IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A MACHINE LEARNING AND DEEP LEARNING FRAMEWORK FOR LOAN APPROVAL PREDICTION IN BANKING

[1] Ms. R. Deepthi, [2]M. Lakshmi

[1] Assistant Professor, [2] Assistant Professor

*Department of Computer Science and Engineering

Geethanjali Institute of Science and Technology (Autonomous), Nellore, Andhra Pradesh, India

Abstract: Loan approval and credit risk assessment are central tasks in the banking sector, where financial institutions must evaluate large volumes of applicant data while minimizing the risk of default. Traditional rule-based and manually evaluated approval processes are time-consuming, inconsistent, and prone to human bias, and they struggle to scale efficiently with the growing number of loan applications and with imbalanced approval/rejection data. This paper presents a hybrid Machine Learning (ML) and Deep Learning (DL) framework for automated loan approval prediction. The proposed system applies classical algorithms, namely Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN), alongside deep learning architectures including an Artificial Neural Network (ANN) and residual and densely connected convolutional architectures (ResNet and DenseNet) adapted for tabular financial data. Class imbalance in the dataset is addressed using the Synthetic Minority Over-sampling Technique (SMOTE), and an ensemble Voting Classifier combining Decision Tree and Random Forest predictions is used to improve robustness. The trained system is deployed through an interactive web-based interface that accepts applicant details and returns a real-time loan approval decision together with a risk probability score. On a dataset of 252,000 loan applications, the evaluated models achieved test accuracies exceeding 98% for DenseNet, above 96% for ResNet, above 95% for Random Forest, and above 85% for the Logistic Regression baseline, indicating that deep learning architectures, particularly DenseNet and ResNet, achieved the strongest predictive performance, with ensemble learning also providing a substantial improvement over the conventional linear baseline. The proposed framework reduces manual effort, improves consistency, and supports faster, data-driven decision-making in loan approval processes.

Index Terms - Loan Approval Prediction, Machine Learning, Deep Learning, Credit Risk, SMOTE, Random Forest, ResNet, DenseNet, Voting Classifier, Banking.

I. INTRODUCTION

Loan approval is one of the most critical processes in the banking sector. Financial institutions assess the eligibility of applicants based on factors such as income, credit history, employment status, and the requested loan amount. This process has traditionally been carried out manually by bank officials, which is time-consuming, vulnerable to human error, and can lead to inconsistent decisions across applicants.

With the advancement of Machine Learning (ML) and Deep Learning (DL), automated systems can be developed to improve both the accuracy and efficiency of loan approval prediction. Such systems analyze large volumes of historical applicant data, identify hidden patterns, and support data-driven decision-making rather than relying solely on manual judgment.

The growing number of loan applications received by banks has made manual evaluation increasingly difficult to sustain efficiently. Inconsistent or biased decisions can expose financial institutions to elevated default risk or unfairly reject eligible applicants. This motivates the development of an intelligent and automated system capable of assisting in accurate and unbiased loan eligibility assessment. Individual machine learning models, however, do not always generalize well on complex financial datasets, which motivates the use of ensemble techniques, such as a Voting Classifier, that combine the strengths of multiple base learners.

This paper proposes a hybrid ML and DL framework that integrates classical algorithms (SVM, KNN, Decision Tree, Random Forest, and Logistic Regression) with deep learning architectures (ANN, ResNet, and DenseNet) and ensemble voting to predict loan approval outcomes. The remainder of this paper is organized as follows: Section II reviews related work in loan and credit risk prediction; Section III describes the proposed methodology; Section IV presents implementation details; Section V discusses the experimental results; and Section VI concludes the paper with directions for future work.

II. LITERATURE SURVEY

A growing body of work has examined the use of ML and DL techniques for loan approval and credit risk prediction. Sayed et al. combined Random Forest with DenseNet and ResNet architectures alongside SMOTE-based data balancing for loan prediction in banking, reporting that hybrid ML-DL ensembles substantially improve classification performance compared with individual models [1]. Related work has explored optimized ensemble learning using Random Forest and XGBoost, showing that combining multiple classifiers reduces misclassification error and improves robustness relative to single models [2].

Other studies have focused specifically on the effect of data balancing on model performance. Thomas examined the use of SMOTE for handling imbalanced loan default datasets and reported improved recall and F1-score for the minority, high-risk class [3]. Hand investigated convolutional neural network-based deep learning models for financial risk prediction and found that they capture complex, non-linear relationships within financial data more effectively than conventional ML techniques [4]. Feature engineering combined with ensemble methods, including correlation-based feature selection and Random Forest importance ranking, has also been shown to improve model stability and prediction accuracy in loan prediction tasks [5].

Additional studies have applied Logistic Regression and Decision Tree models to applicant attributes such as income, credit history, and employment status for initial loan screening [6], used gradient boosting techniques such as XGBoost and LightGBM for credit risk analysis based on borrower repayment history [7], and applied Artificial Neural Networks to borrower information including credit score and repayment history, reporting higher prediction accuracy than traditional ML algorithms on large and complex datasets [8].

Despite these advances, many existing systems continue to rely on manual feature engineering, struggle with imbalanced approval/rejection data, and offer limited scalability for real-time decision-making. Table 1 summarizes representative approaches commonly used in existing loan approval prediction systems along with their key limitations.

Table 1: Existing Loan Approval Prediction Approaches

Model	Description	Limitations
Logistic Regression	Linear baseline classifier that predicts approval or rejection from applicant attributes	Limited ability to capture non-linear relationships in financial data
Decision Tree	Rule-based model that classifies applicants through sequential attribute splits	Prone to overfitting, especially on imbalanced datasets
Support Vector Machine (SVM)	Separates approved and rejected classes using a maximum-margin hyperplane	Computationally expensive on large, high-dimensional datasets
Random Forest	Ensemble of decision trees used to improve classification stability	Requires substantial manual feature engineering for best performance
Gradient Boosting (e.g., XGBoost)	Sequentially built ensemble models used for credit risk scoring	Sensitive to hyperparameter tuning; limited real-time scalability

The proposed framework addresses these gaps through SMOTE-based class balancing, automatic feature extraction using deep learning, and ensemble voting, as described in Section III.

III. PROPOSED METHODOLOGY

The proposed system predicts loan approval outcomes using a hybrid framework that combines classical Machine Learning algorithms, Deep Learning architectures, and ensemble voting. The methodology proceeds through data collection, data preprocessing, feature selection, data splitting, model training, and decision-making stages, as illustrated by the overall system architecture in Fig. 1, in which applicant data is collected, preprocessed (dimensionality reduction, categorical encoding, feature scaling, and class balancing), used to optimize model parameters via the Adam optimizer, and passed through the trained ML and DL classifiers to produce an approved or rejected outcome.

A. Data Collection and Dataset Description

The dataset used in this work consists of the financial and personal details of loan applicants, comprising 252,000 records described by 11 input features and a binary target label indicating loan risk status. Table 2 summarizes the key attributes used.

Table 2: Dataset Attributes Used

Attribute	Data Type	Description
Income	Numerical	Annual income of the applicant
Age	Numerical	Age of the applicant
Experience	Numerical	Total years of work experience
Profession	Categorical	Applicant's occupation (e.g., Software Developer, Civil Servant, Physician)
Married/Single	Categorical	Marital status of the applicant
House Ownership	Categorical	Housing situation: rented, owned, or norent/noown
Car Ownership	Categorical	Indicates whether the applicant owns a car
City / State	Categorical	Applicant's city and state of residence
Current Job Years	Numerical	Years spent at the current employer

Attribute	Data Type	Description
Current House Years	Numerical	Years spent at the current residence
Risk Flag (Target)	Categorical	Indicates loan approval / low risk or rejection / high risk

B. Data Preprocessing

Several preprocessing steps are applied to improve data quality prior to model training. Missing values are handled using mean, median, or mode imputation, and duplicate or irrelevant records are removed to prevent skewed results. Categorical attributes, such as profession, marital status, and house and car ownership, are converted into numerical form using label encoding and one-hot encoding. Numerical features are normalized so that all attributes are scaled consistently. Finally, class imbalance between approved and rejected (or high-risk) loan cases is addressed using the Synthetic Minority Over-sampling Technique (SMOTE), which generates synthetic samples for the minority class to prevent biased learning toward the majority class.

C. Feature Selection

Feature selection identifies the input attributes (X) and target label (y) that are most relevant to predicting loan approval. Correlation analysis is used to remove redundant attributes, while feature importance derived from the Random Forest model is used to retain the most influential variables, such as income, credit history-related attributes, employment stability, and asset ownership. This reduces model complexity and improves overall performance.

D. Data Splitting

The preprocessed dataset is divided into training and testing sets using an 80:20 split, corresponding to 201,600 records used for training the models and 50,400 records reserved for evaluating performance on unseen data. Stratified splitting is applied to preserve the relative proportion of approved and rejected classes across both sets.

E. Model Architecture

Multiple classical Machine Learning algorithms are implemented, including Logistic Regression as a linear baseline, Decision Tree and Random Forest for rule-based and ensemble classification, Support Vector Machine (SVM) for high-dimensional separation, and K-Nearest Neighbors (KNN) for instance-based classification. In parallel, Deep Learning models are implemented to automatically learn complex, non-linear patterns in the applicant data: an Artificial Neural Network (ANN) using multiple fully connected layers, a ResNet-style architecture that applies one-dimensional convolutions with residual (skip) connections to stabilize training and improve gradient flow, and a DenseNet-style architecture that applies densely connected one-dimensional convolutional blocks with batch normalization and dropout to encourage feature reuse. To further improve robustness, a Voting Classifier ensemble combines the predictions of the Decision Tree and Random Forest models. In hard voting, the output class with the majority of individual votes is selected as the final prediction; in soft voting, the class with the highest average predicted probability across the combined models is selected. Table 3 summarizes the key hyperparameters used during model training.

Table 3: Hyperparameters Used

Parameter	Description	Model Component
Learning Rate	Step size for the Adam optimizer (0.001)	ANN / ResNet / DenseNet
Batch Size	Number of samples processed per training iteration (32)	Deep Learning Models
Epochs	Maximum training iterations, with early stopping (up to 50)	Deep Learning Models
Train-Test Split	Proportion of data used for training versus testing (80:20)	All Models
Voting Strategy	Hard or soft voting across base classifiers	Voting Classifier (DT + RF)
Number of Trees	Number of estimators used in the ensemble (100)	Random Forest

F. Loss Function and Optimization

The Binary Cross-Entropy loss function is used to measure the difference between predicted and actual loan risk labels during training, as it is well suited to binary classification tasks. The Adam optimizer is used to update model weights, combining adaptive learning rates with momentum to accelerate convergence while maintaining training stability. Early stopping is applied to the deep learning models to halt training when validation loss ceases to improve, thereby reducing the risk of overfitting.

IV. IMPLEMENTATION

A. Software Requirements

The system is implemented using the Python programming language, which provides extensive support for data analysis and machine learning. Pandas and NumPy are used for data manipulation and numerical computation, Scikit-learn is used for classical ML model implementation and evaluation, and TensorFlow with Keras is used to build and train the deep learning models (ANN, ResNet, and DenseNet). Matplotlib and Plotly are used for data visualization. The trained models are deployed through an interactive web-based interface built with the Streamlit framework, with SQLite used for lightweight data storage. Visual Studio Code is used as the development environment, and the system is designed to run on a standard 64-bit Windows machine with a minimum of 4 GB RAM (8 GB recommended) and an Intel Core i3 or higher processor.

B. System Architecture

The system architecture consists of sequential components beginning with the banking customer dataset as the data input layer, followed by a data analysis stage and a data preprocessing module responsible for dimensionality reduction, categorical encoding, feature scaling, and handling of imbalanced classes. The preprocessed data feeds into a model optimization stage using the Adam optimizer, after which the AI model classification stage applies both Deep Learning and Machine Learning models to generate the final approved or rejected outcome. At the application level, the system follows a layered flow comprising a user interface module for collecting applicant input and displaying results, a data preprocessing module for cleaning, encoding, and scaling input data, a model training and persistence module, a prediction module that loads the trained models and estimates risk, and an evaluation module that reports accuracy and confusion-matrix-based metrics.

C. Model Implementation

The trained Random Forest, Logistic Regression, ResNet, and DenseNet models are loaded by the application at runtime through a model-loading utility. When an applicant submits their details, financial information such as income, age, and experience, personal details such as profession, marital status, and house ownership, and asset information such as car ownership and years in the current job and residence are assembled into a single input record. This record is passed through the same preprocessing pipeline used during training (encoding and scaling) before inference. For the deep learning models, the processed input is reshaped to match the expected one-dimensional convolutional input shape; for the classical ML models, the processed input is passed directly to the trained classifier. A decision threshold of 0.5 is applied to the predicted probability of default to classify the application as low risk (approved) or high risk (rejected). The interface then displays the predicted outcome together with a probability score and a risk-score gauge to support interpretability.

D. Testing

A multi-level testing strategy is used to validate the system. Unit testing verifies individual components, including dataset loading, data preprocessing, feature selection, and model loading. Integration testing checks that these components interact correctly, for example that preprocessed data flows correctly into model training and that trained models can be reloaded for inference. System testing evaluates the complete end-to-end pipeline, from applicant input to final prediction output, while acceptance testing confirms that the system meets the functional requirements expected by end users. Table 4 summarizes representative test cases exercised during this process.

Table 4: Test Cases and Results

Test ID	Test Scenario	Expected Result	Status
TC_01	Dataset Loading	Required columns present and data loads without error	Pass
TC_02	Data Cleaning	Duplicate and irrelevant records removed	Pass
TC_03	Label Encoding	Categorical attributes converted to numerical form	Pass
TC_04	SMOTE Balancing	Minority class samples synthetically generated	Pass
TC_05	Train-Test Split	80:20 split completed without data loss	Pass
TC_06	Model Training	Training loss decreases across epochs for DL models	Pass
TC_07	Loan Prediction	Approved/Rejected outcome generated with probability score	Pass

V. RESULTS AND DISCUSSION

The proposed framework is evaluated on the loan applicant dataset of 252,000 records using accuracy as the primary performance metric, consistent with the metrics reported by the deployed system. The preprocessing pipeline successfully removes duplicate records, encodes categorical attributes, normalizes numerical features, and balances the dataset using SMOTE; these stages were validated through the unit tests summarized in Table 4, all of which passed, confirming that the dataset was correctly prepared for model training.

Models were trained on 80% of the dataset (201,600 records) and evaluated on the remaining 20% (50,400 records). Table 5 reports the test accuracy obtained for each of the four models exercised through the deployed prediction interface.

Table 5: Model Performance Comparison (Test Accuracy)

Model	Type	Accuracy (%)
DenseNet	Deep Learning (Densely Connected 1D-CNN)	98%+
ResNet	Deep Learning (Residual 1D-CNN)	96%+
Random Forest	Ensemble Machine Learning	95%+
Logistic Regression	Linear Baseline	85%+

Among the evaluated models, the DenseNet-based deep learning architecture achieved the highest measured test accuracy, exceeding 98%, demonstrating the effectiveness of its dense connectivity pattern and feature reuse in capturing complex, non-linear relationships within the applicant data. The ResNet-based model also performed strongly, achieving an accuracy above 96%, benefiting from residual connections that stabilize training and support deeper feature extraction. The Random Forest ensemble achieved a closely comparable accuracy above 95%, confirming that a well-tuned ensemble of decision trees combined with SMOTE-based class balancing remains a highly competitive approach for this tabular financial dataset, while additionally offering interpretability through feature importance ranking. As expected, the Logistic Regression baseline recorded the lowest accuracy, above 85%, confirming that while a purely linear decision boundary captures much of the underlying signal in the data, it is less able to model the complex relationships between income, employment stability, asset ownership, and loan risk than the ensemble and deep learning approaches.

These results confirm that deep learning architectures, particularly DenseNet and ResNet, deliver the strongest predictive performance for loan risk classification on this dataset, with the Random Forest ensemble providing a closely competitive, more interpretable alternative. The deployed Streamlit-based interface further validates the system functionally, allowing a user to enter applicant details, select among the four trained models, and receive an immediate approved or rejected decision along with a probability-based risk score, demonstrating the practical applicability of the proposed framework for real-time loan screening.

VI. CONCLUSION

This paper presented a hybrid Machine Learning and Deep Learning framework for loan approval prediction in banking. The system combines classical algorithms such as Logistic Regression, Decision Tree, Random Forest, SVM, and KNN with deep learning architectures including ANN, ResNet, and DenseNet, supported by SMOTE-based class balancing and an ensemble Voting Classifier to improve robustness against imbalanced applicant data. Experimental evaluation on a dataset of 252,000 loan applications showed that the DenseNet-based deep learning model achieved the highest test accuracy, exceeding 98%, followed by the ResNet-based model above 96% and the Random Forest ensemble above 95%, with the Logistic Regression baseline achieving a lower accuracy above 85%, confirming that deep learning architectures, supported by ensemble learning, offer a meaningful improvement over conventional linear classification for this task.

The developed system contributes to reducing manual effort, inconsistency, and bias in the loan approval process by enabling automated, data-driven decision-making through a real-time, web-based prediction interface. Future work will focus on integrating real-time banking data, applying explainable AI techniques such as SHAP and LIME to improve the transparency of model decisions, extending deployment to mobile and cloud-based platforms for greater scalability, and incorporating additional features such as credit score analysis and fraud detection to make the system more comprehensive for real-world banking applications.

REFERENCES

[1] E. H. Sayed, A. Alabrah, K. H. Rahouma, M. Zohaib, and R. M. Badry, "Machine Learning and Deep Learning for Loan Prediction in Banking: Exploring Ensemble Methods and Data Balancing," IEEE Access, vol. 12, pp. 193997–194010, 2024.

[2] "Optimized Ensemble Learning for Loan Approval Prediction," Mathematics (MDPI), vol. 12, no. 21, Art. 3423, 2024.

[3] L. C. Thomas, "Machine Learning Models with Data Balancing Techniques for Loan Default Prediction," Indian Journal of Computer Science, 2023.

[4] D. J. Hand, "Deep Learning Models for Financial Risk Prediction," 2022.

[5] "Feature Engineering and Ensemble Techniques in Loan Prediction," PeerJ Computer Science, 2021.

[6] "Loan Approval Prediction using Logistic Regression and Decision Tree," IEEE Xplore, n.d.

[7] "Credit Risk Analysis using Gradient Boosting Techniques," ScienceDirect, n.d.

[8] "Loan Prediction using Artificial Neural Networks and Deep Learning," ScienceDirect, n.d.

[9] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.

[10] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Proc. IEEE CVPR, 2016.

[12] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," Proc. IEEE CVPR, 2017.

[13] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

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

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

[16] T. Fawcett, "An Introduction to ROC Analysis," Pattern Recognition Letters, vol. 27, no. 8, pp. 861–874, 2006.

[17] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proc. 22nd ACM SIGKDD Conf., 2016.

[18] S. Lundberg and S. Lee, "A Unified Approach to Interpreting Model Predictions," Advances in Neural Information Processing Systems, 2017.

[19] Kaggle, "Loan Prediction Dataset." [Online]. Available: https://www.kaggle.com/