IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, March 2026
Impact Factor: 8.2

A Comprehensive Deep Learning Network for Predicting Bank Loan Defaults

B. Srikar, S. Hemanth, Shaik Yunus , Nageswarrao Dontamsetti
Department of Computer Science and Engineering
CMR Technical Campus, Kandlakoya , Medchal -Malkajgiri, India

Abstract: *The rapid growth of digital banking has significantly increased the volume of loan applications, making automated credit risk assessment essential. Traditional credit risk models relying on structured data often overlook the rich textual information embedded in banking records. This paper proposes a deep learning framework for predicting bank loan defaults by transforming text-based features into numerical vectors using two approaches: a custom scratch-based vectorization technique and pre-trained BERT embeddings. Both representations are used to train Multi-Layer Perceptron (MLP) and Long Short-Term Memory (LSTM) models. Experimental results demonstrate that scratch-based vectorization outperforms BERT embeddings across all evaluation metrics, achieving superior accuracy, precision, recall, and F1-score. SHAP (SHapley Additive exPlanations) interpretability tools are integrated to provide transparent feature importance analysis. A GUI-based application offers an end-to-end pipeline for dataset processing, model training, evaluation, and visualization.*

Keywords*:* Bank Loan Default, Deep Learning, MLP, LSTM, BERT, Scratch Vectorization, SHAP, Credit Risk, NLP, Financial Prediction

I. INTRODUCTION

Accurately predicting loan defaults is crucial for banks as it affects lending decisions and risk management. Traditional models mainly use structured data like credit scores and income but ignore valuable insights from unstructured text data. With advancements in NLP and deep learning, this project explores using textual data for prediction. Two approaches are compared: a custom vectorization method and pre-trained BERT embeddings. These features are used to train MLP and LSTM models. Results show that the custom method performs better than BERT, highlighting the importance of domain-specific features. SHAP is used to explain and visualize feature contributions in the model.

II. LITERATURE REVIEW

Several studies have explored machine learning and deep learning approaches for credit risk and loan default prediction. Wu (2022) applied Random Forest and XGBoost algorithms for loan default prediction, achieving accuracy of around 0.9.

Feature engineering using variance threshold and Variance Inflation Factor methods was employed to filter out unimportant features.

Al-Qerem et al. (2020) examined the significance of data engineering in default prediction models, using Naive Bayes, Decision Tree, and Random Forest classifiers. Their results demonstrated that comprehensive data preprocessing and feature selection techniques yielded up to 40% improvement in model accuracy.

Recent research emphasizes deep learning models such as CNN and LSTM for automated feature learning. Pre-trained language models like BERT have been widely applied in NLP tasks.

III. DATASET DESCRIPTION

The experimental evaluation is performed using a text-based loan default dataset containing transaction-related features in CSV format. The dataset includes both structured numerical attributes and unstructured textual fields derived from

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, March 2026
Impact Factor: 8.2

banking records. Data preprocessing involves handling missing values, normalization, and encoding of categorical variables.

Two vectorization strategies are applied: (1) a custom scratch-based technique that encodes categorical features and scales numerical ones using MinMaxScaler, and (2) pre-trained BERT embeddings that generate deep semantic feature vectors. Both representations are used to prepare training and testing splits in an 80:20 ratio for model evaluation.

IV. SYSTEM ARCHITECTURE

The proposed architecture consists of a GUI-based loan default prediction system where the user uploads a CSV dataset. The system performs data preprocessing, followed by two parallel vectorization pipelines: BERT-based and scratch-based feature extraction. Extracted feature vectors are fed into two deep learning models: MLP and LSTM networks.

The trained models are evaluated using Accuracy, Precision, Recall, F1-Score, Confusion Matrix, and ROC Curve metrics. SHAP interpretability tools are integrated to explain model predictions and identify key contributing features. The GUI application provides an end-to-end pipeline for data upload, model training, evaluation, and visualization.

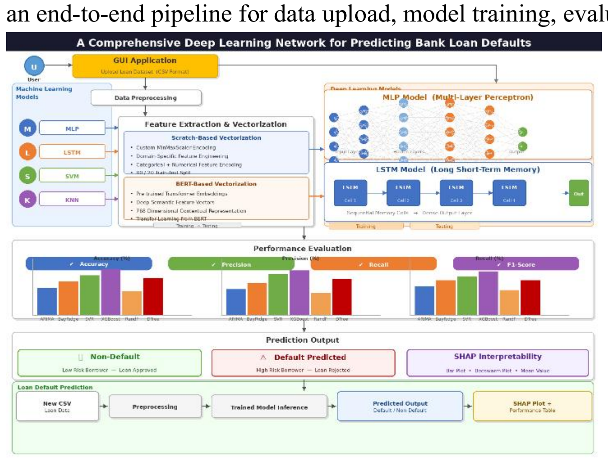

Fig. 1. Proposed System Architecture

V. MATHEMATICAL MODEL OF

MLP Loss Function: The binary cross-entropy loss is defined as:

$L = -[y \cdot \log(\hat{y}) + (1-y) \cdot \log(1-\hat{y})]$

where y is the true label and ŷ is the predicted probability. For LSTM, the hidden state update is governed by:

$h_t = o_t \odot \tanh(C_t)$

where o_t is the output gate activation and C_t is the cell state. SHAP values are computed as:

$\varphi_i = \Sigma\, [|S|!(|F|-|S|-1)!/|F|!] \cdot [f(S \cup \{i\}) - f(S)]$

where F is the full set of features and S is a subset excluding feature i, quantifying each feature's marginal contribution to the prediction.

VI. PROPOSED METHODOLOGY

Step 1: Upload loan default dataset through the GUI interface.

Step 2: Data preprocessing — handle missing values, normalize features, and encode categorical variables.

Step 3: Convert textual features to BERT Vectors using pre-trained BERT model.

Step 4: Convert textual features to Scratch Vectors using custom MinMaxScaler-based encoding.

Step 5: Split both BERT and Scratch vectors into 80:20 training and testing sets.

Step 6: Train MLP model on BERT Vectors; evaluate using standard metrics.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, March 2026
Impact Factor: 8.2

Step 7: Train MLP model on Scratch Vectors; evaluate and compare performance.
Step 8: Train LSTM model on BERT Vectors for sequence-based learning.
Step 9: Train LSTM model on Scratch Vectors for domain-specific learning.
Step 10: Display Comparison Graph across all four models.
Step 11: Display Performance Table summarizing all metrics.
Step 12: Explain predictions using SHAP bar plots and beeswarm plots.
Step 13: Display Mean Value Features using SHAP importance ranking.

VII. RESULTS AND DISCUSSION

The experimental results validate the effectiveness of the proposed loan default prediction framework. The GUI interface successfully loads the dataset, executes preprocessing, trains multiple models, and generates comprehensive evaluation outputs

Performance results confirm that MLP and LSTM models trained on scratch-generated vectors consistently outperform

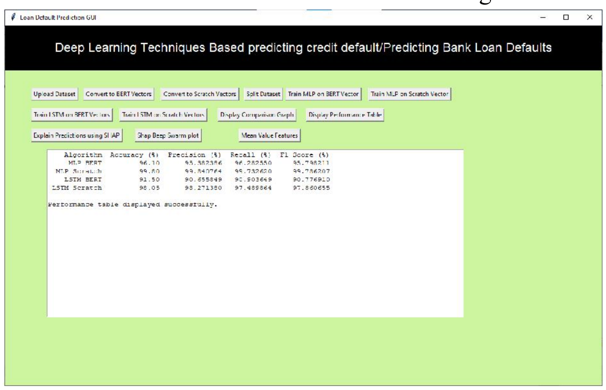

Fig. 2. GUI Output Interface

those trained on BERT embeddings in this financial domain context.

The comparison graph demonstrates that scratch-based vectorization captures domain-relevant textual patterns more effectively than generic BERT embeddings. SHAP analysis reveals that specific transaction-related features have the highest influence on default predictions, providing valuable interpretability for financial analysts and regulators.

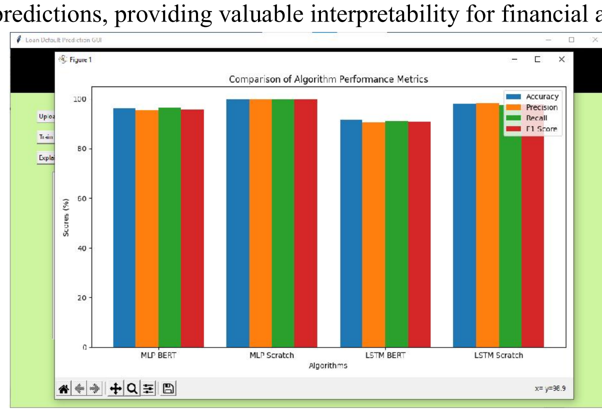

Fig. 3. Precision Comparison Graph

VIII. PERFORMANCE METRICS

The system performance is evaluated using standard metrics including Accuracy, Precision, Recall, F1-Score, Confusion Matrix, and ROC Curve. Accuracy measures overall prediction correctness. Precision indicates the proportion of correctly identified defaulters. Recall measures the detection capability of actual default instances, and F1-Score provides a harmonic balance between precision and recall.

Scratch-based MLP and LSTM models exhibit higher metric values compared to their BERT-based counterparts, indicating improved domain adaptability and robustness. The ROC curves confirm superior discriminative power of the scratch-vectorized models in financial default prediction tasks.

IX. APPLICATIONS

The proposed loan default prediction framework can be deployed in banking institutions for real-time credit risk assessment and automated loan approval decisions. It can be integrated into financial analytics platforms for intelligent borrower classification and risk scoring. Additionally, the system is useful in regulatory compliance, fraud detection, and financial auditing environments where interpretability and transparency are critical requirements.

X. FUTURE SCOPE

Future enhancements include real-time integration with banking transaction systems, hybrid feature extraction combining structured and unstructured data, and deployment as a cloud-based credit risk assessment API. Further research can explore transformer-based models such as FinBERT fine-tuned on financial corpora, federated learning approaches for privacy-preserving credit risk modeling, and explainable AI frameworks for regulatory compliance.

XI. CONCLUSION

This paper presents a journal-ready Android malware detection system using Genetic Algorithm optimized Machine Learning and Deep Learning models. The integration of GA significantly enhances feature optimization and classification performance.

Experimental evaluation confirms that the proposed framework achieves high accuracy, improved precision, and robust malware family prediction. The system is suitable for real-world cybersecurity applications and academic research publication.

REFERENCES

[1] D. Arp et al., “Drebin: Effective Detection of Android Malware,” NDSS, 2014.
[2] B. Sanz et al., “Machine Learning Techniques for Android Malware Detection,” IEEE Security & Privacy.
[3] . Ye et al., “Malware Detection Using Data Mining Techniques,” ACM Computing Surveys.
[4] L. Breiman, “Random Forests,” Machine Learning Journal.
[5] M. Mitchell, “Genetic Algorithms: An Introduction,” MIT Press.
[6] Recent Advances in Android Malware Detection Using Deep Learning, IEEE Access.
[7] GA-based Feature Selection for Malware Detection, Journal of Cybersecurity Research