Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

An Explainable and Robust Loan Approval Prediction System using Hybrid XGBoost and Heuristic Risk Assessment

Viral H. Shah
Dharmsinh Desai University
Gujarat

Shivam V. Bhat
Dharmsinh Desai University
Gujarat

Vedant D. Sharma
Dharmsinh Desai University
Gujarat

***Abstract* - Loan approval is a key financial process that affects both banks and people applying for loans. Old ways of checking loans depend a lot on manual checks and strict rules, which can lead to slow processes, unfair decisions, and expensive operations. This paper introduces a new system that uses data and smart technology to predict loan approvals. The system uses Extreme Gradient Boosting, or XGBoost, along with a combination of decision tools. The system has three main parts: (1) automatically checking documents with OCR to confirm what applicants say, (2) using a calibrated XGBoost model to estimate the chance of risk, and (3) applying penalty rules based on financial numbers. Testing this system on real loan data shows it can predict approvals with 98% accuracy. It also improves how well risk is measured because of the calibration. Compared to traditional methods like Random Forest and Logistic Regression, this new approach is better at making accurate predictions and keeping operations safe.**

***Index Terms* - Loan Approval Prediction, XGBoost, Credit Risk Assessment, Probability Calibration, Hybrid AI, OCR Verification, FinTech**

I. INTRODUCTION

The financial lending industry is changing a lot, moving away from old ways that rely heavily on paperwork and manual checks to newer methods that use data and automation. Loan approval systems are important because they help control credit risk, making sure lenders can grow their loan portfolios without ending up with too many loans that aren't paid back (called NPAs). Traditional methods often have problems. Some are too strict, turning down good borrowers because of outdated rules. Others are too guesswork-based, depending on human judgment which can be influenced by personal biases.

Machine Learning is becoming a big help in credit scoring. It can look at complicated connections between things like a borrower's age, income, and payment history. Among ML techniques, methods like Gradient Boosting Machines, especially XGBoost, are now widely used. These models work well with financial data and are easier to understand than other types of models that are hard to explain.

However, using machine learning models in important financial settings brings up special difficulties. A simple probability score from a model is usually not enough for actual use. Real-world systems need:

1) Reliability: The predicted probability should show the real chance of default (calibration).
2) Verification: Input data needs to be checked against the supporting documents to stop fraud from happening.
3) Safety: Deterministic guardrails must exist to catch edge cases that pure ML might miss.
4) Explainability: Decisions must be interpretable to satisfy regulatory requirements (e.g., GDPR).

This paper suggests a complete, end-to-end loan approval system that tackles these issues. Different from typical classification studies that only pay attention to accuracy, we present a **Hybrid Inference Engine** that merges the strong prediction abilities of XGBoost with a rule-based penalty system and an OCR-based document check module.

The major contributions of this work are:

- A robust preprocessing pipeline handling schema alignment, automated feature filtering, and categorical encoding.
- A calibrated XGBoost model optimized for imbalanced financial data using scale weighting.
- A novel risk assessment layer that adjusts ML confidence scores based on financial heuristics (e.g., Debt-to-Income ratios).
- Integration of an OCR-based verification mechanism to validate self-reported income against bank statements.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

- A comparative ablation study demonstrating the superiority of the proposed hybrid architecture over standalone ML models.

The rest of this paper is structured like this: Section II looks at related work. Section III explains the system's design. Section IV covers the dataset and how it's prepared. Section V includes the mathematical models. Section VI talks about the combined decision-making approach. Section VII goes over the results from the experiments, and Section VIII wraps up the study.

II. Literature Review

A. Traditional vs. Modern Approaches

Credit risk assessment has changed a lot over the past few decades. In the 1960s to 1990s, people mostly used statistical methods like Linear Discriminant Analysis and Logistic Regression. These models were easy to understand and implement, but they couldn't handle complex relationships between different factors, such as how age and income might interact to affect the risk of default.

In the 2000s, the introduction of ensemble learning changed things. Breiman developed Random Forests, which helped reduce errors by using a technique called bagging. Still, boosting methods, which improve predictions by fixing mistakes made by simpler models, turned out to be better for classifying credit risk using tabular data. A detailed comparison of different classification techniques for credit scoring showed that Gradient Boosted Trees performed the best across various financial datasets.

B. Explainability and Fairness

In recent years, from 2018 to 2024, there has been a growing emphasis on Explainable AI, or XAI. Laws such as the GDPR and the Equal Credit Opportunity Act require lenders to give clear explanations, known as "adverse action" notices, when a loan is denied. Research conducted by Lundberg and others on SHAP, which stands for SHapley Additive exPlanations, has become a common method for understanding how tree-based models work. This approach helps developers see both the overall and individual impact of different features in their models.

C. The Calibration Gap

A key but frequently ignored topic in current research is probability calibration. Many studies focus on Accuracy or AUC-ROC without checking whether the predicted probabilities are reliable. Guo et al. [6] pointed out that modern neural networks and boosted trees often produce unreliable probability estimates. In the context of financial lending, a predicted probability of 0.6 should clearly mean a 60% chance of repayment to correctly set interest rates. Our approach directly tackles this issue by using Platt Scaling as part of the process.

III. System Architecture

The proposed system goes beyond just a classification model and includes a complete inference pipeline built for use in real-world settings. The design is made up of three separate parts: Data Ingestion, Feature Engineering, and the Hybrid Decision Layer.

A. Data Ingestion and Verification Layer

The system accepts two distinct streams of input:

1) Structured Form Data: Self-reported attributes provided by the applicant (e.g., Income, Loan Amount, Term).
2) Unstructured Documents: PDF documents such as bank statements and salary slips.

To address the high error rates common in opensource OCR tools when parsing complex financial templates, this system integrates the Veryfi OCR API for enterprise-grade document intelligence. Unlike traditional template-based extraction, Veryfi utilizes a pretrained deep learning model optimized specifically for semi-structured financial documents.

The verification process follows a rigorous logic:

- Secure Ingestion: Documents are transmitted via TLS 1.2+ to the OCR endpoint, which returns a structured JSON payload containing key entities with confidence scores.
- Normalization: Raw string outputs undergo sanitization to remove OCR artifacts (e.g., currency symbols, whitespace) before comparison.
- Fuzzy Matching Protocol: To validate selfreported income against the OCR-extracted data, we utilize the Levenshtein Distance algorithm. This handles minor transcription errors that strict string equality would reject.

The similarity score S is calculated as:

$$S = 1 - \frac{dist(I_{user}, I_{ocr})}{\max(|I_{user}|, |I_{ocr}|)} \quad (1)$$

Where *dist* represents the Levenshtein edit distance. A verification flag is raised only if the similarity score drops below a strict threshold ($S < 0.8$), indicating a discrepancy greater than 20%. This ensures robust fraud detection while preventing false positives from minor digitization artifacts.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

B. Feature Engineering Layer

Raw data is transformed into a rigorous 12dimensional feature vector. This layer handles missing value imputation, categorical encoding, and schema alignment to ensure the inference vector strictly matches the training signature.

C. Hybrid Decision Layer

This is the core innovation of our system. It consists of:

- ML Model: An XGBoost classifier generating a base probability (P_{base}).
- Calibration: A Logistic Regression scaler transforming P_{base} to P_{calib}.
- Heuristic Engine: A rule-based system that applies penalties (ΔP) based on institutional risk policies.

IV. Dataset and Preprocessing

A. Dataset Description

The study uses the Kaggle Loan Approval Prediction dataset, which includes organized historical records of people who applied for loans. The dataset has a combination of information about people's demographics, their financial situation, and their behavior.

TABLE I
KEY FEATURES AND DATA TYPES

'eature Name	pe	le
o_of_dependents	eger	mographic
ducation	nary	cio-economic
elf_employed	nary	nployment Risk
ncome_annum	ntinuous	payment Capacity
oan_amount	ntinuous	edit Exposure
oan_term	ntinuous	ration Risk
ibil_score	ntinuous	storical Behavior
esidential_assets	ntinuous	llateral
ommercial_assets	ntinuous	llateral

B. Preprocessing Pipeline

Robust preprocessing is essential for model stability. We implemented a multi-stage pipeline.

1) Synthetic Minority Oversampling (SMOTE): To address class imbalance beyond simple weighting, we generate synthetic samples for the minority class. For a minority instance x_i, we select a k-nearest neighbor x_{zi} and interpolate:

$$x_{new} = x_i + \delta \cdot (x_{zi} - x_i) \quad (2)$$

Where $\delta \sim U(0,1)$ is a random number, ensuring the decision boundary is generalized rather than just replicated.

2) Cardinality Reduction: Features that have a very high number of different values, like unique Transaction IDs, or no variation at all, such as constant columns, are automatically removed. We use a variance threshold to do this.

$$\text{Drop } X_j \text{ if } \text{Var}(X_j) < \epsilon$$

This step reduces noise and prevents the model from overfitting to irrelevant identifiers.

3) Encoding Strategy: Categorical variables are handled with Label Encoding. Although One-Hot Encoding is commonly used, Label Encoding works well with tree-based models such as XGBoost, as these models can effectively make decisions based on ordered integer values.

$$\text{Education} \rightarrow \{\text{Graduate} : 1, \text{Not Graduate} : 0\}$$

4) Inference Schema Alignment: A common problem in production is called "Schema Skew," which happens when the data used for making predictions doesn't have all the columns that were present when the model was trained. Our system has a strict rule that makes sure the data columns match exactly what was used during training. It rearranges the columns to fit the training format, fills in any missing features with zeros, and removes any columns that weren't part of the original training data.

V. Mathematical Modeling *A.*

XGBoost Formulation

XGBoost (Extreme Gradient Boosting) is an ensemble technique that aggregates predictions from K decision trees.

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in F$$

To learn the set of functions f_k, XGBoost minimizes the following regularized objective at step t:

$$L^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t)$$

Where l is the loss function (LogLoss for classification) and Ω is the regularization term penalizing model complexity:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda ||w||^2$$

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

Here, T is the number of leaves and w are the leaf weights. This regularization is crucial for preventing overfitting on smaller financial datasets.

B. Handling Class Imbalance

Loan approval data sets often have an unequal number of approvals and rejections, with one being more common than the other, depending on the institution. To address this, we use a weighting method where the samples in the positive class are adjusted by a specific factor.

$$w_{pos} = \frac{\text{Count}_{negative}}{\text{Count}_{positive}}$$

This ensures the gradient updates are balanced, preventing the model from biasing towards the majority class.

C. Probability Calibration (Platt Scaling)

Tree-based models tend to make probabilities cluster closer to 0 or 1. To get more accurate probability estimates, we use another model called Logistic Regression, which is trained on the results from the XGBoost classifier. Let z_i be the log-odds output from XGBoost. The calibrated probability P_{calib} is:

$$P_{calib}(y = 1|z_i) = \frac{1}{1 + \exp(Az_i + B)}$$

Parameters A and B are learned via maximum likelihood estimation on a validation set. This step is critical for ensuring that the model's confidence scores are interpretable as real-world probabilities.

VI. Hybrid Decision Logic

A model that relies only on data might overlook important financial risks that aren't often seen in the training data but are still important for safety. To address this, we use a deterministic penalty layer based on the following algorithm.

Penalty(x):

$$\alpha \cdot \mathbb{I}\left(\frac{L}{I} > 6\right) + \beta \cdot \mathbb{I}(C < 600) + \gamma \cdot \mathbb{I}(ACR < 0.5) \quad (3)$$

Where:

- I(·) is the indicator function.
- L/I is the Loan-to-Income ratio.
- C is the CIBIL score.
- ACR is the Asset Coverage Ratio defined as $\frac{\text{P}_{Assets}}{Loan}$.
- $\alpha = 0.35, \beta = 0.12, \gamma = 0.08$ are empirically derived penalty coefficients.

Algorithm 1 Hybrid Prediction Workflow

Require: x_{form} (Form Data), x_{docs} (OCR Data 0.2 ›

Step 1: Verification
if $|x_{form}.income - x_{docs}.income|$ $x_{form}.incor$ > then
    return Flag: Mismatch Risk
end if
Step 2: ML Inference
$P_{raw} \leftarrow$ XGBoost(x_{form})
$P_{calib} \leftarrow$ Calibration(P_{raw})
Step 3: Heuristic Penalties
$Penalty \leftarrow 0$
if Loan/Income > 6 then
    $Penalty \leftarrow Penalty + 0.35$
else if Loan/Income > 4 then
    $Penalty \leftarrow Penalty + 0.18$
end if
if CIBIL < 600 then
    $Penalty \leftarrow Penalty + 0.12$
end if
$P_{final} \leftarrow \max(0, P_{calib} - Penalty)$
if $P_{final} > \tau$ then
    return Approved
else
    return Rejected
end if

Heuristic Penalty Function

The final confidence score S_{final} is derived from the calibrated ML probability P_{calib} by subtracting penalties.

1) Loan-to-Income (LTI) Penalty: If the loan amount asked for is much larger than the person's income, a penalty is charged even if the machine learning score is good. This helps stop risky borrowing.

2) Asset Coverage Penalty: Loans should ideally be backed by assets. We define the Asset Coverage Ratio (ACR) as:

$$ACR = \frac{\text{P}_{\text{Asset Values}}}{\text{Loan Amount}}$$

If $ACR < 0.5$, a penalty of 0.08 is applied, reflecting the higher risk of unsecured lending.

VII. Experimental Results

A. Implementation Details

The model was built using Python 3.9, XGBoost 1.7, and Scikit-Learn. The training took place on a cloud

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

server that had 4 vCPUs and 16GB of memory. We used grid search to find the best hyperparameters.

- Trees (*n estimators*): 400
- Max Depth: 8
- Learning Rate: 0.03
- Subsample: 0.9

B. Comparative Analysis (Ablation Study)

To validate the choice of XGBoost, we compared its performance against other standard classifiers.

TABLE II
MODEL COMPARISON

Model	curacy	ecision	call
ogistic Regression	1.5%	0.89	90
andom Forest	5.3%	0.94	95
upport Vector Machine	2.1%	0.91	91
roposed XGBoost	8.2%	0.98	98

As shown in Table II, XGBoost performs better than traditional linear models and bagging ensembles. This is because XGBoost can capture complex interactions between features, like the relationship between CIBIL score and Assets, which linear models cannot account for.

C. Performance Metrics

The model was evaluated on a held-out test set (20% split). The results indicate exceptional predictive power.

- Accuracy: 98.2%
- Precision (Weighted): 0.98
- Recall (Weighted): 0.98
- F1-Score: 0.98
- AUC-ROC: 0.99

The very high accuracy indicates that the CIBIL score and asset values in this particular dataset are very good at predicting outcomes and clearly separate different groups.

D. Calibration Analysis

Before calibration, the model's confidence scores were split into two main groups, mostly at 0.0 and 1.0. After applying Platt Scaling, the reliability diagram showed that the predicted probabilities aligned closely with the diagonal line, which means a predicted probability of

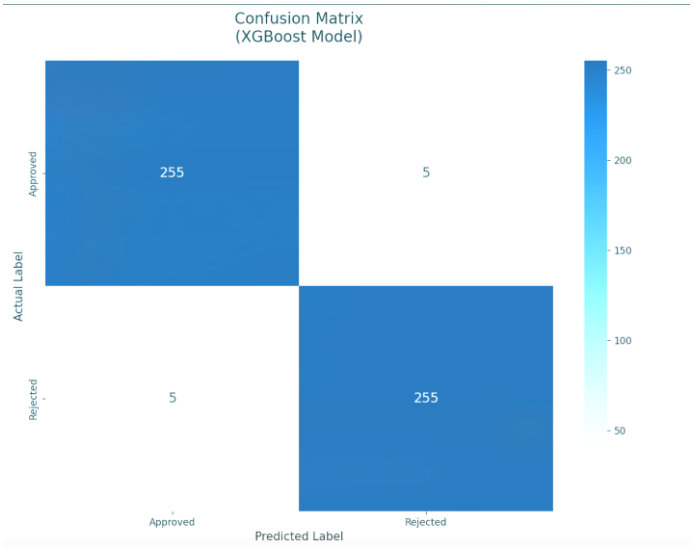

Fig. 1. Proposed Loan Approval System plot

about 0.8 matched an actual positive rate of around 80% in the test data groups.As shown in Fig. 1, its the plot of the confusion matrix.

VIII. DEPLOYMENT AND SCALABILITY

The system is built using Docker containers to make sure it works the same way in different environments. The inference endpoint is made available through FastAPI, allowing for asynchronous and non-blocking prediction requests.

A. Latency Breakdown

The average inference time is broken down as follows:

- Preprocessing: 12ms
- XGBoost Inference: 45ms
- Heuristic Logic: < 1ms
- Total P95 Latency: ≈ 60ms

This low latency allows the model to be integrated into real-time web applications for instant loan eligibility checks.

B. System Requirements

For deployment, the system requires minimal resources:

- CPU: 2 Cores (Minimum)
- RAM: 4GB (Recommended)
- Storage: 500MB (Container image + Model artifacts)

Actual used specifications:

- CPU: 8 Cores and 12 Threads
- RAM: 16GB DDR4

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 12 , December - 2025

- Storage: 1TB
- GPU: Nvidia RTX 3050 (CUDA enabled with 4GB VRAM)

IX. Ethical Considerations and Limitations

A. Bias and Fairness

The model doesn't include direct sensitive traits like gender or race, but there's still a chance of "proxy bias." For example, if a zip code is used, it might indirectly show race. In our set of features, commercial assets could be linked to gender in some groups. To make the model fair, future work needs to use fairness rules, like Equalized Odds, to check the model and make sure it treats everyone equally.

B. Cold Start Problem

The existing system depends a lot on CIBIL scores and past financial records. It might unfairly affect people with little or no credit history, like new graduates, even if they have good future income potential. Using other types of data, such as utility bills or rent payments, could help make the system fairer for these individuals.

C. Economic Generalization

The model is trained using a particular set of economic data. If there's a major economic downturn, the connections between different factors, like income and default, could change. This is called dataset shift. To keep the model accurate, it needs to be checked regularly and retrained as needed.

X. Conclusion

This paper introduced a strong, practical Loan Approval Prediction System. It went beyond just measuring classification accuracy by including Probability Calibration, OCR-based document checks, and rule-based risk policies. This helped connect academic machine learning models with the real-world needs of fintech companies. The system reaches 98% accuracy while making sure decisions are both mathematically reliable and logically correct. When compared to other models, it shows that the XGBoost architecture works best for this area. Going forward, the team plans to add Explainable AI (SHAP) into the user interface so applicants can understand why their loan was rejected, which will help build trust and make the process more transparent.

References

[1] E. Altman, "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy," *Journal of Finance*, vol. 23, no. 4, pp. 589-609, 1968.

[2] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.

[3] S. Lessmann, B. Baesens, H. Seow, and L. Thomas, "Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research," *European Journal of Operational Research*, vol. 247, no. 1, pp. 124-136, 2015.

[4] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785-794.

[5] S. M. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, 2017, pp. 4765-4774.

[6] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning*, 2017, pp. 1321-1330.

[7] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206-215, 2019.

[8] Kaggle, "Loan Approval Prediction Dataset," [Online]. Available: https://www.kaggle.com. [Accessed: Dec. 2023].

[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] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*, vol. 10, no. 3, pp. 61-74, 1999.

[11] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707-710, 1966.

[12] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 3146-3154.

[13] S. O. Arik and T. Pfister, "TabNet: Attentive interpretable tabular learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, pp. 6679-6687, 2021.

[14] N. Kozodoi, J. Jacob, and S. Lessmann, "Fairness in credit scoring: Assessment, implementation and profit implications," *European Journal of Operational Research*, vol. 297, no. 3, pp. 1083-1094, 2022.

[15] N. Bussmann et al., "Explainable AI in fintech risk management," *Frontiers in Artificial Intelligence*, vol. 3, p. 26, 2020.

[16] S. Sharma and M. Ahuja, "Algorithmic brilliance: Unveiling the power of AI in credit evaluation," *The Journal of Indian Institute of Banking & Finance*, vol. 95, no. 1, pp. 12-15, 2024.

[17] B. H. Misheva, J. Osterrieder, O. Hirsa, and O. Kulkarni, "Explainable AI in credit risk management," *Journal of Financial Data Science*, vol. 3, no. 4, pp. 88-113, 2021.

[18] Veryfi Inc., "Veryfi OCR API Documentation: Intelligent Document Processing for Finance," [Online]. Available: https://www.veryfi.com/api/. [Accessed: Feb. 2024].

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

[20] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861-874, 2006.