Comparative Analysis of Machine Learning Models for IT Ticket Classification

Roberto Requejo Fernandez

Abstract: *This study develops and evaluates automated classification models for IT service tickets using Natural Language Processing techniques. A balanced dataset of 5,000 IT support tickets was preprocessed through tokenization, lemmatization, and stop word removal, then vectorized using Word2Vec and TF-IDF approaches. Five classification models were implemented to distinguish hardware-related from non-hardware tickets: Logistic Regression, Random Forest, Support Vector Machine (SVM), Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN).*

Results demonstrate that traditional machine learning models outperformed deep learning approaches for this binary classification task. Logistic Regression achieved the highest test accuracy at 87.70% (F1-score: 0.878), followed by SVM at 87.00% (F1-score: 0.873), and Random Forest at 85.40% (F1-score: 0.852). Deep learning models (LSTM and CNN) reached approximately 81% accuracy. The superior performance of conventional algorithms suggests they are more suitable for binary ticket classification with limited datasets, offering robust accuracy with lower computational requirements. However, it should be noted that traditional ML models used TF-IDF vectorization while deep learning models used Word2Vec embeddings, which may have influenced comparative performance. These findings provide practical guidance for implementing efficient automated ticket routing systems in IT service management environments.

1. Introduction

IT service management relies heavily on efficient ticket classification and routing to ensure timely resolution of technical issues. Manual ticket categorization is time-consuming, error-prone, and scales poorly with increasing ticket volumes. Automated classification using machine learning offers a promising solution, potentially reducing response times and improving resource allocation.

This study addresses the practical question: Which machine learning approach provides the best accuracy-efficiency trade-off for IT ticket classification in resource-constrained environments? By comparing five distinct models across traditional and deep learning paradigms, this research provides empirical evidence to guide implementation decisions in real-world IT service desks

2. State of the Art

The application of Natural Language Processing and machine learning to IT service ticket classification has gained significant research attention as organizations seek to automate IT service management processes.

2.1 Machine Learning Approaches for IT Ticket Classification

Zangari et al. (2023) examined ticket automation using transformer-based language models like BERT for hierarchical classification in *Expert Systems with Applications*. Tested on datasets with over 20,000 customer complaints and 35,000 bug reports, their Multi-Level BERT (ML-BERT) approach achieved 5.7% higher F1-scores and 5.4% better accuracy than traditional classifiers. The study demonstrated that contextualized language models significantly improve classification performance for hierarchically labeled tickets, though effectiveness depends on dataset characteristics and embedding strategies.

2.2 Comparative Analysis of Traditional and Deep Learning Models

Al-Hawari (2021) compared traditional machine learning algorithms (SVM, Random Forest, Naïve Bayes) against deep learning approaches for IT ticket classification in the Journal of King Saud University - Computer and Information Sciences. SVM and Random Forest achieved 82-89% accuracy on binary and multi-class tasks. Importantly, the research found that traditional algorithms often match or exceed deep learning performance on smaller datasets (under 10,000 instances), requiring less training data and computational resources while maintaining robust accuracy. This finding is particularly relevant for practical implementations where labeled data may be limited.

2.3 Research Gap and Contribution

While existing studies demonstrate the potential of both traditional and deep learning approaches, few provide systematic comparisons under controlled conditions with balanced datasets. The current work extends this research by systematically comparing five architectures—from Logistic Regression to CNN—on a balanced binary classification task with 5,000 IT support tickets, providing practical insights for organizations with moderate-sized ticket databases.

3. Methodology:

3.1 Dataset

The study utilized an IT service ticket dataset containing 47,837 instances across eight original categories (Hardware, Access, Miscellaneous, and others). To optimize computational efficiency and ensure balanced classification, the dataset was reduced to 5,000 instances with binary categorization: 2,500 hardware-related tickets and 2,500 non-hardware tickets. This balanced distribution prevents class bias and enables fair model evaluation.

Dataset Split:

- Training set: 4,000 instances (80%)
- Test set: 1,000 instances (20%)
- Stratification applied to maintain 50-50 class balance in both sets

3.2 Text Preprocessing

A comprehensive seven-step preprocessing pipeline was implemented to prepare ticket text for analysis:

1. Lowercasing: Converting all text to lowercase for uniformity
2. Contraction Expansion: Expanding contractions (e.g., "don't" → "do not") using the contractions library
3. Character Filtering: Removing non-alphanumeric characters and punctuation using regular expressions

4. Tokenization: Splitting text into individual words using NLTK's word_tokenize function
5. Stop Word Removal: Eliminating common English stop words that lack significant meaning
6. Lemmatization: Reducing words to their base forms using WordNetLemmatizer
7. Reconstruction: Joining processed tokens back into cleaned text strings

3.3 Feature Extraction

Two vectorization approaches were employed to convert preprocessed text into numerical representations:

- **Word2Vec Embeddings:** A pre-trained NLP model generating 300-dimensional dense semantic vectors capturing contextual word relationships and meanings. This approach was used for deep learning models (LSTM and CNN)
- **TF-IDF Vectorization:** Statistical approach computing term frequency-inverse document frequency vectors, emphasizing word importance across the dataset. This approach was used for traditional machine learning models (Logistic Regression, Random Forest, SVM).

Note on Methodological Design: The decision to use different vectorization methods reflects common practice in the field—TF-IDF typically pairs well with traditional ML algorithms, while Word2Vec embeddings are standard input for neural networks. However, this introduces a confounding variable in direct performance comparisons.

3.4 Dimensionality Reduction and Visualization

Additionally, t-SNE dimensionality reduction (2 components, perplexity=5) was performed for visualization, confirming meaningful separation between hardware and non-hardware ticket clusters.

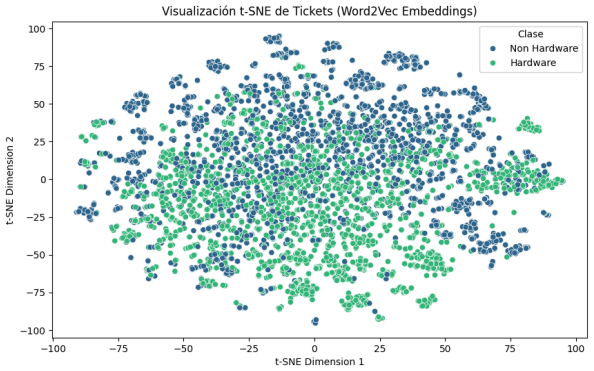

***Fig 1**. 2D t-SNE Visualization of data using Word2Vec vectors.*

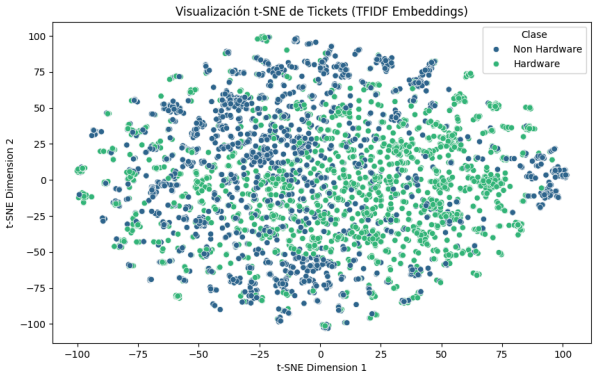

***Fig 2**. 2D t-SNE Visualization of data using TF-IDF vectors*

In the two data visualizations, we can see how the data arranged differently depending on the vectorization method. With Word2Vec embedding, a clearer pattern emerges with hardware tickets clustering more toward the bottom of the graph, while TF-IDF embedding shows less distinct separation. However, definitive conclusions cannot be drawn from these visualizations alone, as t-SNE is sensitive to hyperparameters and doesn't preserve global structure.

4. Classification Models

All models were evaluated using an 80-20 train-test split with stratification to maintain class balance. Five distinct models were implemented and evaluated:

4.1 Logistic Regression: Binary classifier using TF-IDF features with regularization

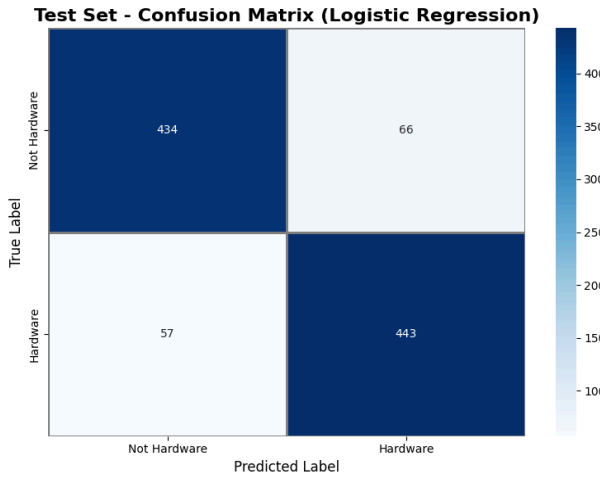

***Fig 3**. Confusion matrix of test set for Logistic Regression*

Accuracy	Precision	Recall	F1 Score
0.87	0.87	0.88	0.87

***Fig 4.** Evaluation metrics for test set in Logistic Regression*

Logistic Regression demonstrated strong performance with 434 true negatives and 443 true positives on the test set, with relatively low misclassification rates (66 false positives, 57 false negatives). The model showed minimal overfitting with consistent performance between training and test sets.

4.2 Random Forest:
Ensemble method with 100 decision tree estimators using TF-IDF features

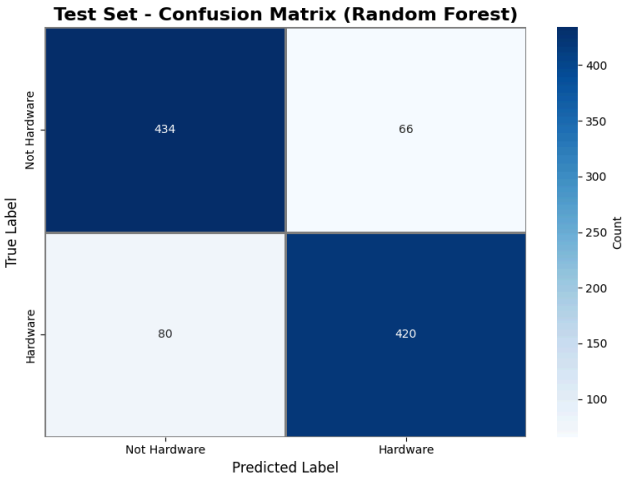

***Fig 5.** Confusion matrix of test set for Random Forest*

Accuracy	Precision	Recall	F1 Score
0.85	0.86	0.85	0.85

***Fig 6.** Evaluation metrics for test set in Random Forest*

Random Forest achieved perfect training accuracy (100%) but showed signs of overfitting, with test performance including 434 true negatives, 420 true positives, 66 false positives, and 80 false negatives. The gap between training and test accuracy suggests the model memorized training patterns rather than learning generalizable features.

4.3 Support Vector Machine (SVM):
RBF kernel classifier with TF-IDF features

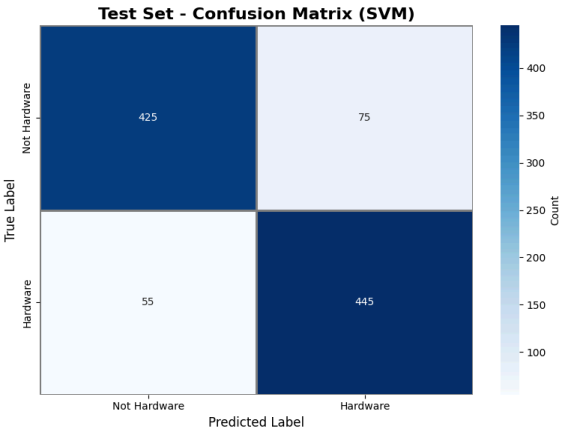

***Fig 7.** Confusion matrix of test set for Support Vector Machine*

Accuracy	Precision	Recall	F1 Score
0.87	0.85	0.89	0.87

Fig 8. *Evaluation metrics for test set in Support Vector Machine*

SVM exhibited excellent training performance (1972 and 1984 correct classifications) with balanced test results: 425 true negatives, 445 true positives, 75 false positives, and 55 false negatives.

4.4 LSTM Network

Bidirectional recurrent neural network with Word2Vec embeddings. Architecture: Bidirectional LSTM layers (128 and 64 units), Dense layers (64 and 32 units with ReLU activation), Dropout (0.3), output layer with softmax activation. Trained for 100 epochs with batch size 32, Adam optimizer, and early stopping.

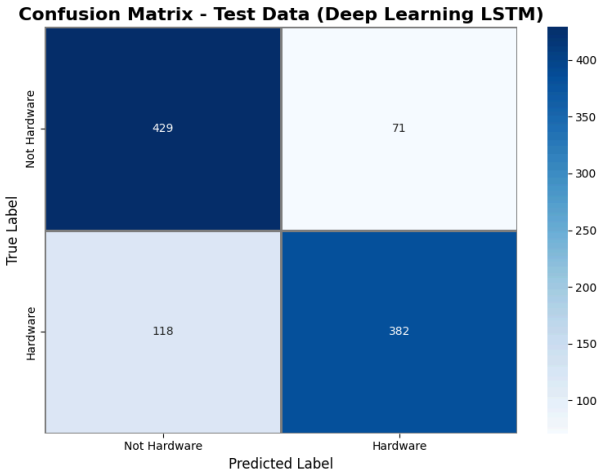

Fig 9. *Confusion matrix of test set for LSTM*

Accuracy	Precision	Recall	F1 Score
0.81	0.84	0.76	0.80

Fig 10. *Evaluation metrics for test set in LSTM*

LSTM showed the most conservative predictions with 429 true negatives and 382 true positives, along with 71 false positives and 118 false negatives. The model demonstrated lower recall, suggesting it was more cautious in predicting the positive class (hardware tickets).

4.5 CNN

Convolutional neural network with Word2Vec embeddings. Architecture: Three Conv1D layers (128, 256, 128 filters with kernel size 1), BatchNormalization, Dropout (0.3), GlobalMaxPooling1D, Dense layers (64 and 32 units), output layer with softmax. Trained for 100 epochs with batch size 32, Adam optimizer, and early stopping.

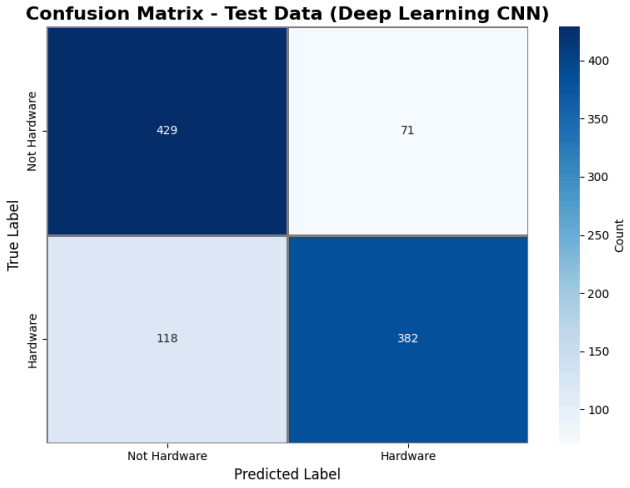

Fig 11. *Confusion matrix of test set for CNN*

Accuracy	Precision	Recall	F1 Score
0.81	0.84	0.76	0.80

Fig 12. *Evaluation metrics for test set in CNN*

CNN performed identically to LSTM with 429 true negatives, 382 true positives, 71 false positives, and 118 false negatives, suggesting both deep learning models faced similar challenges with the dataset size and feature representation.

5. Experiments & Results

5.1 Comparative Performance

The following table presents comprehensive performance metrics for all five models on the test dataset

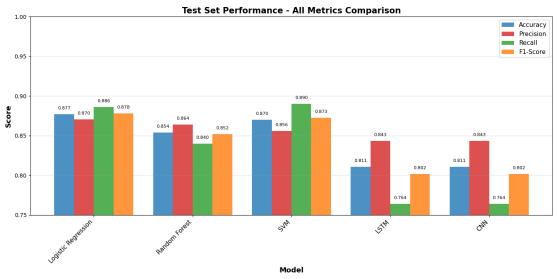

Fig 13. *Comparison of test metrics between models*

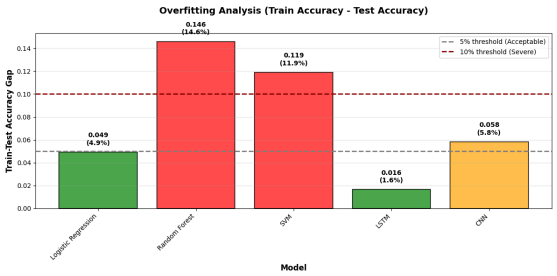

Fig 14. *Comparison of overfitting between models*

5.2 Key Findings

1. **Traditional ML Superiority**: Logistic Regression achieved the highest test accuracy (87.70%), outperforming all other models including deep learning approaches by 6-7 percentage points.
2. **Overfitting in Complex Models**: Random Forest demonstrated perfect training accuracy (100%) but lower test performance (85.40%), indicating significant overfitting despite still-respectable generalization. SVM also showed signs of overfitting, though less pronounced.
3. **Deep Learning Limitations**: Both LSTM (81.10%) and CNN (81.10%) underperformed compared to traditional algorithms, suggesting the dataset size and binary task complexity favor simpler models. The identical performance suggests both models converged to similar decision boundaries.
4. **Consistent Performance**: SVM (87.00%) showed nearly equivalent performance to Logistic Regression with excellent balance between precision (0.85) and recall (0.89), making it a strong alternative.
5. **Feature Representation Impact**: TF-IDF features (used by top-performing models) appeared more effective than Word2Vec embeddings (used by deep learning models) for this specific binary classification task, though this comparison is confounded by model architecture differences.

6. Discussion

6.1 Why Traditional ML Outperformed Deep Learning

Traditional ML models (Logistic Regression, SVM) outperformed deep learning models (LSTM, CNN) by 6-7% in classifying IT tickets, primarily due to:

1. **Dataset Size Constraints**: The 5,000-instance dataset was insufficient for deep learning models to learn complex patterns and leverage their capacity. Deep learning typically requires 10,000+ samples to demonstrate advantages.
2. **Task Complexity**: Binary classification (hardware vs. non-hardware) was simple enough for traditional algorithms to capture decision boundaries effectively

without requiring deep feature hierarchies.

3. **Feature Representation**: TF-IDF's statistical emphasis on discriminative terms worked better than Word2Vec's semantic embeddings for this specific task. Word2Vec captures semantic similarity (e.g., "mouse" and "keyboard" are similar), which may not be optimal for classification where discriminative features matter more.
4. **Model Efficiency**: Traditional algorithms converged faster and required less hyperparameter tuning, benefiting from decades of optimization research.

6.2 Overfitting Analysis

The overfitting observed in Random Forest (100% train vs. 85.4% test) and to a lesser extent in SVM suggests that ensemble methods' high capacity can be detrimental with limited data. Regularization through:

- Limiting tree depth or number of estimators (Random Forest)
- Stronger regularization parameters (SVM)
- Cross-validation for hyperparameter selection

could potentially improve generalization, though this was not extensively explored in the current study.

6.3 Practical Implications

For IT service desks with moderate-sized ticket databases:

- Start with Logistic Regression: Offers best accuracy-efficiency balance, fast training, interpretable coefficients for understanding important terms
- Consider SVM for slight performance boost: At the cost of interpretability and longer training times
- Avoid deep learning unless: Dataset exceeds 50,000 tickets, task involves multi-class hierarchical classification, or computational resources are abundant

6.4 Strengths

- Balanced dataset eliminates class bias concerns
- Comprehensive comparison across five diverse architectures
- Achieved strong 87.70% accuracy with efficient Logistic Regression
- Provides actionable guidance for resource-limited implementations
- Systematic preprocessing pipeline ensures reproducibility

6.5 Limitations

- Methodological Confound: Deep learning models used Word2Vec while traditional ML used TF-IDF, preventing pure model architecture comparison. Future work should test all models with both vectorization methods.
- Dataset Reduction: The reduction from 47,837 to 5,000 instances may have disproportionately handicapped deep learning models that benefit from larger datasets.
- Binary Simplification: Real-world IT tickets often require multi-class classification (8+ categories), which may yield different comparative results.
- Limited Hyperparameter Optimization: Deep models received basic tuning; exhaustive optimization might improve their performance.

- No Computational Cost Analysis: Training time, inference speed, and resource requirements were not formally compared, though anecdotally traditional ML was significantly faster.
- Temporal Validation Absent: Models were not tested on tickets from different time periods, which would better simulate production deployment.

7. Conclusion

This study systematically compared five machine learning architectures for binary IT ticket classification, revealing that traditional algorithms (Logistic Regression, SVM) achieve superior accuracy (87-87.7%) compared to deep learning approaches (81%) on a balanced 5,000-ticket dataset. The results strongly support the use of simpler models for moderate-sized IT service management implementations, offering robust performance with minimal computational overhead.

The primary limitation—different vectorization methods for different model types—suggests an important avenue for future research: comprehensive testing of all models with both TF-IDF and Word2Vec embeddings would isolate the impact of model architecture from feature representation

References:

Zangari, A., Marcuzzo, M., Schiavinato, M., Gasparetto, A., & Albarelli, A. (2023). Ticket automation: An insight into current research with applications to multi-level classification scenarios. *Expert Systems with Applications*, 225, 119984. https://doi.org/10.1016/j.eswa.2023.119984

Al-Hawari, M. A. (2021). A machine learning based help desk system for IT service management. *Journal of King Saud University - Computer and Information Sciences*, 33(6), 702-718. https://doi.org/10.1016/j.jksuci.2019.04.001