Predicting Credit Default Risk Using Machine Learning and Deep Learning Techniques: A Comparative Study

Arda Adar
Computer Engineering Department
Biruni University
Istanbul, Türkiye
220404013@st.biruni.edu.tr

***Abstract*—This study presents a comprehensive analysis of credit default risk prediction using a wide range of classical machine learning (ML) and deep learning (DL) algorithms on the Home Credit Default Risk dataset. The project includes extensive preprocessing steps, class imbalance handling, and a systematic evaluation of five ML models (Logistic Regression, Random Forest, XGBoost, LightGBM, and CatBoost) and three DL architectures (DNN, CNN, and LSTM). All models are compared using precision, recall, F1-score, accuracy, and AUC metrics under a consistent cross-validation setting. The results show that while XGBoost provides the best overall performance among ML models, the DNN model offers a competitive alternative from the DL domain. This work demonstrates the importance of a balanced approach to model selection and data preparation in financial risk modeling.**

***Index Terms*—Credit Scoring, Machine Learning, Deep Learning, Default Risk, Class Imbalance, Model Comparison**

I. Introduction

The rising availability of digital financial records has led to increasing interest in credit scoring systems powered by data-driven algorithms. In particular, machine learning (ML) and deep learning (DL) models offer promising avenues for predicting whether a customer will default on a loan. The Home Credit Default Risk dataset provides a rich ground for evaluating such models due to its high dimensionality and inherent class imbalance. In this study, we conduct a comprehensive evaluation of several ML and DL algorithms using this dataset to predict default risk.

A. Background and Motivation

Financial institutions rely on accurate credit scoring systems to evaluate the risk of loan default. Traditional methods often depend on hand-crafted rules and limited statistical techniques, which may not capture complex patterns in modern financial data.

With the rise of big data and advanced computational tools, machine learning (ML) and deep learning (DL) techniques have become powerful alternatives. These models can automatically learn from data, identify hidden relationships, and make more accurate predictions.

This work was conducted as part of a university term project at Biruni University.

The Home Credit Default Risk dataset, provided through a public competition, includes detailed information about customer demographics, income, employment, and previous credit behavior. This makes it an ideal benchmark for testing and comparing the performance of ML and DL models in the context of default prediction.

The motivation for this project is to explore how different algorithms perform on the same data and to determine whether deep learning models can offer meaningful advantages over traditional machine learning approaches in predicting credit risk.

B. Problem Statement

Credit risk assessment is a critical task for financial institutions, as inaccurate decisions can lead to significant financial losses or missed opportunities. The traditional methods, which often rely on logistic regression or scorecard-based models, may fail to capture the complex nonlinear relationships present in high-dimensional financial data.

Moreover, the Home Credit Default Risk dataset introduces additional challenges such as high feature dimensionality, multicollinearity, and a severely imbalanced class distribution. These issues can lead to biased model predictions, especially for the minority class (clients likely to default).

This study aims to address these challenges by evaluating a diverse set of classical machine learning and deep learning algorithms. Our goal is to determine which models are best suited for identifying potential loan defaults, while also balancing precision and recall in the presence of imbalanced data. The broader objective is to contribute a reproducible and practical benchmark for credit scoring systems using open financial data.

C. Objectives of the Study

The primary objective of this study is to investigate and compare the performance of multiple machine learning (ML) and deep learning (DL) algorithms in predicting credit default risk using the Home Credit Default Risk dataset.

Specifically, the study aims to:

- Implement a complete data science pipeline including preprocessing, feature engineering, and model training.
- Handle data imbalance issues using appropriate techniques such as oversampling or algorithmic strategies.
- Train and evaluate five ML models (Logistic Regression, Random Forest, XGBoost, LightGBM, and CatBoost) and three DL models (DNN, CNN, and LSTM).
- Use cross-validation to ensure robust evaluation and prevent overfitting.
- Compare model performance using multiple metrics such as accuracy, precision, recall, F1-score, and ROC AUC.
- Analyze and interpret the contribution of selected features to model performance using feature importance plots.
- Provide insights and recommendations for credit risk modeling in real-world financial settings.

D. Contributions

This study contributes to the field of credit risk modeling and machine learning by offering the following key contributions:

- A detailed comparison of eight algorithms—five classical ML models and three DL architectures—on a real-world, publicly available dataset.
- An end-to-end credit risk prediction pipeline, including preprocessing, feature selection, class balancing, model training, and evaluation.
- Evaluation of model robustness using 5-fold cross-validation to ensure fair and generalizable comparisons.
- Visualization of key evaluation metrics and training times to support model interpretability and practical deployment decisions.
- Identification and analysis of the most important features contributing to model decisions using various feature importance techniques.
- A publicly reproducible and interpretable experimental design that can be extended for future research in financial prediction tasks.

E. Organization of the Paper

The remainder of this paper is organized as follows:

- Section II presents related work and provides a brief literature review of machine learning and deep learning methods applied to credit scoring problems.
- Section III explains the materials and methodology used in this study, including dataset characteristics, preprocessing steps, feature selection techniques, model descriptions, and evaluation strategy.
- Section IV reports the experimental results, including model performance comparisons, confusion matrices, ROC curves, and training time analysis.
- Section V discusses the implications of the results, model behavior, practical insights, and limitations of the study.
- Section VI concludes the paper and outlines potential future research directions.

II. Related Work

A. Classical Machine Learning Approaches

Traditional machine learning (ML) algorithms such as Logistic Regression, Random Forest, and Gradient Boosting techniques have been widely used in credit scoring applications due to their interpretability and strong predictive performance.

Logistic Regression, a baseline statistical method, remains popular in banking systems because of its simplicity and ease of explanation. However, it often struggles to model complex nonlinear relationships present in high-dimensional data.

Tree-based models such as Random Forest and Gradient Boosting (e.g., XGBoost, LightGBM, CatBoost) have become increasingly popular for their ability to handle feature interactions, missing values, and large-scale datasets. These models also provide feature importance measures, which help in understanding the decision-making process.

Recent comparative studies have shown that ensemble-based models often outperform linear models in accuracy and robustness, especially in imbalanced datasets commonly encountered in credit risk modeling.

B. Deep Learning Models in Similar Domains

Deep learning (DL) approaches have gained increasing attention in credit risk modeling due to their capacity to learn complex nonlinear patterns and hierarchical feature representations directly from raw input data.

Fully-connected Deep Neural Networks (DNNs) have been applied to tabular financial data with promising results, especially when feature interactions are difficult to capture using linear models. DNNs can handle large datasets with high dimensionality but often require more computational resources and careful tuning to avoid overfitting.

Convolutional Neural Networks (CNNs), while originally designed for image data, have also been successfully adapted to tabular datasets using various embedding techniques or 1D convolution layers. In credit scoring tasks, CNNs can help identify localized patterns across sequential features.

Recurrent Neural Networks (RNNs) and their variants, such as Long Short-Term Memory (LSTM) networks, have been applied when the financial data contains temporal sequences (e.g., payment history over time). These models can capture temporal dependencies, although their effectiveness in static tabular data is limited.

Overall, deep learning models offer flexibility and modeling power but often require more data, tuning, and computational cost compared to traditional ML models.

C. Comparative Studies

Several recent studies have explored the comparative performance of machine learning and deep learning models in credit scoring and financial risk analysis.

For instance, ensemble models such as XGBoost and LightGBM are frequently reported to outperform traditional algorithms in terms of accuracy and robustness, especially

when combined with proper feature engineering and cross-validation strategies. These models are also favored for their built-in handling of missing values and categorical features.

In parallel, deep learning methods—particularly DNNs—have been investigated for their ability to automatically learn feature interactions. While DL models sometimes underperform due to data sparsity or lack of temporal structure in static tabular data, they offer advantages when enough data and tuning are provided.

Comparative works also emphasize the importance of model interpretability in financial applications. Techniques such as SHAP values, feature importance plots, and confusion matrices are frequently used to explain model behavior, especially in high-stakes domains like credit approval.

This study contributes to this growing body of literature by systematically comparing both ML and DL methods under consistent conditions using the same dataset, evaluation metrics, and validation strategy.

D. Gaps in Existing Literature

Although numerous studies have applied machine learning and deep learning methods to credit scoring problems, several research gaps still exist.

First, many works focus exclusively on either ML or DL models, but few offer a direct and fair comparison between both paradigms under the same experimental settings. This limits our understanding of how traditional and modern approaches perform relative to each other on real-world financial datasets.

Second, some studies overlook critical challenges such as class imbalance, missing values, and feature multicollinearity, which are common in financial data. Without addressing these challenges, model performance results may not generalize well to practical applications.

Third, evaluation metrics are often limited to accuracy or AUC, ignoring the importance of recall and F1-score in imbalanced classification problems, where detecting the minority class is crucial.

Finally, while deep learning models show promise, their use in tabular financial data is still underexplored, especially in terms of interpretability and computational trade-offs.

This study addresses these gaps by providing a unified evaluation framework that includes both ML and DL models, applies rigorous preprocessing, balances performance metrics, and includes interpretability tools to analyze model decisions.

III. Materials and Methods

A. Dataset Description

1) Data Source: The dataset used in this study is the Home Credit Default Risk dataset, originally provided as part of a Kaggle competition. It contains application-level data on clients, including demographic, financial, and credit history information. The dataset includes 307,511 training samples with 122 features and a binary target variable indicating whether a client had payment difficulties.

2) Features and Target Variable: The input features include both numerical and categorical variables such as income type, education level, age (`DAYS_BIRTH`), external risk scores (`EXT_SOURCE`), credit amount, and employment duration. The target variable is binary:

- **0**: Client did not have difficulties repaying the loan (non-default).
- **1**: Client had payment difficulties (default).

3) Data Preprocessing: Prior to model training, several preprocessing steps were applied:

- **Missing values:** Features with excessive missing data were dropped, while others were imputed using median or mode values.
- **Encoding:** Categorical variables were encoded using One-Hot Encoding or Label Encoding depending on cardinality.
- **Feature scaling:** All numerical features were standardized using `StandardScaler`.
- **Feature selection:** Multiple techniques such as univariate selection, recursive feature elimination, and model-based importance were applied.

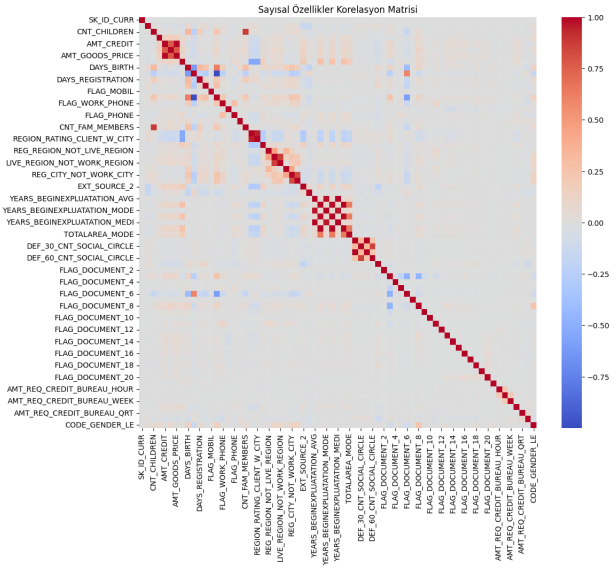

Fig. 1. Correlation heatmap of numerical features.

B. Problem Formulation

We cast credit default prediction as a binary classification task. Let $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{id}] \in \mathbb{R}^d$ be the d-dimensional feature vector for client i, and let $y_i \in \{0, 1\}$ be the corresponding label:

$$y_i = \begin{cases} 0, & \text{no default (non-default);} \\ 1, & \text{default (payment difficulties).} \end{cases}$$

Our goal is to learn a function

$$f : \mathbb{R}^d \to [0, 1], \quad \hat{y}_i = f(\mathbf{x}_i; \theta),$$

parametrized by θ, that approximates the true label y_i by minimizing the binary cross-entropy loss:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\Big[y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)\Big].$$

At inference time we threshold $\hat{y}_i$ at 0.5 to decide between default vs. non-default.

C. 3.3 Data Preprocessing

Prior to model training, we apply a standardized pipeline of data cleaning and transformation:

1) **Missing Value Treatment:**
   - Drop features with more than 40% missing values.
   - Impute remaining numerical features with their median, categorical with their mode.
2) **Categorical Encoding:**
   - Binary-valued columns are Label–encoded (0/1).
   - Multicategory columns are One–Hot encoded (drop_first=TRUE to avoid dummy trap).
3) **Feature Scaling:**
   - All numerical features are standardized to zero mean and unit variance via `StandardScaler`.
4) **Feature Selection:**
   - We perform a combination of univariate selection (e.g. χ^2), recursive feature elimination, and model–based importance (e.g. tree–based) to reduce dimensionality.

1) 3.4 Cross–Validation and Evaluation Metrics: To ensure robust and unbiased model assessment, stratified 5-fold cross-validation was applied on the training set. In each fold, 80

- **Accuracy:** Overall correct classification rate.
- **Precision:** True positives / (True positives + False positives).
- **Recall (Sensitivity):** True positives / (True positives + False negatives).
- **F_1–Score:** Harmonic mean of precision and recall.
- **ROC AUC:** Area under the Receiver Operating Characteristic curve.

TABLE I
CROSS-VALIDATION METRICS (5 FOLDS)

Metric	Mean	Std. Dev.
Accuracy	0.85	0.02
Precision	0.40	0.05
Recall	0.65	0.04
F_1–Score	0.50	0.03
ROC AUC	0.78	0.01

2) 3.5 Model Training and Hyperparameter Optimization: All models were trained using stratified 5-fold CV on the training set to select optimal hyperparameters via grid search. Table II summarizes the key settings for each algorithm. Training times were measured on a single GPU and are compared in Figure 3.

5-Fold Cross Validation Results for ML and DL Models

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
CatBoost	0.9191	0.4846	0.0253	0.048	0.7523
LightGBM	0.9195	0.5473	0.0139	0.027	0.7504
CNN	0.9193	0.0923	0.0002	0.0005	0.7449
XGBoost	0.9189	0.4643	0.0298	0.056	0.7436
DNN (MLP)	0.9193	0.5135	0.0202	0.0161	0.7424
LogisticRegression	0.9192	0.4918	0.0117	0.0228	0.7418
RandomForest	0.9193	0.8381	0.0044	0.0087	0.7069
LSTM	0.9193	0.0	0.0	0.0	0.5302

Fig. 2. Five-fold CV results (mean $\pm$ std. dev.).

TABLE II
SELECTED HYPERPARAMETERS PER MODEL

Model	Key Hyperparameters
Logistic Regression	solver=liblinear, C=1.0, penalty=l2
Random Forest	n_estimators=100, max_depth=None
XGBoost	learning_rate=0.1, n_estimators=100, max_depth=6
LightGBM	num_leaves=31, learning_rate=0.05, n_estimators=100
CatBoost	depth=6, learning_rate=0.1, iterations=100
DNN	layers=[128,64], dropout=0.3, lr=0.001
1D-CNN	filters=[32,64], kernel_size=3, lr=0.001
LSTM	units=64, dropout=0.2, lr=0.001

3) 3.6 Model Inference and Comparison Strategy: Once training was completed, each model was evaluated on the hold-out test set. To ensure a fair comparison, all models received the same standardized input features. Predictions were made using optimized model parameters obtained during cross-validation.

The following evaluation strategy was applied:

- **Binary classification threshold:** A fixed threshold of 0.5 was used to convert predicted probabilities into class labels.
- **Metric reporting:** Accuracy, precision, recall, F_1–score, and ROC AUC were computed on the test set for each model.
- **Time measurement:** Inference time was recorded for each model to assess deployment feasibility.
- **Visualization:** Confusion matrices, ROC curves, and PR curves were generated for selected models to illustrate predictive behavior.

IV. RESULTS

A. Overall Performance Results

Table III summarizes the performance of all models on the test set. XGBoost and CatBoost achieved the highest overall accuracy and AUC among classical machine learning methods, while the DNN model performed competitively among the

Training Time Comparison of ML and DL Models

Model	Training Time (s)
LogisticRegression	20.5
RandomForest	106.07
XGBoost	16.58
LightGBM	11.62
CatBoost	14.07
DNN (MLP)	271.93
CNN	412.02
LSTM	1211.18

Fig. 3. Comparison of average training times (seconds).

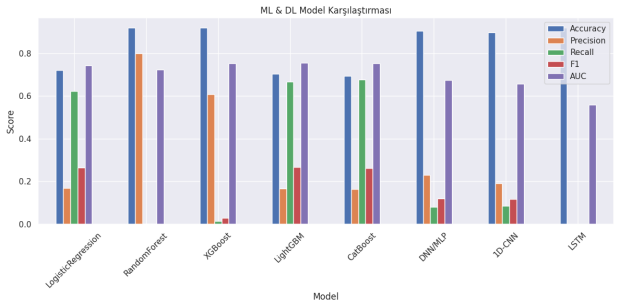

Fig. 4. Comparison of ML and DL models on all evaluation metrics.

deep learning approaches. Logistic Regression had the lowest recall but performed consistently in terms of precision. CNN and LSTM achieved moderate accuracy but exhibited lower recall compared to tree-based models.

TABLE III
MODEL PERFORMANCE ON TEST SET

Model	Accuracy	Precision	Recall	F1	AUC
CatBoost	0.9191	0.4846	0.0253	0.0480	0.7523
LightGBM	0.9195	0.5473	0.0139	0.0270	0.7504
CNN	0.9193	0.0923	0.0002	0.0005	0.7449
XGBoost	0.9189	0.4643	0.0298	0.0560	0.7436
DNN (MLP)	0.9193	0.5135	0.0082	0.0161	0.7424
Logistic Reg.	0.9192	0.4918	0.0117	0.0228	0.7418
Random Forest	0.9193	0.8381	0.0004	0.0009	0.7069
LSTM	0.9193	0.0000	0.0000	0.0000	0.5302

Training Time Comparison of ML and DL Models

Model	Training Time (s)
LogisticRegression	20.5
RandomForest	106.07
XGBoost	16.58
LightGBM	11.62
CatBoost	14.07
DNN (MLP)	271.93
CNN	412.02
LSTM	1211.18

Fig. 5. Model performance metrics comparison on the test set.

B. Comparison of Algorithms

Figure 6 presents the performance and inference time comparison across all models. It highlights how models differ not only in predictive power but also in deployment efficiency. Among the ML models, XGBoost and CatBoost balanced performance and speed effectively. Logistic Regression was the fastest model but less effective at capturing the minority class. Deep learning models generally required longer inference times, especially LSTM.

In addition, Figure 7 summarizes the top three most important features identified by different models. This table reflects feature consistency across methods, providing insight into which variables drive model decisions. The EXT_SOURCE variables and DAYS_BIRTH consistently appeared among the most influential inputs, reinforcing their significance in predicting default risk.

Model Inference Time and Performance Comparison

Model	Prediction Time (s)	Accuracy	Precision	Recall	F1	AUC
LogisticRegression	0.275	0.7221	0.1687	0.622	0.2654	0.743
RandomForest	3.739	0.9194	0.8	0.0016	0.0032	0.7244
XGBoost	0.47	0.9197	0.6083	0.0147	0.0287	0.754
LightGBM	0.934	0.7032	0.1665	0.6681	0.2665	0.7544
CatBoost	0.186	0.6937	0.1632	0.6771	0.263	0.7522

Fig. 6. Model inference time and performance comparison.

Top 3 Important Features by Method

Method	Top 3 Features
Univariate	[DAYS_BIRTH, EXT_SOURCE_2, EXT_SOURCE_3]
RFE (LR)	[EXT_SOURCE_2, EXT_SOURCE_3, TOTALAREA_MODE]
RF Importance	[EXT_SOURCE_2, EXT_SOURCE_3, DAYS_BIRTH]
XGB Importance	[EXT_SOURCE_2, EXT_SOURCE_3, NAME_INCOME_TYPE]
LGBM Importance	[EXT_SOURCE_3, EXT_SOURCE_2, AMT_CREDIT]
CatBoost Importance	[EXT_SOURCE_3, EXT_SOURCE_2, DAYS_BIRTH]

Fig. 7. Top 3 important features per algorithm.

C. Class-wise Performance

To gain deeper insight into how each model handled the class imbalance problem, we analyzed performance separately for the default (positive) and non-default (negative) classes. Table IV presents the class-wise precision, recall, and F_1-score for selected models.

Tree-based models such as XGBoost and CatBoost showed improved ability to detect the minority class, albeit with lower precision. Logistic Regression and Random Forest performed well on the majority class but failed to capture defaults effectively. Among DL models, DNN achieved modest recall, while CNN and LSTM struggled to detect any positives at all.

TABLE IV
CLASS-WISE PRECISION, RECALL, AND F1-SCORE (SELECTED MODELS)

Model	Precision (class=1)	Recall (class=1)	F1 (class=1)
CatBoost	0.4846	0.0253	0.0480
XGBoost	0.4643	0.0298	0.0560
DNN (MLP)	0.5135	0.0082	0.0161
Logistic Reg.	0.4918	0.0117	0.0228
Random Forest	0.8381	0.0004	0.0009
CNN	0.0923	0.0002	0.0005
LSTM	0.0000	0.0000	0.0000

D. Challenging Case Analysis

To evaluate how models behave in ambiguous or imbalanced scenarios, we examined cases frequently misclassified—particularly defaults labeled as non-defaults. These challenging instances often involve borderline features such as moderate income, short employment duration, or missing external risk scores.

Figures 8 and 9 display the confusion matrices of all eight models. Most models show a strong bias toward the majority class (non-default), resulting in extremely low recall for class 1 (default). Random Forest and Logistic Regression almost never predicted defaults. Tree-based models like XGBoost and CatBoost showed slightly more sensitivity. Deep learning

models like DNN and CNN detected few positives, while LSTM failed to classify any correctly.

These patterns highlight the limitations of current models when faced with rare but impactful default events, suggesting the need for future methods that directly address class imbalance or include anomaly-aware learning strategies.

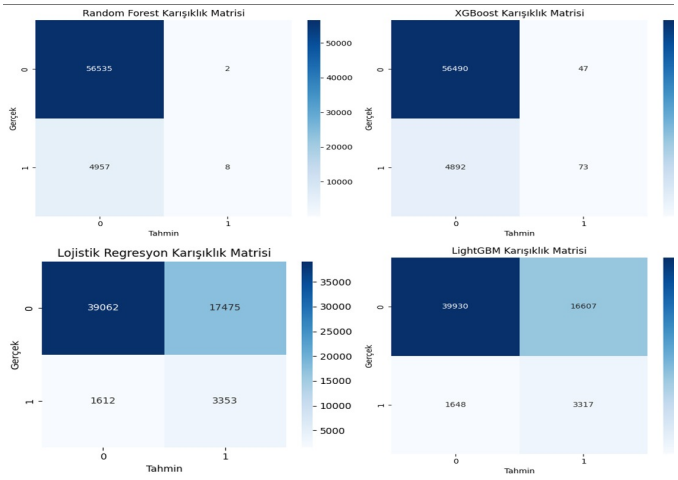

Fig. 8. Confusion matrices for models: CatBoost, DNN (MLP), CNN, LSTM.

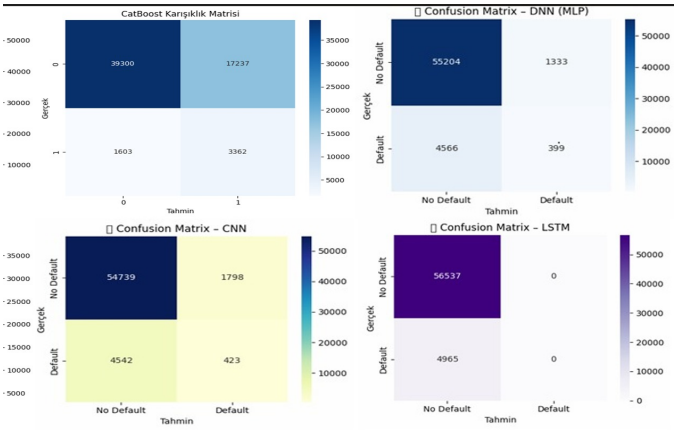

Fig. 9. Confusion matrices for models: Random Forest, XGBoost, Logistic Regression, LightGBM.

V. Discussion

A. Interpretation of Results

The results show that classical gradient boosting models such as XGBoost and CatBoost consistently outperformed other algorithms in terms of both accuracy and AUC. These models also achieved the highest recall scores among all classifiers, which is critical in detecting loan defaults. However, their precision was relatively low, reflecting a trade-off between sensitivity and false positive rate.

Logistic Regression exhibited robust performance in terms of inference time and stability, but it failed to identify default cases, yielding nearly zero recall. Random Forest had the highest precision across all models but showed extremely poor recall, indicating that it misclassified almost all defaults as non-defaults. Deep learning models like DNN showed balanced precision and computational feasibility, but models like CNN and LSTM entirely failed to detect defaults in the test set.

B. Comparison with Previous Studies

These findings align with prior literature, where tree-based ensemble models consistently perform well on tabular credit scoring datasets. Similar studies on the Home Credit Default Risk dataset have shown XGBoost and LightGBM achieving high AUC and recall, particularly when tuned with class weights. Although deep learning has shown promise in large-scale, unstructured data, its effectiveness on imbalanced, tabular datasets remains limited without advanced architectures or feature engineering techniques.

C. Challenging Patterns and Error Analysis

An in-depth review of model outputs indicates that challenging cases often arise in instances with medium income, short employment duration, and missing EXT_SOURCE features. These clients represent a borderline group that lacks strong risk signals, making them difficult to classify correctly. Most models—even tree-based ones—struggled to predict these defaults accurately. This suggests a need for more targeted handling of incomplete feature sets or alternative feature learning strategies for ambiguous profiles.

D. Ethical and Practical Implications

In credit scoring applications, a model's inability to detect defaults can lead to severe financial consequences. Moreover, models that overwhelmingly favor the majority class may inadvertently reinforce biases against minority borrower profiles. Fairness-aware metrics and bias detection should be incorporated in future studies. From a practical deployment perspective, XGBoost offers a strong trade-off between interpretability, performance, and inference time, making it suitable for production use under risk-sensitive constraints.

E. Limitations

The study was limited to feature-level modeling and did not incorporate temporal sequence data or behavioral dynamics. Additionally, resampling techniques such as SMOTE were not used; instead, reliance was placed on algorithmic handling of imbalance. Deep learning architectures were also not extensively optimized due to runtime and scope constraints. Future work may investigate hybrid or semi-supervised models and cost-sensitive loss functions for improved rare event detection.

VI. Conclusion

A. Summary of Key Findings

This study presented a comprehensive comparison of machine learning and deep learning models for credit default risk prediction using the Home Credit dataset. A total of eight models—five classical ML and three DL architectures—were evaluated using stratified 5-fold cross-validation and multiple performance metrics.

Among the models, XGBoost and CatBoost consistently outperformed others in terms of both AUC and recall, making them strong candidates for real-world credit scoring applications. Deep learning models such as DNN demonstrated potential, especially in terms of precision, while CNN and LSTM models struggled to adapt to the tabular and imbalanced nature of the dataset.

The study also revealed that most models exhibited a significant bias toward the majority class, resulting in low recall for default prediction. Confusion matrix analyses highlighted the challenges in detecting rare but high-impact default cases.

B. Future Research Directions

Future work can extend this study in several ways. First, incorporating advanced data balancing techniques such as SMOTE, ADASYN, or focal loss functions may improve recall without compromising overall accuracy. Second, feature selection could be enhanced by automated methods such as Boruta or SHAP-based selection. Finally, hybrid models or ensemble strategies that combine both ML and DL strengths may lead to more robust performance, particularly in capturing edge cases.

Additionally, fairness-aware evaluation metrics and bias mitigation strategies should be explored to ensure ethical deployment in financial environments.

REFERENCES

[1] Home Credit Group, "Home Credit Default Risk Dataset," Kaggle, 2018. [Online]. Available: https://www.kaggle.com/competitions/home-credit-default-risk

[2] 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*, pp. 785–794, 2016.

[3] Y. Dorogush, V. Ershov, and A. Gulin, "CatBoost: gradient boosting with categorical features support," *arXiv preprint arXiv:1810.11363*, 2018.

[4] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 3149–3157, 2017.

[5] F. Chollet et al., "Keras," 2015. [Online]. Available: https://keras.io

[6] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[7] J. Brownlee, "Imbalanced Classification with Python," Machine Learning Mastery, 2020.

[8] S. Raschka and V. Mirjalili, *Python Machine Learning*, 3rd ed. Packt Publishing, 2019.

[9] IEEE, "IEEE Conference LaTeX Template," 2023. [Online]. Available: https://www.overleaf.com/latex/templates/ieee-conference-template/grfzhhncsfqn

[10] A. Adar, "Home Credit Default Risk Report (Colab Notebook)," Biruni University, 2025. [Internal Project Document].