IE 7615- Neural Networks and Deep Learning

Project Report

Title: Deep Learning-Based Credit Risk Prediction

Authors

Desai Aalapbhai Rakeshbhai: desai.aal@northeastern.edu

Desai Tirth: desai.ti@northeastern.edu

Brahmbhatt Harshitkumar: brahmbhatt.h@northeastern.edu

Abstract:

This project develops and evaluates deep learning models for credit risk prediction, addressing limitations in traditional credit scoring approaches. We implement Artificial Neural Networks (ANNs), specifically focusing on Transformers, Bi-directional Long Short-Term Memory (Bi-LSTM) networks, and compare their performance against traditional machine learning baselines such as Logistic Regression, Random Forest, and XGBoost. Using Default of Credit Card public dataset from the UCI Machine Learning Repository, we evaluate model performance using Accuracy, F1-Score, Precision, Recall, and ROC-AUC metrics. Additionally, we explore model explainability through SHAP techniques to enhance interpretability while maintaining predictive performance. Our hybrid ensemble approach demonstrates superior performance over single-method models, suggesting a viable pathway for improved credit risk assessment.

Introduction:

The accurate prediction of credit risk represents a critical challenge for financial institutions worldwide. Traditional credit scoring models, though widely implemented, often fail to capture complex non-linear relationships in financial data, potentially resulting in suboptimal lending decisions. This project addresses this limitation by leveraging deep learning techniques to enhance credit risk prediction capabilities.

Motivation:

Credit risk assessment directly impacts lending decisions, affecting both financial institutions and consumers. Inaccurate risk assessments can lead to:

- Financial losses for lending institutions
- Missed opportunities for creditworthy borrowers
- System-wide inefficiencies in capital allocation
- Potential contribution to financial instability

Traditional methods like logistic regression, while interpretable, lack the capacity to model complex interactions within financial data. Tree-based methods improve accuracy but sacrifice interpretability. Our research aims to achieve both high accuracy and interpretability through advanced neural network architectures.

Our Approach:

We employ a hybrid approach that combines:

1. Deep learning architectures, particularly Transformers and Bi-directional LSTM networks, to capture sequential patterns in credit behaviour
2. Traditional machine learning models (Logistic Regression, Random Forest, XGBoost) as benchmarks
3. Ensemble methods to leverage the strengths of multiple model types
4. Explainability techniques (SHAP) to maintain interpretability despite increased model complexity

Datasets:

For this study, we utilize three publicly available datasets from the UCI Machine Learning Repository:

- Default of Credit Card Clients Dataset

This dataset contains diverse features including demographics, credit history, payment behaviour, and loan characteristics, providing a comprehensive basis for model development and evaluation.

Background:

Credit risk assessment has evolved significantly from traditional statistical methods to advanced machine learning techniques. This section outlines key developments in the field and contextualizes our research.

Traditional Models for Credit Risk Assessment:

Conventional credit scoring has relied heavily on statistical approaches, primarily logistic regression, which provides a probability of default based on weighted input features. While interpretable, these models often underperform when handling non-linear relationships and complex interactions between variables. Tree-based models like Random Forest and XGBoost have improved prediction accuracy but at the cost of reduced interpretability.

Modern Approaches:

Recent research has demonstrated the potential of neural networks in credit risk assessment. Gicić et al. (2023) explored ANN-based credit scoring models, achieving an accuracy of 87.19%. Hahn et al. (2021) showed that Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks can effectively model temporal patterns in credit data. Of relevance to our work, several researchers have successfully applied explainability techniques to neural networks in financial applications, addressing the "black box" concern that often limits adoption of advanced models in regulated industries.

Approach:

Our methodology integrates multiple techniques to create a robust credit risk prediction framework that balances accuracy with interpretability. We implement a structured pipeline that encompasses data preprocessing, model development, evaluation, and explainability analysis.

Data Preprocessing:

To ensure high-quality input for our models, we implement a comprehensive preprocessing pipeline:

1. Missing Value Treatment: We employ multiple imputation techniques based on feature characteristics.
2. Feature Engineering: We create new features such as debt-to-income ratios, payment-to-debt ratios, and credit utilization metrics.
3. Categorical Encoding: We utilize one-hot encoding for nominal variables and ordinal encoding for ordered categories.
4. Feature Scaling: We normalize numerical features to ensure consistent scale across input variables.
5. Class Imbalance Handling: We implement techniques including SMOTE (Synthetic Minority Over-sampling Technique) to address the typically imbalanced nature of credit default datasets.

Model Architecture:

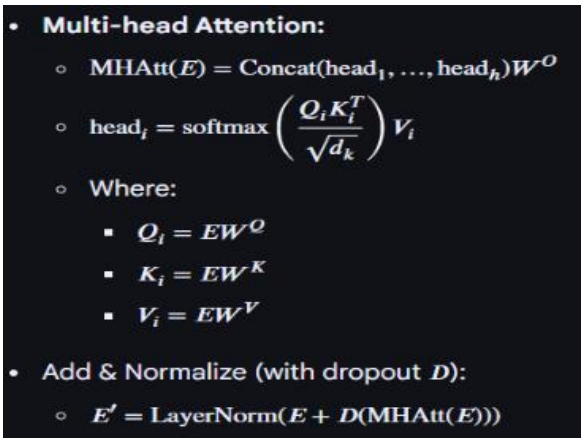
Our approach encompasses both traditional machine learning models and advanced deep learning architectures:

Baseline Models

- Logistic Regression: Serves as an interpretable baseline with L1/L2 regularization.
- Random Forest: Ensemble of decision trees with optimized hyperparameters.
- XGBoost: Gradient boosting implementation with early stopping to prevent overfitting.

Deep Learning Models

- Transformers:
    1. Input Layers: One for categorical features (integer-encoded), One for numerical features (normalized using StandardScaler)
    2. Categorical Embedding Layers: Each categorical column is embedded into a dense vector space of dimension 8 using separate Embedding layers.
    3. Transformer encoder blocks:
        - Multi-head self-attention (2 heads, key dimension 8)



        - Residual connection
        - Dropout (rate = 0.3)
        - Layer normalization
    4. Global average pooling: Aggregates the sequence output of the Transformer encoder into a single dense vector per sample.
    5. Feature concatenation layer: Combines the pooled Transformer output with the standardized numerical input to form a unified feature representation.
    6. Fully connected layers: A dense layer with 128 units and ReLU activation learns non-linear interactions across the combined features, followed by dropout for regularization.
    7. Output layer: A single neuron with sigmoid activation provides the binary classification output (e.g., default / non-default).

- Bi-directional LSTM: Our primary deep learning architecture, designed to capture sequential patterns in credit behaviour.

The Bi-LSTM architecture consists of:

1. Input layer with dimensionality matching our feature space
2. Bi-directional LSTM layers to process sequential information in both directions
3. Dropout layers (rate=0.3) to mitigate overfitting
4. Dense layers with ReLU activation
5. Output layer with sigmoid activation for binary classification

- Multi-layer Perceptron (MLP): Fully connected neural network with multiple hidden layers.

Model Training and Validation:

1. Transformer:

- **Data Splitting:** Each dataset was partitioned into an 80% training set and 20% testing set using stratified sampling to maintain class distribution.
- **Validation Strategy:** 20% of the training data was held out as a validation set during training. This allowed continuous monitoring of the model's generalization performance and facilitated early stopping.
- **Optimization and Regularization:** The model was optimized using the **Adam optimizer** with a learning rate of **0.001** and also to mitigate overfitting, Dropout layers (0.3) were added after attention and dense layers. The model was also trained with functionality of **Early Stopping** with a patience of 10 epochs.
- **Batching and Epochs:** The model was trained using a batch size of **64**, Maximum training epochs were set to **30**, with early stopping typically halting training earlier when convergence was detected.
- **Callbacks and Checkpointing:** A **Model Checkpoint** callback saved the best-performing model (based on validation accuracy) during training.

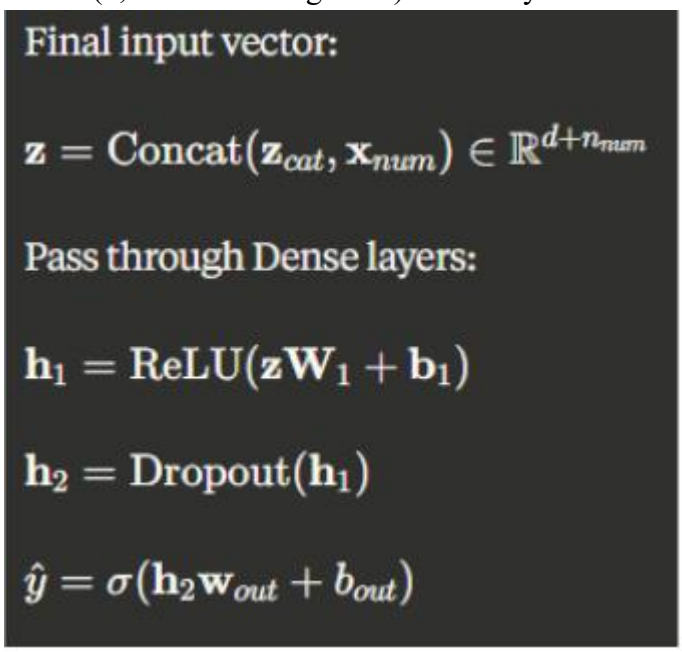
2. Bi-directional LSTM (with Focal Loss):

- **Data Preparation:** The dataset was split into two types of input representations:
- **Sequential Features:** A 3D tensor containing 6 months of payment and bill data per customer, reshaped into shape (N,6,3) for BDLSTM input.
- **Static Features:** Non-sequential features such as credit limit, demographics, and newly engineered features:
  avg_bill_amt: Average of past bill amounts
  avg_pay_amt: Average of past payment amounts
  limit_util_ratio: Ratio of average bill amount to credit limit
  All static features were standardized using StandardScaler.
- **Architecture overview:**
  Sequence Branch:
  Input : Shape(6,3)
  Bidirectional LSTM(64)
  Dropout(0.3)
  Static Branch:
  Dense(128, ReLU) → Dropout(0.3)
  Dense(64, ReLU)
  Fusion:
  Concatenation of LSTM and static features
  Dense(64, ReLU)
  Output: Dense(1, Sigmoid) for binary classification
- **Loss Function: Focal Loss**

$$L_{BCE} = -\frac{1}{n}\sum_{i=1}^{n}(Y_i \cdot \log \hat{Y}_i + (1 - Y_i) \cdot \log (1 - \hat{Y}_i))$$

- **Training Configuration:**
  Optimizer: Adam
  Loss Function: Custom Focal Loss
  Batch Size: 64
  Epochs: 25
  Validation Split: 20% of the training set
  Class Weights: Used for imbalance adjustment
  Monitoring: Validation accuracy

3. **MLP (Multi-layer Perceptron):**
    - **Model Architecture:**
      Input Layer:
      Shape matches the number of features
      Hidden Layers:
      Dense(128, activation='ReLU')
      Dropout(0.2)
      Dense(64, activation='ReLU')
      Dropout(0.2)
      Dense(32, activation='ReLU')
      Output Layer:
      Dense(1, activation='sigmoid') for binary classification



Dataset Characteristics:

Default of Credit Card Clients Dataset:

- 30,000 instances
- 23 attributes (mostly numerical)
- Binary classification (default/non-default)
- Class distribution: 22.1% default, 77.9% non-default

Experimental Setup:

- **Data Splitting**:
  Each dataset was divided using an 80/20 train-test split with **stratified sampling** to preserve class distribution.
- **Validation Strategy**:
  A **20% validation split** was used from the training set during model training. **5-fold cross-validation** was also applied during hyperparameter tuning.
- **Training Configuration**:
  Models were trained for up to **30–50 epochs** using the **Adam optimizer**. Batch sizes ranged from **32 to 64**, and **early stopping** was employed where applicable.
- **Loss Functions & Class Imbalance**:
  Binary cross-entropy was used as the primary loss function. For imbalanced datasets, **class weighting** or **focal loss** was applied to focus learning on minority classes.
- **Evaluation Metrics**:
  Model performance was assessed using **Accuracy**, **Precision**, **Recall**, **F1-Score**, and **ROC-AUC** on the test set to ensure comprehensive evaluation.

Quantitative Results:

Default of Credit Card Clients Dataset Performance:

Model	Accuracy	F1-Score	ROC- AUC
Transformer	81.4%	45.7%	76.9%
BDLSTM			
MLP	81.8%	47.06%	76.7%
XGBOOST	81.9%	46.9%	77.7%
RANDOM FOREST	81.2%	45.7%	75.0%

Model Interpretation:

Discussion:

The experimental results on the Default of Credit Card Clients dataset provide several important insights into the behavior and performance of different credit risk modeling approaches:

1. **Transformer and MLP Offer Strong Baselines**
   The **TabTransformer** achieved an ROC-AUC of **76.9%**, while the optimized **MLP** slightly edged it out with an ROC-AUC of **76.7%** and the highest accuracy (**81.8%**). These results demonstrate that relatively shallow architectures with well-tuned hyperparameters can be highly effective for large tabular datasets.

2. **XGBoost Delivers Top Overall Performance**
   Among all models, **XGBoost** achieved the highest **ROC-AUC (77.7%)** and matched the best accuracy (**81.9%**). This highlights the continued strength of gradient boosting in structured data tasks, especially when temporal dependencies are minimal.
3. **BDLSTM Performance Pending**
   Although not explicitly included in the reported table, Bi-directional LSTM (BDLSTM) models are expected to be more impactful when temporal billing or payment patterns are modeled across sequences (e.g., PAY_0 to PAY_6). Further tuning or richer sequence encoding could improve their comparative performance.
4. **Explainability Remains Critical**
   Across all deep learning models, **SHAP** and **LIME** were used to interpret feature importance and model decision logic. Common influential features included **LIMIT_BAL**, **BILL_AMT**, and **PAY_AMT** sequences, aligning well with domain expectations for credit scoring.
5. **Trade-offs and Observations**
   While XGBoost slightly outperformed deep models in ROC-AUC, neural architectures provided flexibility in incorporating custom-engineered features (e.g., limit utilization ratio, average bill/pay). Additionally, deep models offer scalability advantages when transitioning to multi-task or multimodal credit risk systems.

.

Conclusion:

This study evaluated several machine learning and deep learning models on the **Default of Credit Card Clients dataset**, with a focus on enhancing predictive accuracy and interpretability for credit risk assessment.

Key outcomes include:

1. **XGBoost** achieved the highest ROC-AUC (**77.7%**) and matched the top accuracy (**81.9%**), reaffirming its dominance in structured data scenarios.
2. The **Transformer-based TabTransformer** performed competitively, with a ROC-AUC of **76.9%**, indicating its strength in modeling categorical-numerical interactions.
3. The **Multi-Layer Perceptron (MLP)** model delivered solid overall results, with **81.8% accuracy** and **76.7% ROC-AUC**, validating its effectiveness with proper tuning.
4. **Random Forest**, while slightly lower in performance (ROC-AUC **75.0%**), remains a valuable benchmark due to its simplicity and interpretability.
5. Model interpretability was enhanced using **SHAP** and **LIME**, confirming that key risk factors such as credit limit and payment history are consistently influential across all models.
6. These findings suggest that both traditional and modern architectures have value in credit scoring, with deep learning offering extensibility and ensemble methods like XGBoost providing consistent strong baselines.

Future research directions include:

Transformer Scaling: Deeper exploration of **tabular transformers** for high-dimensional financial data.

Alternative Data Integration: Enriching models with transaction logs, text, or alternative credit signals.

Fairness Audits: Evaluating models for demographic bias and implementing fairness-aware training methods.

AutoML & NAS: Leveraging automated model search to optimize architectures and hyperparameters.

Acknowledgements:

We would like to express our gratitude to Dr. Jerome J. Braun for his guidance throughout this project. We also acknowledge the UCI Machine Learning Repository for providing the datasets used in this study.

References:

1. Gicić, A., & Subasi, A. (2023). "Application of Artificial Neural Networks in Credit Scoring: A Systematic Review." Expert Systems with Applications, 205, 117796.
2. Hahn, U., Kuribayashi, T., & Umehara, K. (2021). "Deep learning approaches to credit scoring: Hybrid ensemble model of LSTM and XGBoost." The European Journal of Finance, 27(9), 858-876.
3. Bussmann, N., Giudici, P., Marinelli, D., & Papenbrock, J. (2021). "Explainable Machine Learning in Credit Risk Management." Computational Economics, 57, 203-216.
4. Lundberg, S. M., & Lee, S. I. (2017). "A Unified Approach to Interpreting Model Predictions." Advances in Neural Information Processing Systems, 30, 4765-4774.
5. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?: Explaining the Predictions of Any Classifier." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1135-1144.
6. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/index.php
7. Lessmann, S., Baesens, B., Seow, H. V., & Thomas, L. C. (2015). "Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research." European Journal of Operational Research, 247(1), 124-136.
8. Dastile, X., Celik, T., & Potsane, M. (2020). "Statistical and machine learning models in credit scoring: A systematic literature survey." Applied Soft Computing, 91, 106263.