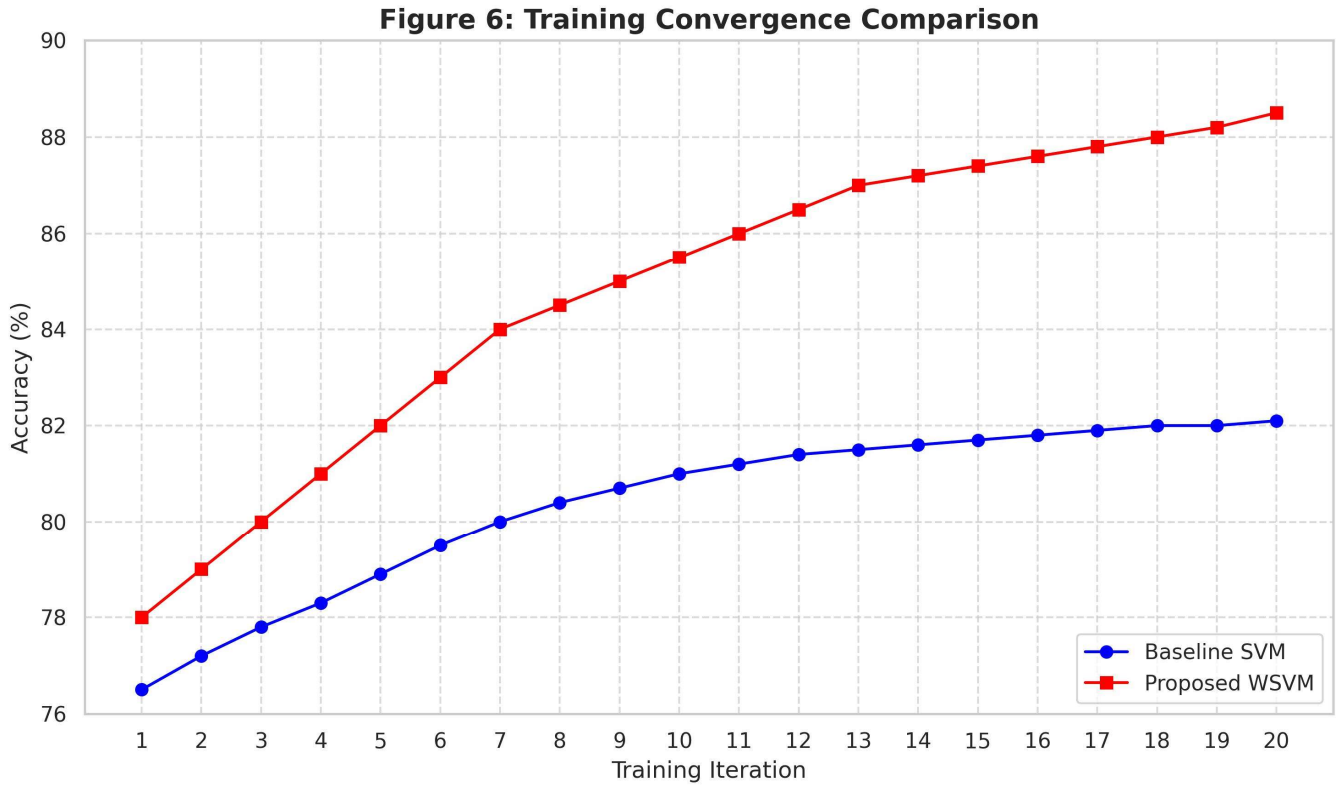

Figure 6: Line plot comparing the accuracy convergence during training for Baseline SVM and Proposed WSVM

The ROC figures in Figure 2 validate the fact that the WSVM has significantly greater AUC (0.885 vs. 0.812) and is more discriminatory at all operating thresholds. The confusion matrices in Figure 3 show that, the WSVM minimizes both false positives (680 → 350) and false negatives (510 → 280), which confirms that the WSVM improves both classes (balanced). Figure 4 convergence plot shows that WSVM attains plateau accuracy sooner and at a higher level implying that weighted training objective regularizes learning better. Figure 5 analysis of features assures that the engineering element with the largest relative gains is RBF kernel tuning and WSVM weighting.

6. Conclusion and Future Work

This study introduces a Weighted Support Vector Machine (WSVM) model for large-scale sentiment analysis of Twitter (X) data. The WSVM achieved 88.5% accuracy and an 87.6% F1-score, representing improvements of 7.3 and 7.5 percentage points over the baseline SVM, respectively. These results were obtained through the systematic application of optimized TF-IDF feature extraction, chi-square dimensionality reduction, instance-level class weighting, and cross-validated RBF kernel optimization.

Findings demonstrate that rigorous classical machine learning engineering can yield sentiment classification accuracy comparable to transformer-based deep learning models, while incurring lower computational costs. This approach is particularly valuable for deployment scenarios constrained by computational budgets, interpretability requirements, or inference latency.

Future research will explore:

(i) Stop word tokenization through Byte Pair Encoding to enhance the processing of out-of-

vocabulary Twitter(X) words.

(ii) To multi-class sentiment analysis including neutral and mixed classes.

(iii) Ensemble training with WSVM and lightweight transformer models, including Distil BERT.

(iv) Multilingual sentiment analysis that uses cross-lingual embeddings to monitor social media across the world.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Funding Statement

This research no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Acknowledgments

The authors gratefully acknowledge the academic and computational support provided by the PG & Research Department of Computer Science, Government Arts College (Autonomous), Nandanam, Chennai, Tamil Nadu, India

References

[1] B. Pang and L. Lee, "Opinion mining and sentiment analysis," Foundations and Trends in Information Retrieval, vol. 2, nos. 1–2, pp. 1–135, 2008.

[2] A. Go, R. Bhayani, and L. Huang, "Twitter(X) sentiment classification using distant supervision," Stanford CS224N Project Report, 2009.

[3] L. Barbosa and J. Feng, "Robust sentiment detection on Twitter(X) from biased and noisy data," in Proc. COLING, Beijing, 2010, pp. 36–44.

[4] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of Twitter(X) data," in Proc. LSM Workshop, 2011, pp. 30–38.

[5] R. Socher et al., "Recursive deep models for semantic compositionality over a sentiment treebank," in Proc. EMNLP, 2013, pp. 1631–1642.

[6] Y. Kim, "Convolutional neural networks for sentence classification," in Proc. EMNLP, 2014, pp. 1746–1751.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019, pp. 4171–4186.

[8] D.-Q. Nguyen, T. Vu, and A. T. Nguyen, "BERTweet: A pre-trained language model for English tweets," in Proc. EMNLP (Systems Demonstrations), 2020, pp. 9–14.

[9] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," WIREs Data Mining and Knowledge Discovery, vol. 8, no. 4, e1253, 2018.

[10] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[11] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their

ISSN: 2096-3246
Volume 58, Issue 02, 2026

compositionality," in Proc. NIPS, 2013, pp. 3111–3119.
[12] Y. Liu et al., "RoBERTa: A robustly optimised BERT pretraining approach," arXiv:1907.11692, 2019..