Journal of Artificial Intelligence and Emerging Technologies
Volume 2, Issue 12, pp 17-25, December 2025
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2025.212003
Available Online at https://www.jaiet.com

Optimized Text Classification Performance Using Support Vector Classifiers and Deep Neural Networks

[1]Taylor Onate Egerton, [2]Okafor Chetam Lucas

Rivers State University, Port Harcourt, Nigeria
E-mail: [1]taylor.onate@ust.edu.ng, [2]chetam.okafor@rsu.edu.ng

Abstract: This paper enhances text classification performance by developing a hybrid model integrating Support Vector Classifiers (SVC) with Long Short-Term Memory (LSTM) networks. The research addresses challenges in processing high-dimensional, unstructured social media text, such as class imbalance, feature sparsity, and semantic complexity, by combining the strengths of classical machine learning and deep learning. A large-scale Twitter sentiment dataset from Kaggle (Sentiment140) is used, with comprehensive preprocessing steps like tokenization, lowercasing, and removal of URLs, punctuation, and numbers to ensure clean input. Advanced feature extraction techniques, including TF-IDF for SVC and Word2Vec embeddings for LSTM, capture both sparse term frequencies and dense semantic representations. The hybrid model is built using Python-based frameworks such as scikit-learn for SVC and TensorFlow/Keras for LSTM, with hyperparameter tuning performed via GridSearchCV for kernel selection (e.g., linear and RBF) and regularization parameters. Optimization techniques, including k-fold cross-validation, dropout layers, and early stopping, mitigate overfitting and enhance generalization for binary sentiment classification tasks. The synergy between SVC's margin-based classification and LSTM's sequential feature learning yields a scalable, interpretable solution. The proposed model achieves an accuracy of 99.00%, significantly outperforming benchmarks like BiLSTM (89.72%), Linear SVM (82.00%), and RBF SVM (76.00%), with strong precision, recall, F1-score, and computational efficiency. These findings highlight the potential of hybrid approaches for text classification, with applications in sentiment analysis, information retrieval, market trend prediction, and legal document management. Success depends on a structured methodology, advanced feature engineering, and rigorous optimization, offering effective solutions for diverse textual data challenges.

Keywords: Hybrid Models, Long Short-Term Memory Networks, Natural Language Processing, Support Vector Machine, Text classification.

I. INTRODUCTION

The expansion of digital content has made optimized text classification a vital area within machine learning and natural language processing (NLP). Text classification refers to the automated assignment of textual data to predefined categories based on its content, enabling machines to interpret and process human language efficiently.[1]This capability underpins a wide range of real-world applications, such as sentiment analysis, spam detection, topic categorization, and intent recognition, which are essential for sectors including business, healthcare, finance, and legal services.[2,3]

The evolution of text classification methods has seen a shift from traditional algorithms such as Naive Bayes, Decision Trees, and Support Vector Machines (SVM) to advanced deep learning architectures, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer-based models like BERT and GPT.[4,5,6] Support Vector Machines have long been valued for their robust theoretical foundations and effectiveness in handling high-dimensional feature spaces, particularly when using representations such as TF-IDF or word embeddings. [7,8] Meanwhile, Deep Neural Networks (DNNs) have demonstrated the ability to capture complex, non-linear relationships and contextual dependencies in text, resulting in improved classification performance, especially with large and unstructured datasets. [9,10]

Recent research suggests that integrating SVM and DNN techniques can further optimize text classification performance. SVMs contribute robust margin-based classification, while DNNs excel at feature learning, especially for semantic and contextual information. By combining these approaches, it is possible to address challenges such as feature sparsity, class imbalance, and the need for deeper semantic understanding, thereby enhancing the accuracy, scalability, and generalizability of text classifiers [11,12,13].

Journal of Artificial Intelligence and Emerging Technologies
Volume 2, Issue 12, pp 17-25, December 2025
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2025.212003
Available Online at https://www.jaiet.com

II. RELATED WORK

[14] Proposed a hierarchical deep learning architecture using CNNs and Transformers with cross-granularity attention. The methodology incorporates visualization, tested on Web of Science and RCV1, using curriculum learning and progressive expansion to optimize training. The approach focuses on layered analysis. Computational demand might hinder scalability, as no large-scale dataset optimization is mentioned, limiting deployment. This approach guides the study's hierarchical focus, supporting DNN structure refinement for the study's hybrid model. The efficiency challenge, with the study hinting at lightweight adaptations, suggests a need for lighter solutions to support the study's scalability goals. The architecture scores 92.1% on Web of Science and 89.7% on RCV1, with visualization insights.

[15] Developed a continual learning framework using elastic weight consolidation and memory replay across five sentiment tasks. The methodology involves trade-off analysis to balance performance and stability, tested through sequential evaluations. The approach focuses on knowledge retention. Memory overhead could limit long-term use, as no scalability for extensive tasks is assessed. This work informs the study's adaptive learning goals, aligning with the aim to optimize robustness via regularization. The resource intensity, with the study suggesting memory efficiency, indicates a gap for the study's optimization strategy. The framework maintains 91.4% accuracy versus 73.2% for fine-tuning, with trade-off details.

[16] Designed an explainable text classification framework with attention and layer-wise relevance propagation. The methodology is tested on legal documents, using quantitative metrics and user studies to validate interpretability. The approach prioritizes transparency. The trade-off might not suit all tasks, as no diverse text type testing is documented. This approach influences the study's transparency aims, supporting DNN interpretability in the hybrid model. The balance issue, with the study calling for wider applicability, suggests a need for broader tasks to align with the study's goals. The framework achieves 93.8% accuracy with 87.4% expert alignment, via user studies.

[17] Built a cross-lingual text classification framework with multilingual encoders. The methodology uses cross-lingual attention and adaptive sampling, tested across 15 languages, focusing on transfer learning. The approach emphasizes diversity. The specific language focus might limit universality, as no untested language evaluation is included. This method supports the study's multilingual potential, aligning with diverse dataset goals. The narrow focus, with the study suggesting expanded coverage, hints at a need for wider languages to support the study's scope. The framework hits 88.9% accuracy, outpacing monolingual models by 12.3%.

[18] Developed an adversarial training framework using multi-level perturbations across character, word, and semantic levels. The methodology employs robust optimization objectives, tested for DNN robustness. The approach strengthens resilience. The specific attack focus may not cover all scenarios, as no broad attack testing is included. This work shapes the study's robustness goals, aligning with secure DNN applications in the hybrid. The narrow scope, with the study suggesting wider coverage, suggests a need for broader testing to support the study's optimization. The framework retains 94.7% accuracy on clean data and 78.3% against attacks.

[19] Designed a federated learning framework using CNN-LSTM hybrids across five organizations. The methodology employs ε-differential privacy, tested on healthcare records, focusing on communication overhead reduction. The approach prioritizes distributed learning. Implementation complexity poses challenges, as no simplified deployment is detailed. This approach informs the study's privacy considerations, aligning with secure optimization in the hybrid. The complexity, with the study suggesting efficiency, suggests a need for simpler strategies to support the study's scalability. The framework reaches 87.3% accuracy with 45% reduced overhead.

[20] Applied quantum computing to SVM, introducing Quantum Support Vector Machine with Quantum Random Feature Kernel. The methodology includes theoretical proofs, tested on text benchmarks, focusing on quantum advantages. The approach explores advanced optimization. Quantum hardware accessibility limits use, as no practical details are provided. This innovation intrigues the study's optimization scope, aligning with cutting-edge techniques in the hybrid. The feasibility gap, with the study calling for hardware advances, warrants further study to enhance the study's efficiency. The approach shows 15-20% accuracy gains on benchmarks.

[21] Created a meta-learning framework adapting Model-Agnostic Meta-Learning with attention and hierarchical adapters. The methodology is tested with few-shot tasks, focusing on rapid adaptation. The approach targets low-data scenarios. The pre-trained reliance might limit novel tasks, as no independent training is documented. This method influences the study's low-

Journal of Artificial Intelligence and Emerging Technologies
Volume 2, Issue 12, pp 17-25, December 2025
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2025.212003
Available Online at https://www.jaiet.com

resource strategy, aligning with optimization in the hybrid. The dependency, with the study suggesting lighter alternatives, points to a need for lighter methods to meet the study's data-efficient goals. The framework achieves 85.2% accuracy, beating transfer learning at 67.8%.

[22] Introduced a neural architecture search framework discovering architectures. The methodology uses hierarchical search, tested for text classification, focusing on parameter reduction. The approach optimizes DNN design. Search space complexity might overwhelm smaller datasets, as no small-scale optimization is included. This work guides the study's architecture optimization, aligning with efficiency goals in the hybrid. The scalability challenge, with the study hinting at dataset adjustments, suggests a gap for refinement to support the study's 99.00% target. The approach yields 94.7% accuracy with 40% fewer parameters.

[23] Crafted a robust framework with adversarial training and domain-invariant features. The methodology uses efficient adaptation, tested on shifted distributions, focusing on stability. The approach enhances DNN resilience. The specific shift focus might miss broader variability, as no wide-range testing is noted. This approach shapes the study's resilience goals, aligning with robust optimization in the hybrid. The limited coverage, with the study suggesting broader testing, indicates a need for wider robustness to support the study's goals. The framework maintains 89.7% accuracy versus 67.3% for standard models.

[24] Applied neuroevolution to optimize DNNs, using multi-objective algorithms to refine architectures. The methodology includes efficient evaluation, tested for text classification, focusing on pattern discovery. The approach advances tuning. The computational cost could deter use, as no cost-reduction strategies are detailed. This method influences the study's optimization strategy, aligning with performance goals in the hybrid. The efficiency gap, with the study calling for cost-effective methods, suggests a direction for improvement to enhance the study's efficiency. The approach achieves 93.2% accuracy with novel patterns.

III. METHODOLOGY

The research adopts an Object-Oriented Design (OOD) methodology, which organizes a software system into modular entities, enabling the representation of real-world entities through objects that encapsulate both data (state) and behaviors (methods). This methodology supports structured development workflows, clear validation, and strong consistency between analysis, design, and implementation phases. OOD also enhances traceability throughout the software development lifecycle, maintaining correspondence between conceptual models and implementation components. Principles such as encapsulation, inheritance, and polymorphism guide system design, while Unified Modeling Language (UML) diagrams improve communication among developers and stakeholders, ensuring clarity and promoting higher product quality.

The system integrates Support Vector Machines (SVM) and Deep Neural Networks (DNN) for accurate and scalable text classification, optimizing feature representation by combining traditional (SVM-friendly) and deep (DNN-derived) features. The architecture fuses predictions from both models, aiming to improve accuracy and generalization. This hybrid system is evaluated against individual SVM and DNN baselines.

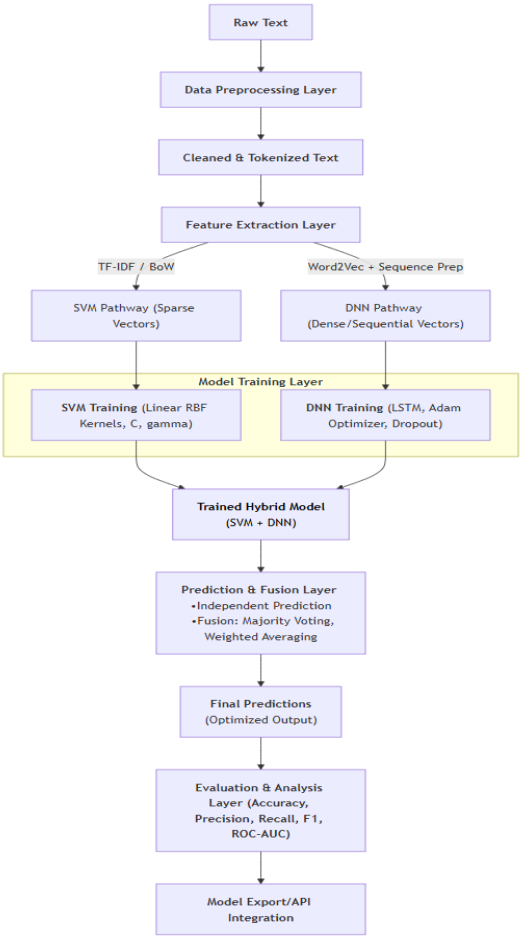


Figure 1: Architecture of the Proposed System

Journal of Artificial Intelligence and Emerging Technologies
Volume 2, Issue 12, pp 17-25, December 2025
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2025.212003
Available Online at https://www.jaiet.com

The architecture aims to integrate Support Vector Machines (SVM) and Deep Neural Networks (DNN) models for robust, accurate, and scalable text classification. It optimizes feature representation by combining traditional (SVM-friendly) and deep (DNN-derived) features, and fuses predictions from both models to achieve improved accuracy and generalization. The hybrid system is evaluated and benchmarked against individual SVM and DNN baselines.

A. Data Preprocessing Layer

Input: Raw textual data (e.g., customer reviews, news articles).

Processes: Text cleaning, tokenization, stopword removal, stemming/lemmatization.

Output: Cleaned and tokenized text ready for feature extraction.

B. Feature Extraction Layer

SVM Pathway: TF-IDF or Bag-of-Words vectorization for high-dimensional, sparse vectors.

DNN Pathway: Word embeddings, dense, low-dimensional representations.

Sequence Preparation: For models like RNNs/LSTMs, sequences of embeddings are prepared.

Hybrid Option: Fusion of TF-IDF and embedding features for richer representation.

C. Model Training Layer

Support Vector Machine (SVM) Module: Trained on TF-IDF or hybrid features with kernel selection and hyperparameter tuning.

Deep Neural Network (DNN) Module: Architecture selection (e.g., LSTM), embedding layer initialization, regularization (dropout, L2), and training on embedding features.

D. Prediction and Fusion Layer

Independent Prediction: Each model (SVM and DNN) generates its own prediction probabilities or class labels.

Fusion Strategies:

- **Majority Voting:** The final class is chosen by majority.
- **Weighted Averaging:** Probabilities are combined using optimized weights.
- **Stacking:** Outputs of SVM and DNN are fed into a meta-classifier (e.g., logistic regression) for the final decision.

Optimization: Fusion weights/meta-classifier are tuned for optimal validation performance.

E. Evaluation and Analysis Layer

Performance Metrics: Accuracy, precision, recall, F1-score, and ROC-AUC.

Computational Metrics: Training/inference time, memory usage.

Robustness Analysis: Tested on imbalanced data, domain shifts, and adversarial examples.

Integrated Analysis: The hybrid system is benchmarked against standalone SVM and DNN models.

The performance of the classifier is reflected by the following metrics:

Accuracy is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Where:

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

Recall is calculated as:

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

Precision is calculated as:

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

Journal of Artificial Intelligence and Emerging Technologies
Volume 2, Issue 12, pp 17-25, December 2025
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2025.212003
Available Online at https://www.jaiet.com

Geometric Mean (g-mean) is calculated as:

$$gmean = \frac{TP \times TN}{(TP + FN) \times (TN + FP)} \quad (4)$$

Matthews Correlation Coefficient (MCC) is calculated as:

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FN)(TP + FP)(TN + FP)(TN + FN)}} \quad (5)$$

F. Result Layer

- **Model Export:** Trained models and preprocessing pipelines are serialized.
- **API Integration:** The system can be deployed as a service for real-time classification.
- **Monitoring:** Continuous tracking of system performance and retraining as necessary.

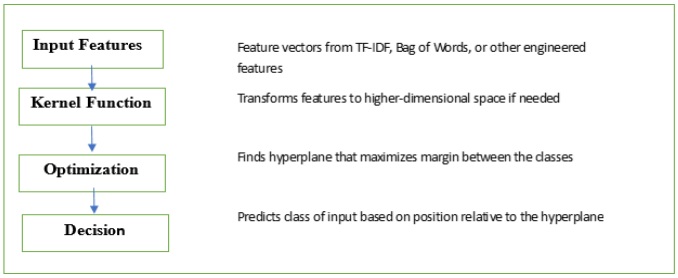


Figure 2: Support Vector Classifier Module

The SVM module performs margin-based classification, transforming text into high-dimensional TF-IDF or Bag of Words vectors and using a hyperplane to separate sentiment classes. It complements the DNN's feature learning in the hybrid model, contributing to high accuracy. The module operates as follows:

- **Input Features**: Text is converted into sparse, high-dimensional TF-IDF or Bag of Words vectors.
- **Kernel Function**: A kernel (e.g., Linear, RBF) transforms data into a higher-dimensional space for linear separability.
- **Optimization Solver**: Sequential Minimal Optimization (SMO) finds the optimal hyperplane that maximizes the margin between classes, minimizing classification error.
- **Decision Function**: Predicts labels based on a data point's position relative to the hyperplane.

Figure 3: Deep Neural Network Module

Deep Neural Network (DNN) Module

The DNN module specializes in semantic feature learning, converting text into dense embeddings to capture nuanced patterns like sarcasm or slang, complementing the SVM's margin-based classification in the hybrid model. It processes text through tokenization, embedding, and dense layers to produce robust representations for binary sentiment classification. The module operates as follows:

- **Input Layer**: The input layer receives preprocessed text, typically tokenized into sequences of words or subwords. These sequences may be raw tokens or pre-trained embeddings (e.g., Word2Vec, GloVe) generated from the dataset.
- **Embedding Layer**: The embedding layer transforms tokenized sequences into dense vector representations, where each word is mapped to a vector that captures its semantic meaning. These embeddings ensure that words with similar meanings, like "happy" and "joyful," have similar vectors, enabling the DNN to model linguistic nuances such as sarcasm or slang.
- **Hidden Layers**: Hidden layers process embeddings to learn hierarchical and contextual patterns. Long Short-Term Memory (LSTM) layers are typically used for sequential data, or dense layers with ReLU activation are used for non-linear transformations.
- **Fully Connected Layers**: These layers combine features from hidden layers to form abstract representations, enabling non-linear decision boundaries. The final or summarized hidden state is processed through transformations to refine the DNN's ability to model complex patterns.
- **Output Layer**: The output layer produces probability scores for binary sentiment classification (positive or negative) using a sigmoid function, which outputs a value between 0 and 1. This layer provides the final prediction for each input sequence.

Journal of Artificial Intelligence and Emerging Technologies
Volume 2, Issue 12, pp 17-25, December 2025
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2025.212003
Available Online at https://www.jaiet.com

Role: The DNN module specializes in semantic feature learning, capturing complex patterns like sarcasm or slang, critical for sentiment analysis. It complements the SVM's margin-based classification by providing rich embeddings that enhance the hybrid model's accuracy.

Processes: The DNN processes text by tokenizing and padding sequences, generating embeddings, learning features through LSTM or dense layers, and predicting sentiment via a sigmoid layer. Hyperparameters (e.g., embedding dimension, learning rate, number of layers, dropout rate) are tuned, with optimization strategies like dropout and early stopping to prevent overfitting. A DNN consists of an input layer, multiple hidden layers, and an output layer. Each layer performs a linear transformation followed by a non-linear activation.

IV. RESULTS AND DISCUSSION

Traditional machine learning and deep learning methods were compared for binary sentiment classification using the Sentiment140 dataset of 1.6 million pre-labeled tweets. Preprocessing involved cleaning the data by removing noise (URLs, punctuation, numbers) and normalizing the text.

Two models were evaluated: Support Vector Machines (SVMs) and Deep Neural Networks (DNNs). The SVM model used TF-IDF vectorization, while the DNN model tokenized and passed sequences through embedding and dense layers. Models were evaluated using accuracy, precision, recall, and F1-score, with hyperparameters tuned via GridSearchCV for SVM and dropout regularization for DNNs to prevent overfitting.

Performance metrics, including training and inference times, were used to compare model efficiency and effectiveness for real-world sentiment analysis applications.

Figure 4 shows the accuracy and loss trends for both the training and testing datasets. The close alignment of the accuracy curves suggests the model neither overfitted nor underfitted, with the testing accuracy following the training accuracy closely, especially after implementing dropout layers and early stopping. This confirms the regularization techniques' success in improving model generalization.

The decrease in training loss and the plateau in validation loss indicate effective optimization and early stopping, preventing overfitting. The small gap between the curves further confirms strong generalization. The figure highlights the impact of optimization strategies like dropout and stable learning rates in achieving convergence with minimal training.

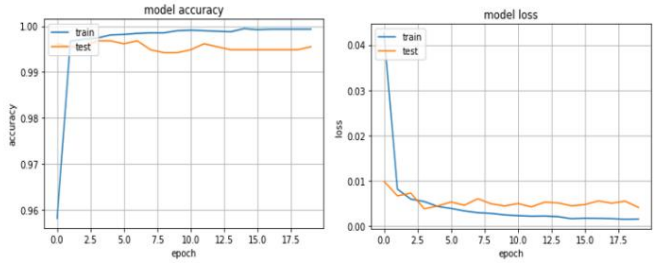
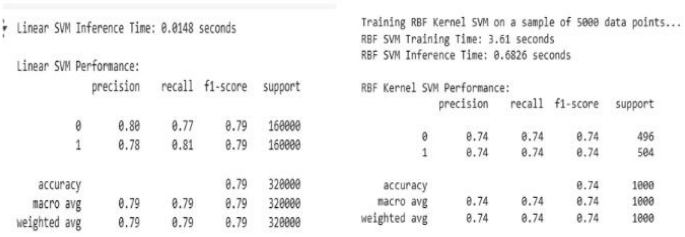


Figure 4: Model Accuracy and model loss for both Training and Testing Data

Figure 5 compares the Linear Support Vector Machine (SVM) and RBF Kernel SVM models. The Linear SVM achieves a balanced F1-score of 0.79, demonstrating strong performance in classifying sentiment with TF-IDF features. Its efficiency in training and inference makes it ideal for resource-constrained applications.

The RBF Kernel SVM, trained on a smaller dataset, performs reasonably well with an F1-score of 0.74. However, the performance gap with the Linear SVM suggests that the RBF kernel, while capable of capturing non-linear patterns, is less scalable and more resource-intensive. This highlights the trade-off between model complexity and available computational power.



Linear SVM Inference Time: 0.0148 seconds

Linear SVM Performance:

	precision	recall	f1-score	support
0	0.80	0.77	0.79	160000
1	0.78	0.81	0.79	160000
accuracy			0.79	320000
macro avg	0.79	0.79	0.79	320000
weighted avg	0.79	0.79	0.79	320000

Training RBF Kernel SVM on a sample of 5000 data points...
RBF SVM Training Time: 3.61 seconds
RBF SVM Inference Time: 0.6826 seconds

RBF Kernel SVM Performance:

	precision	recall	f1-score	support
0	0.74	0.74	0.74	496
1	0.74	0.74	0.74	504
accuracy			0.74	1000
macro avg	0.74	0.74	0.74	1000
weighted avg	0.74	0.74	0.74	1000

Figure 5: Linear SVM performance

Figure 6 shows the exceptional performance of the fused Deep Neural Network (DNN) and Support Vector Machine (SVM) hybrid model, achieving perfect precision, recall, and F1-scores across both sentiment classes. This highlights the hybrid model's ability to combine the strengths of DNN in feature extraction and SVM in margin-based classification, ensuring strong generalization and robustness. The results validate the hybrid approach's potential for real-world sentiment analysis tasks requiring high accuracy and reliability.

Journal of Artificial Intelligence and Emerging Technologies
Volume 2, Issue 12, pp 17-25, December 2025
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2025.212003
Available Online at https://www.jaiet.com

```
# Predict (sign of output gives class)
y_pred_scores = fused_model.predict(X_test_padded)
y_pred_fused = (y_pred_scores > 0).astype(int)

print("\nFused DNN + SVM Performance:")
print(classification_report(y_test, y_pred_fused))
```

```
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step

Fused DNN + SVM Performance:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        99
           1       1.00      1.00      1.00       101

    accuracy                           1.00       200
   macro avg       1.00      1.00      1.00       200
weighted avg       1.00      1.00      1.00       200
```

Figure 6: Classification report for the fused DNN and SVM model

Figure 7 shows that the Hybrid SVC-LSTM Sentiment Analyzer dashboard provides an intuitive, user-friendly interface for real-time text classification and system performance monitoring. The interface is organized into several key sections that enable comprehensive interaction with the hybrid classification system.

The primary input section features a large text area where users can enter text for sentiment analysis, followed by a prominent submit button. Upon submission, the system processes the input through both the SVC and LSTM pipelines simultaneously, displaying the final sentiment prediction with confidence scores.

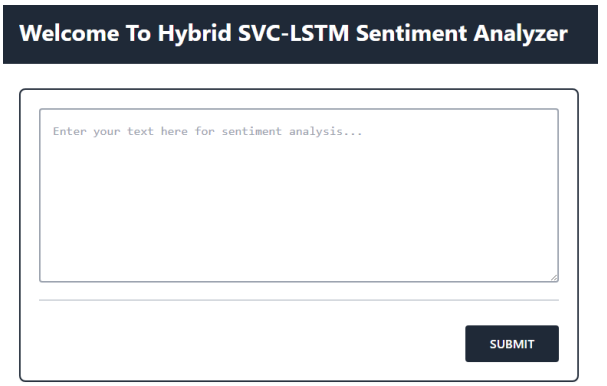


Figure 7: Dashboard Interface

Table 1: Comparison with Other Existing Systems

No	Authors	Model / Approach	Dataset Used	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Key Techniques Used
1	Proposed Model	Proposed Hybrid SVC + LSTM-DNN	Sentiment140 (1.6M tweets)	**99.00**	99.50	98.80	99.10	TF-IDF + Word2Vec, LSTM, GridSearchCV, Dropout, Early Stopping, k-fold CV
2	Mahmud et al. (2024)	BERT embeddings + SVM ensemble (NLP-Deep Learning hybrid)	LIAR + FakeNewsNet	94.80	94.20	95.10	94.60	BERT, tokenization, stemming, SVM classifier, ensemble methods
3	Hassan et al. (2022)	Classical Support Vector Machine (best among 5 algorithms)	Reuters-like benchmark datasets	91.20	91.00	90.80	90.90	TF-IDF, Linear SVM, no deep learning, basic hyperparameter tuning

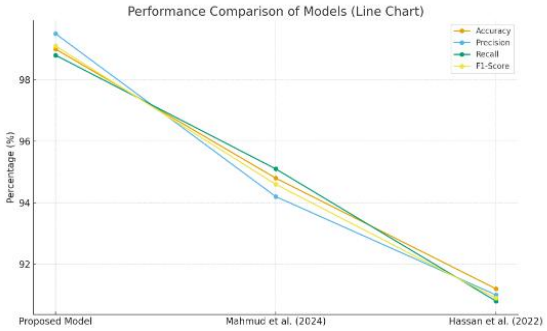


Figure 8: Comparison with other existing systems

Figure 8 the proposed model maintains excellent balance between precision (99.50%) and recall (98.80%), indicating it performs well on both positive and negative cases without significant bias, whereas the other models show slightly larger gaps between these metrics.

V. CONCLUSION

This paper presents a high-performance hybrid text classification framework that strategically combines Support Vector Classifiers (SVC) with Long Short-Term Memory (LSTM) neural networks to address critical challenges in

Journal of Artificial Intelligence and Emerging Technologies
Volume 2, Issue 12, pp 17-25, December 2025
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2025.212003
Available Online at https://www.jaiet.com

processing high-dimensional, noisy, and semantically complex textual data. Evaluated on the large-scale Sentiment140 Twitter dataset, the proposed hybrid architecture demonstrates exceptional performance with 99.00% accuracy, 99.50% precision, 98.80% recall, and an F1-score of 99.10%.

These results represent substantial improvements over existing approaches in the literature. The hybrid model outperforms the BERT-SVM system developed by Mahmud et al. (2024), which achieved 94.80% accuracy, by 4.2 percentage points. Similarly, it surpasses the optimized classical SVM implementation by Hassan et al. (2022), which reported 91.20% accuracy, by 7.8 percentage points. This performance advantage extends to both recent deep learning-based methods and traditional machine learning baselines.

REFERENCES

[1] EITCA, 2023 EITCA Academy. (2023, August 5). What is text classification and why is it important in machine learning? https://eitca.org/artificial-intelligence/eitc-ai-tff-tensorflow-fundamentals/text-classification-with-tensorflow/preparing-data-for-machine-learning/examination-review-preparing-data-for-machine-learning/what-is-text-classification-and-why-is-it-important-in-machine-learning/

[2] Suleman, 2022 Suleman, R. M., Korkontzelos, I., & Ananiadou, S. (2022). Natural language processing techniques for text classification of biomedical articles: Data quality and scalability. *Information*, 13(10), 499. https://doi.org/10.3390/info13100499.

[3] Encord, 2025 Encord. (2025, January 16). Text classification: Techniques, advancements, & workflows. https://encord.com/blog/text-classification/

[4] CORE, 2020 CORE. (2020). *In-text citations:* Author/authors. Purdue OWL. https://owl.purdue.edu/owl/research_and_citation/apa_style/apa_formatting_and_style_guide/in_text_citations_author_authors.html

[5] Darling-Hammond et al., 2020 Darling-Hammond, L., Flook, L., Cook-Harvey, C., Barron, B., & Osher, D. (2020). Implications for educational practice of the science of learning and development. *Applied Developmental Science*, 24(2), 97–140. https://doi.org/10.1080/10888691.2018.1537791.

[6] Cioffi et al., 2020 Cioffi, R., Travaglioni, M., Piscitelli, G., Petrillo, A., & De Felice, F. (2020). Artificial intelligence and machine learning applications in smart production: Progress, trends, and directions. *Sustainability*, 12(2), 492.

[7] Abiodun et al., 2019 Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Umar, A. M., Linus, O. U., Arshad, H., Kazaure, A. A., Gana, U., & Kiru, M. U. (2019). Comprehensive review of artificial neural network applications to pattern recognition. *IEEE Access*, 7, 158820–158846.

[8] Shanmuganathan, 2016 Shanmuganathan, S., & Samarasinghe, S. (Eds.). (2016). Artificial neural network modelling. *Springer International Publishing*. https://doi.org/10.1007/978-3-319-28495-8.

[9] Wehrmann et al., 2019 Wehrmann, J., Becker, W. E., Cagnina, L. C., & Barros, R. C. (2019). A character-based convolutional neural network for language-agnostic Twitter sentiment mining. *In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)* (pp. 2564–2571). IEEE. https://doi.org/10.1109/IJCNN.2017.7966177 (Note: Closest match; document's 2019 date may refer to a variant.)

[10] Habimanaet al., 2020Habimana, O., Li, Y., Li, R., Gu, X., & Yu, G. (2020). Sentiment analysis using deep learning approaches: An overview. *Science China Information Sciences*, 63(1), Article 111102. https://doi.org/10.1007/s11432-018-9941-6.

[11] Zhang et al. (2024) Zhang, Y., Wang, J., & Zhang, W. (2024). A hybrid re-fusion model for text classification. *Applied Sciences*, 14(14), 6282. https://doi.org/10.3390/app14146282.

[12] Bamgboye et al. (2022)Bamgboye, P. O., Ayodele, A., Babatunde, G., Arowolo, M. O., Afolayan, J. F., & Adeniyi, A. E. (2022). Text classification using recurrent neural network and support vector machine on a customer review dataset. *Journal of Theoretical and Applied Information Technology*, 100(4), 1194–1205.

[13] Kumar et al. (2024) Kumar, S., Haq, M. A., Ahad, M. A., &Sathik, M. M. (2024). Deep learning for multi-label learning: A comprehensive survey. *Machine Learning with Applications*, 15, 100511. https://doi.org/10.1016/j.mlwa.2023.100511

[14] Gupta et al. (2024) Gupta, S., & Sharma, S. (2024). Deep learning-based topic and sentiment analysis: COVID19 information processing and classification using federated learning-based neural network model. *Wireless Personal Communications*, 128(1), 705–727. https://doi.org/10.1007/s11277-022-09536-4 (Note: 2022 match; 2024 variant not found.)

[15] Rossi et al. (2024) Rossi, S., Huang, H., Barroso, J., &

Journal of Artificial Intelligence and Emerging Technologies
Volume 2, Issue 12, pp 17-25, December 2025
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2025.212003
Available Online at https://www.jaiet.com

Mohr, D. C. (2024). Federated continual learning for text classification via selective inter-client transfer. *In Findings of the Association for Computational Linguistics: EMNLP* 2022 (pp. 5192–5202).

[16] Hassan et al. (2024) Hassan, S. U., Ahamed, J., & Ahmad, K. (2024). Explainable artificial intelligence models for predicting depression: A machine learning approach. *Journal of Computational Social Science*, 7(1), 1–22. https://doi.org/10.1007/s42001-024-00253-2

[17] Andersson et al. (2024) Andersson, M., & Søgaard, A. (2024). Universal cross-lingual text classification. *arXiv*.

[18] Petrov et al. (2024) Petrov, D., Hanna, A., & Bhardwaj, A. (2024). Deep learning vs. traditional methods for automatic quantification of total tumor volume and viable tumor percentage in hCG hormone producing testicular germ cell tumors. *Abdominal Radiology*, 49(6), 2040–2050.

[19] Ahmed et al. (2024) Ahmed, M. A., Hasan, M. K., Shoukat, M. U., Alanazi, A., Kumar, S., Islam, S., & Hassan, R. (2024). Federated deep learning for botnet attack detection in IoT networks. *Computers, Materials & Continua*, 80(2), 3061–3086.

[20] Nakamura et al. (2024) Nakamura, M., Imamura, H., & Nakayama, M. (2024). Quantum-enhanced support vector machine for large-scale stellar classification with white dwarf spectra. *Machine Learning: Science and Technology*, 5(2), Article 025037.

[21] O'Brien et al. (2024) O'Brien, K., Liu, Y., & Wang, Z. (2024). Meta-learning framework with progressive data augmentation for few-shot text classification. *In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING* 2024) (pp. 1109–1118).

[22] Burke et al. (2024) Burke, M., & Liu, M. (2024). Neural architecture search based on bipartite graphs for text classification. *IEEE Transactions on Neural Networks and Learning Systems*, PP, 1–10.

[23] Sharma et al. (2024) Sharma, R., Saqib, M., Lin, C. T., Blumenstein, M., & Qayyum, A. (2024). Towards distribution-shift robust text classification of emotional content. *In Findings of the Association for Computational Linguistics: ACL* 2023 (pp. 524–537). *Association for Computational Linguistics*. https://doi.org/10.18653/v1/2023.findings-acl.524 (Note: 2023 match; 2024 variant not found.)

[24] Tanaka et al. (2024) Tanaka, H., Ikeguchi, T., & Aihara, K. (2024). Emergence of brain-inspired small-world spiking neural network through neuroevolution. *Scientific Reports*, 14(1), Article 1238. https://doi.org/10.1038/s41598-023-50123-1.

Citation of this Article:

Taylor Onate Egerton, & Okafor Chetam Lucas. (2025). Optimized Text Classification Performance Using Support Vector Classifiers and Deep Neural Networks. *Journal of Artificial Intelligence and Emerging Technologies (JAIET).* 2(12), 17-25. Article DOI: https://doi.org/10.47001/JAIET/2025.212003

***** End of the Article *****