

Article

A Transformer-Based Neural Network to Predict Credit Card Default

Zongqi Hu *  and Chai Kiat Yeo 

College of Computing and Data Science, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore; asckyeo@ntu.edu.sg

* Correspondence: zongqi001@e.ntu.edu.sg; Tel.: +65-97550776

Abstract

We propose a transformer-based neural network for predicting credit card default using raw multivariate credit data represented as a 2D time series, eliminating the need for manual feature engineering. Unlike existing state-of-the-art (SOTA) tree-based models that rely heavily on handcrafted features, our model leverages self-attention to extract latent temporal patterns directly from the raw data. Evaluated on two real-world datasets, our approach outperforms the popular LightGBM baselines and achieves performance on par with the leading ensemble methods. To further explore if our proposed model can enhance common ensemble methods, we incorporate it into an ensemble together with LightGBM. Experimental results show that the ensemble integrating our proposed transformer-based model outperforms existing ensemble approaches. Designed with deployment in mind, the model architecture is lightweight, generalizable, and maintainable, making it suitable for integration into real-world credit risk pipelines. Our results demonstrate strong practical relevance and a clear path towards scalable deployment in financial applications. In addition, we have built in an optional feature augmentation extension to the proposed model to facilitate hybrid adoption of our model by existing users who are accustomed to engineered features from domain expertise and industry practice. Hence, our model is user-friendly and can leverage hybrid learning to support both user-crafted and model-learned features to improve model performance and deployment.

Keywords: credit default; time series prediction; machine learning; transformer

1. Introduction

1.1. Credit Default Risk and Prediction Challenges

Credit default risk refers to the likelihood that a borrower (individual or corporate) will be unable to repay their loan obligations [1,2]. It is a critical component of financial risk, which encompasses market, liquidity, operational, and legal risks [3]. The ability to accurately assess this risk is essential for maintaining the financial health of lending institutions and the broader economy.

While credit has long played a foundational role in commerce [4,5], the digital era has intensified the challenge. The rise of online lending platforms and short-term credit products has increased the volume and velocity of credit issuance [6]. When borrowers default, the resulting losses can propagate across the financial system, as seen during the 2008 financial crisis [7]. In response, global regulatory frameworks—such as the Basel Accords—have mandated more stringent credit risk management, prompting institutions to adopt advanced risk assessment and mitigation tools [8].



Academic Editor: George A. Tsihrintzis

Received: 6 April 2026

Revised: 27 April 2026

Accepted: 14 May 2026

Published: 15 June 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\) license](https://creativecommons.org/licenses/by/4.0/).

Modern financial institutions are now provisioned with vast quantities of customer data, enabling the use of data mining and machine learning techniques to identify risk patterns [9,10]. Accurate default prediction models are especially valuable in this context: even a modest increase in predictive accuracy (e.g., 1%) can translate into substantial reductions in financial losses [11,12]. However, existing systems often rely on extensive feature engineering and handcrafted rules, which are costly to maintain and slow to adapt to changing borrower behaviors. This motivates the development of automated, generalizable models capable of learning directly from the raw data, which forms the basis of our proposed approach.

1.2. Our Contributions

This paper extends our earlier proof-of-concept transformer-based credit default prediction model introduced in [13]. While the earlier conference paper established the core hypothesis, this extended work elevates the approach into a mature, rigorously validated, and deployment-ready framework. The key contributions are:

- **Optimized Transformer for 2D credit time-series modeling:** We model multivariate credit behavior data as a 2D time series (time $\times$ data) and apply a transformer-based neural network to capture latent temporal dependencies.
- **Eliminating the need for feature engineering:** Our model outperforms traditional LightGBM baselines without requiring handcrafted features, reducing the development burden and enabling faster deployment.
- **Performance comparable with SOTA models which use extensive feature engineering:** Even without engineered inputs, our approach achieves predictive performance on par with SOTA models that use thousands of features, demonstrating its strong learning capacity and efficiency.
- **Generalization across diverse datasets:** We validate the model on two real-world datasets (AMEX and Taiwan Bank), showing consistent performance across different regions, customer segments and data distributions.
- **Supports flexible integration and enhancement:** The transformer-based model can be easily extended to include an optional feature augmentation extension or combined in ensemble architectures. We show that incorporating the proposed model into an ensemble with LightGBM achieves better performance than prior SOTA ensembles.
- **Designed for deployment:** The model architecture and data pipeline are lightweight, interpretable and easily integrated into existing financial infrastructure, supporting practical adoption in credit risk systems.

While traditional methods such as tree-based ensembles have proven effective in credit risk prediction, they typically rely heavily on static borrower attributes and require extensive feature engineering to capture the temporal trends [14]. This makes them less adaptable and harder to deploy in real-world financial systems, where rapid iteration and model maintenance are critical.

In contrast, our approach leverages a transformer-based neural network that models raw sequential credit data as a 2D time series, eliminating the need for handcrafted features. This design not only improves prediction accuracy, but also reduces development overhead, paving the way for faster, scalable and maintainable deployment in practical credit scoring applications. To promote reproducibility and facilitate further research, the code availability information for this study is provided in the Data Availability Statement.

2. Related Works

Credit default prediction task can be essentially treated as a binary classification problem (default vs non-default). Based on recent review studies [6,15,16], existing methods to predict credit default can generally be grouped into the following:

- Statistical models;
- Traditional machine learning models;
- Neural networks;
- Ensemble approaches.

The field of credit scoring boasts a variety of statistical methods, including discriminant analysis [17], logistic regression [18] and linear probability models [15]. These techniques are popular due to their ease of use and ability to provide insights into borrower behaviors [19]. There are also studies that implement traditional machine learning models such as support vector machine (SVM), Naive Bayes (NB), and k-nearest-neighbor (KNN), etc., to predict credit default [9,20].

Refs. [21,22] are among the early works to compare the different methods for credit scoring on different real-world datasets and suggest that neural networks could potentially be an effective approach to the problem. Ref. [23] proposed transformer-based networks. Refs. [24,25] proposed networks that combine the power of transformer and convolutional networks (CNN). Note that for these studies, the majority of the credit risk datasets (e.g., [26,27]) are in a tabular format where each row represents a unique borrower or a specific loan application and each column corresponds to a particular piece of information about that borrower (e.g., financial information, loan details, etc.).

Ensemble techniques in machine learning aim to improve the prediction performance by combining the outputs of several base models and can be generally classified into three types: bagging (e.g., random forest), boosting (e.g., XGboost, lightGBM), and stacking. Recent studies such as [28,29] suggest that tree-based boosting algorithms generally perform better than others (including deep tabular models such as TabNet or SAINT) on tabular datasets and therefore become more dominant in credit risk prediction. For datasets that are not originally exactly tabular (e.g., [30,31]), many studies (e.g., [32–35]) also tend to transform the raw data into a single wide tabular dataset first and then apply tree-based boosting models.

To further improve prediction performance, recent studies gradually shift focus to explore non-tabular data input. Ref. [36] proposed RNN model to specifically process raw transactional data. Ref. [37] proposed an edge weight-shared graph convolutional network to process transactional data. Ref. [10] introduced attentive feature fusion for credit default prediction. Ref. [38] proposed the use of a BERT-based NLP to process text features. Ref. [39] proposed a transformer-based network to specifically process user behavior events. Ref. [40] proposed dynamic graph learning to process user-merchant and user-user interactions on two real-world private datasets.

Furthermore, Ref. [41] proposed that advanced representation learning and anomaly detection techniques, such as optimized enhanced stacked autoencoders, have been demonstrated to be highly effective in related financial risk domains like banking fraud detection. Similarly, Ref. [42] proposed that capturing temporal dynamics through deep learning is vital in these broader contexts, as demonstrated by the successful use of transactional-behavior-based hierarchical gated networks for credit card fraud detection. Ref. [43] proposed a tabular transformer to process multivariate time series and verified the efficiency on a synthetic credit card transaction dataset to detect fraud as well as on a real pollution dataset to predict atmospheric pollutant concentrations.

In summary, the majority of existing studies focus on tree-based approaches for tabular credit risk dataset and hence require extensive efforts on feature engineering. This

makes them less favorable for deployment in real-world setting due to time and resource constraints. While some studies have started to explore use of different neural networks to process raw transactional data or user behavior events often as a supplement to the heavily feature engineered approaches or require flattening of the sequential records into a 1D vector. In contrast, our proposed architecture aims to directly ingest the multivariate credit data as a 2D sequence. Furthermore, unlike many studies relying on proprietary financial data, our framework is validated entirely on publicly available datasets to ensure full reproducibility.

3. Dataset and Evaluation Metrics

3.1. AMEX Credit Default Dataset

The first dataset we use is the AMEX credit default dataset [31], the same dataset used by [33–35]. This dataset includes 458,913 records for training and a separate test set of 924,621 records. Each customer record has over 200 rich and multi-temporal features across five categories: delinquency, spend, payment, balance, and risk. There is also a default label that indicates whether the customer will default after an 18-month observation window.

To enable a direct comparison with existing research, we employ the official Kaggle competition evaluation metric (M). Because the true labels for the test dataset are permanently withheld by the competition hosts, this is the only metric available to evaluate our test set performance and objectively benchmark against state-of-the-art models. This metric combines two sub-measures: the normalized Gini coefficient (G) with the default rate at a 4% threshold (D), providing a comprehensive assessment of the model performance.

$$M = 0.5(G + D)$$

The normalized Gini coefficient is calculated as follows.

$$G = 2 \times \text{AUC} - 1$$

The normalized Gini coefficient can be viewed as a scaled version of the AUC. The AUC represents the light red area under the curve, ranging from 0 to 1 as shown in Figure 1. G always falls between -1 and 1 and a larger red area indicates a better score.

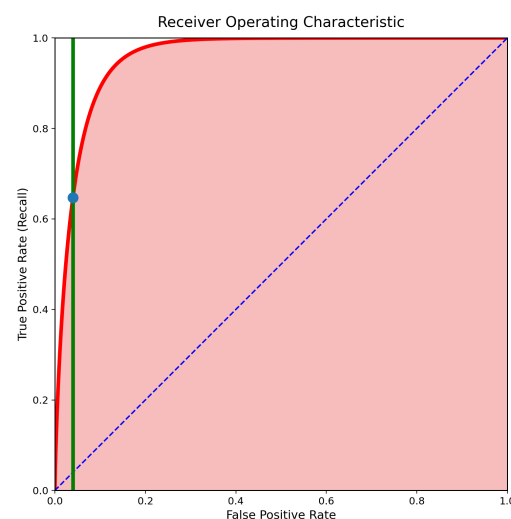


Figure 1. Receiver operating characteristic (ROC). The red curve represents the ROC curve, the light red shaded area represents the AUC, the blue dashed diagonal line indicates the random-classifier baseline, the green vertical line marks the 4% threshold, and the blue dot denotes the corresponding default-rate-captured point.

The default rate captured at 4% represents the true positive rate for a threshold set at 4% of the total sample size. This corresponds to the y-coordinate of the intersection between the green line and the red ROC curve and always falls between 0 and 1. A higher intersection point signifies a better score. Additionally, to account for down-sampling, negative labels in both sub-metrics are assigned a weight of 20.

3.2. Taiwan Bank Credit Default Dataset

To verify the proposed approach's generalization ability, we also tested it on a second dataset [27]. This dataset investigates credit card defaults among customers of a major Taiwanese bank. It comprises 30,000 observations, including 6636 cardholders (22.1%) who defaulted on their following month's payment [21]. The data include one dependent variable, "default payment for next month" which is binary (coded as Yes = 1, No = 0) and 23 independent variables [44]. These independent variables can be further categorized into demographics (first five variables), past payment status (next six variables), past billing amounts (following six variables) and past payment amounts (final six variables).

Since existing research using the Taiwan dataset lacks a consistent evaluation metric, this paper uses AUC for a comprehensive assessment across various classification thresholds. More details about AUC are included in Appendix A.

4. Method

We propose a transformer-based neural network that can process multi-variate time-series credit risk data without the need for elaborate feature engineering which renders it very favorable for deployment in real world settings.

4.1. Model Training and Prediction Pipeline Overview

Figure 2 shows the steps involved in the model training pipeline. The raw data are pre-processed and then split 5-fold for cross-validation purposes and then passed into the proposed model for training.

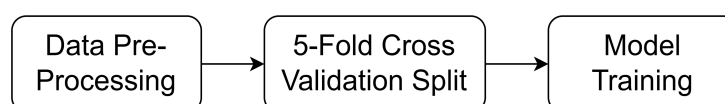


Figure 2. Model training pipeline.

Figure 3 shows the overview of the inference pipeline. The raw data are similarly pre-processed as per during training and then passed into the trained model to generate the prediction results. Note that the pre-processing step is dataset-specific.

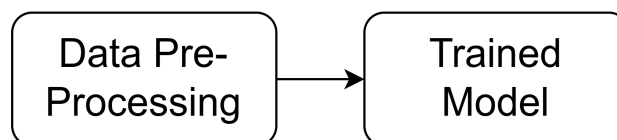


Figure 3. Inference pipeline.

4.2. Data Pre-Processing of the Amex Dataset

4.2.1. De-Noise

For the Amex dataset, an analysis of the features such as 'B_2' in Figure 4 shows that there is random uniform noise in the range [0, 0.01] for all features (probably introduced by the data provider).

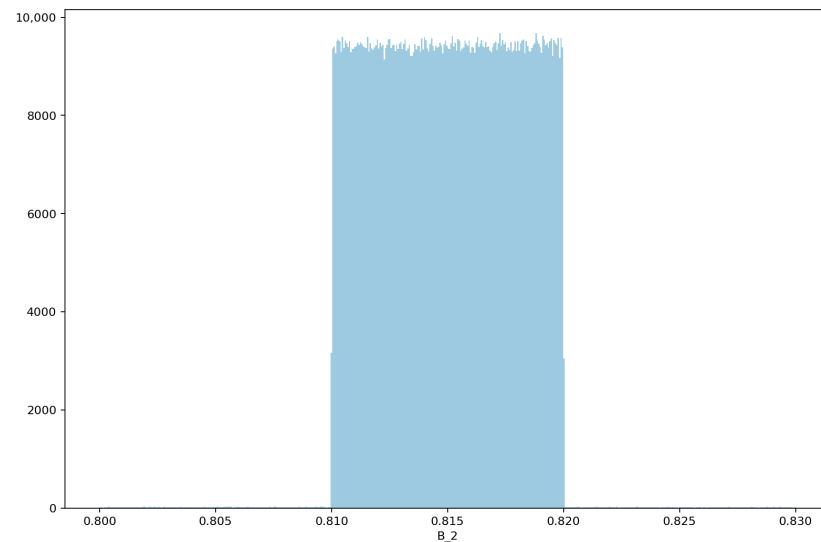


Figure 4. Data Analysis of Feature ‘B2’.

Hence, all feature columns in the raw data are de-noised by the following equation.

$$Feature = \text{floor}(Feature * 100)$$

4.2.2. Missing Value Handling

Null values are replaced with zeros.

4.2.3. One-Hot Encoding

One-hot encoding is performed for categorical features such as “B_30” and “B_38”.

4.3. Data Pre-Processing of the Taiwan Dataset

4.3.1. Train-Test Split

Owing to the lack of a predefined train–test split and evaluation metrics in the Taiwan dataset, prior studies such as [9,45], adopt their own evaluation metrics, making direct benchmarking challenging. To evaluate the proposed method’s generalization capability on this dataset, we employ a standard random train–test split with a 70%–30% ratio and use AUC as the evaluation metric to facilitate comparison.

4.3.2. Dataframe Reshaping

Three time series columns are generated from the following to adapt to the multivariate time series input framework of the proposed model:

- The six variables about the status of the past payments;
- The six variables about the amount of past bill statement;
- The six variables about the amount of paid bills.

4.4. Five-Fold Cross-Validation

We adopted 5-fold cross-validation in our model training. More details can be found in the Appendix A.

4.5. Proposed Transformer-Based Model

The transformer was first introduced in [46]. More details on transformer/attention/self-attention/multi-head-attention are given in the Appendix A.

Figure 5 shows the overall architecture of our proposed transformer-based model. It comprises five blocks, namely, ‘linear layer and normalization’, ‘positional encoding’,

‘transformer encoder’, ‘bidirectional GRU’ and ‘feedforward neural network’. The model takes in the processed input data from the data pre-processing module and the outputs a prediction probability of whether it is a default or not.

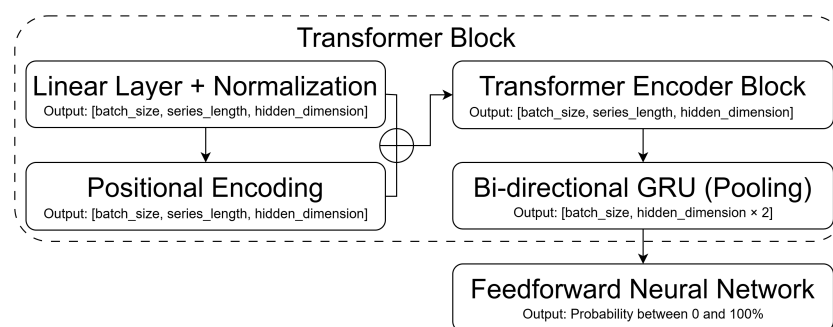


Figure 5. Our model.

The linear layer converts the preprocessed time series data into a fixed hidden dimension. Layer normalization is applied to maintain a consistent distribution and mitigate the internal covariate shift that can occur during training. Positional encoding is applied to incorporate temporal information of the relative position in the time series. It is computed as follows [46]:

$$PE_{(pos,2i)} = \sin(pos/10000^{\frac{2i}{d_{model}}})$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{\frac{2i}{d_{model}}})$$

Encoder-only transformer structure is adopted here, and the transformer encoder block consists of a stack of N transformer encoder layers specifically designed to capture the temporal dynamics and other latent features in the credit card data. Figure 6 shows the implementation of the encoder layer comprising a multi-head self-attention mechanism.

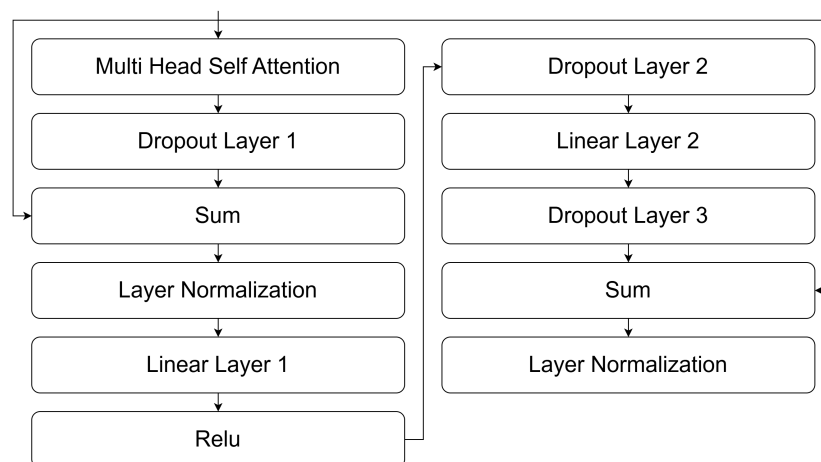


Figure 6. Encoder design of the transformer.

To account for the varying lengths of the time series data among the customers, a bidirectional GRU is employed as a specialized pooling mechanism for each customer.

The feedforward neural network block takes in the latent features from the previous step and generates the final prediction probability of whether the customer will default or not. The details are illustrated in Figure 7.

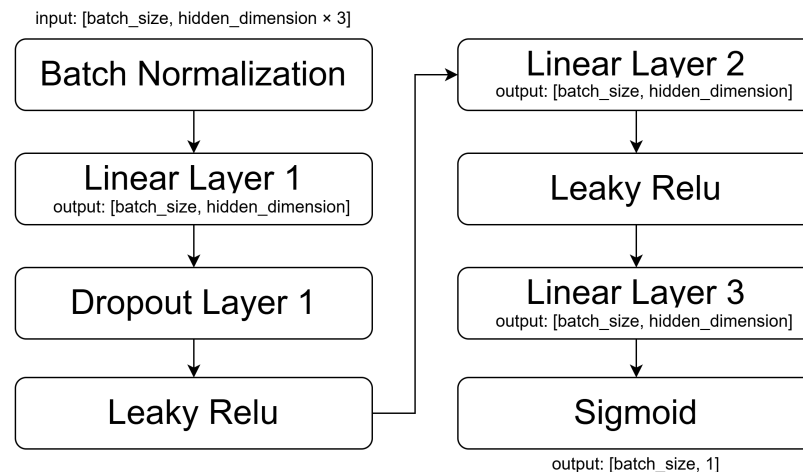


Figure 7. Feedforward neural network (FNN) block.

4.6. Optional Feature Augmentation Extension

Given that existing research and deployment for credit card default prediction rely heavily on engineered features from domain expertise and industry practice, users are often hesitant to relinquish these familiar inputs. To facilitate adoption, we designed an optional feature augmentation extension called the hidden feature block. The block is specifically built to process high-dimensional, user-crafted features, using its internal feedforward architecture (as shown in Figure 8) to distill them into a single, powerful, and dense feature vector. This parallel design is effective because it distills high-dimensional engineered features independently, preventing them from interfering with the transformer’s native temporal learning.

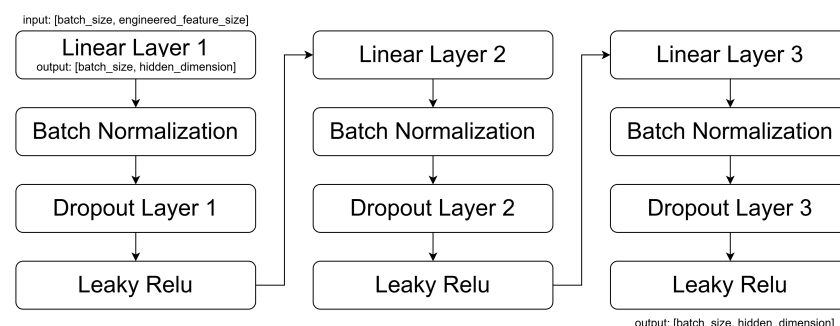


Figure 8. Hidden feature block.

Figure 9 illustrates how this module is integrated into the complete model architecture. The hidden feature block operates in parallel with the transformer block and is designed for flexibility; it is only activated when engineered features are available. Specifically, the raw 2D time-series data is processed exclusively by the transformer block to extract temporal patterns, while the pre-calculated engineered features are routed in parallel through the hidden feature block. The resulting dense latent representations from both distinct pathways are then concatenated before being passed into the final FNN block for prediction.

We engineered over 6000 features to be used in the experiments in this study. The results will be reported in the Experiment section. Feature engineering methods such as statistical transformation and knowledge distillation are applied, and details can be found in the Appendix A. It should be noted that feature engineering is very time-consuming, which motivates us to propose the transformer model without the need for feature engineering.

The model with feature engineering also requires significantly more computation resources, as shown in Table 1 below.

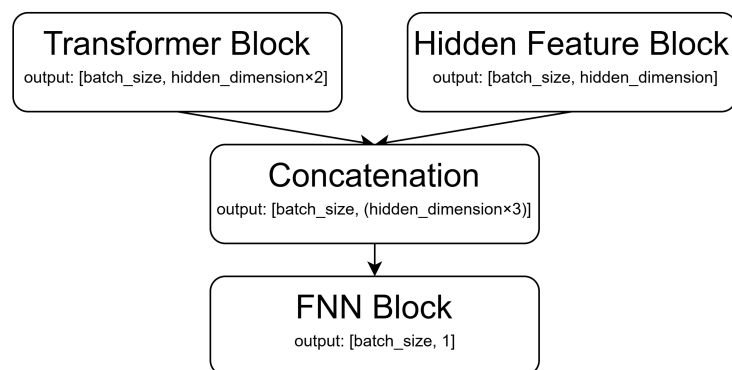


Figure 9. Our model with feature augmentation.

Table 1. Resource Usage on Amex Dataset.

	Without FE	With FE
Training GPU Memory	3.90 GB	3.95 GB
Training RAM	17.3 GB	41.6 GB
Training time	1.5 h	3.4 h
Inference GPU Memory	3.71 GB	3.72 GB
Inference RAM	26.8 GB	74.9 GB
Inference time	8.0 m	8.9 m

4.7. Evaluating Model Performance in an Ensemble

To evaluate the performance of our proposed transformer-based model in an ensemble, we integrate our model with the SOTA LightGBM in an ensemble framework. The proposed ensemble structure comprises the following sub-models:

- The proposed transformer-based model (without feature engineering);
- The proposed transformer-based model with feature augmentation extension to use engineered features;
- A SOTA LightGBM using engineered features.

Figure 10 shows the overview of the proposed ensemble architecture.

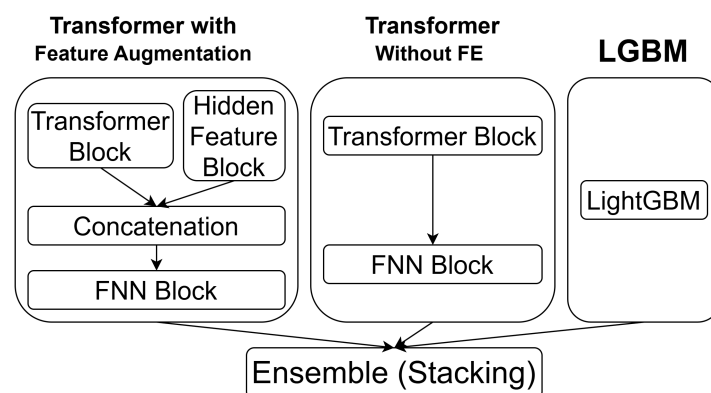


Figure 10. Ensemble architecture.

For the Taiwan dataset, due to its small size, a reduced ensemble structure is adopted, incorporating only the proposed transformer-based model and the SOTA LightGBM model.

5. Experiments

Experiment hardware/software setup and empirically best hyperparameter values for the LightGBM and Transformer approaches are described in the Appendix A.

5.1. Experimental Results for Amex Dataset

For the Amex dataset, the proposed approaches and benchmark models are assessed on the private test dataset. Table 2 shows that our proposed transformer-based model (i.e., without feature engineering) is better than the LightGBM benchmark model without feature engineering and achieves performance on par with SOTA LightGBM with feature engineering (Ref. [35] shown in Table 3).

Table 2. Results comparison.

Model	Test Set
LightGBM without FE	0.800
Transformer without FE (Ours)	0.802

Bold: highest score in the table.

Table 3 shows the performance comparison of the proposed transformer-based model and its optional feature augmentation extension (used here for hybrid comparison against feature-heavy benchmark) as well as performance in an ensemble framework. Our feature augmentation extension achieves a higher score than SOTA LightGBM with feature engineering [35] while contributing to better performance than existing ensemble approaches [33,34].

Table 3. Results on Amex Dataset with FE/Ensemble.

Model	Test Set
Transformer with Feature Augmentation (Ours)	<u>0.806</u>
LightGBM with FE [35]	0.801
Ensemble (Ours, incorporated into ensemble)	0.810
GBDT Ensemble [33]	0.809
DART Ensemble [34]	0.808

Underline: proposed non-ensemble model result. **Bold:** highest score in the table.

Paired *t*-tests across the five-fold validation results indicate statistically significant improvements at the 0.05 level. Detailed variance analysis and statistical testing results are provided in the Appendix A.

5.2. Experimental Results for Taiwan Dataset

The benchmark, the proposed and the ensemble models are evaluated on the Taiwan dataset and the results are summarised in Table 4. Note that for Taiwan dataset, we did not include feature engineering in all the models.

Table 4. Results on Taiwan Dataset.

Model	Test Set
LightGBM without FE	0.772
Transformer without FE (Ours)	<u>0.779</u>
Ensemble (Ours, incorporated into ensemble)	0.783

Underline: proposed non-ensemble model result. **Bold:** highest score in the table.

The proposed transformer-based model surpasses the benchmark LightGBM model in performance. Additionally, the proposed ensemble model enhances the prediction

accuracy beyond the standalone model. The performance of these models is consistent with the results on the Amex dataset, testifying to the generalization ability of the proposed transformer-based model.

6. Path to Deployment and Practical Impact

Our proposed model directly addresses a critical bottleneck in deploying credit risk models. Traditional methods, including SOTA academic ensembles such as [32,33] and commercial systems e.g., Upstart [47], Zest AI [48], often rely on thousands of manually handcrafted features. This practice slows development, complicates maintenance and hinders scalability in real-world financial systems.

By processing raw multivariate credit data as a 2D time series, our transformer-based model eliminates the arduous feature engineering efforts. This approach yields immediate practical benefits, including accelerated development cycles, drastically reduced maintenance overhead, lower inference latency (Table 1), and streamlined integration into existing credit scoring pipelines. The lightweight architecture is designed for deployment, and its effectiveness is validated across the diverse AMEX and Taiwan Bank datasets, showing its promise for different banking contexts.

We do recognize that existing users who are accustomed to engineered features from domain expertise and industry practice will be reluctant to give up engineered features completely. We therefore built a feature augmentation extension to the proposed model to facilitate adoption of our model by existing users. Our model is user-friendly and can leverage hybrid learning to support both user-crafted and model-learned features to improve model performance and facilitate current user buy-in. Our proposed model with the feature augmentation extension also helps to bridge the gap until current users are comfortable to transition to more advanced, feature-learning paradigms without abandoning their existing highly interpretable, regulatory-compliant workflows.

This work, therefore, lays the groundwork for practical adoption in production credit systems. Future efforts to enhance interpretability will further support its adoption in regulated environments, solidifying a clear path from emerging application to deployed solution.

7. Conclusions and Future Work

This paper presents a transformer-based neural network for credit card default prediction that processes raw 2D time-series credit data without the need for labor-intensive feature engineering. The proposed model outperforms LightGBM baselines and achieves performance comparable to SOTA methods with engineered features, demonstrating strong generalization across two real-world datasets.

In addition, we have incorporated a feature augmentation extension to the proposed model to facilitate adoption by existing users who are accustomed to engineered features from domain expertise and industry practice. Hence, our model is user-friendly and supports hybrid learning to support both user crafted and model learned features to improve model performance and facilitate deployment. We also evaluated our proposed model in an ensemble framework with LightGBM, and its performance surpasses existing ensemble approaches. These results highlight the model's robustness, simplicity, generalizability and suitability for scalable deployment in financial applications.

For future work, we will explore mitigating potential dataset biases and scalability limits of self-attention on long sequences, as well as improving model interpretability and integrating transformer-encoded features into tree-based frameworks for enhanced performance and compliance readiness.

Author Contributions: Conceptualization, Z.H.; methodology, Z.H. and C.K.Y.; validation, Z.H.; formal analysis, Z.H. and C.K.Y.; investigation, Z.H.; resources, C.K.Y.; data curation, Z.H.; writing—original draft preparation, Z.H.; writing—review and editing, C.K.Y.; visualization, Z.H.; supervision, C.K.Y.; project administration, Z.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available in a publicly accessible repository—[31]; third party data—[27]. The code and pre-processing scripts for this study are available in a public GitHub repository: <https://github.com/hzqn1234/A-Transformer-Based-Neural-Network-to-Predict-Credit-Card-Default> (accessed on 5 April 2026).

Conflicts of Interest: The authors declare no conflicts of interest. No external funders had any role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Appendix A

Appendix A.1. AUC for Taiwan Dataset

Since existing research using the Taiwan dataset lacks a consistent evaluation metric (e.g., [21,44,45]), this paper utilizes AUC for a comprehensive evaluation across various classification thresholds. AUC essentially represents the probability that a randomly chosen positive instance ranks higher than a random negative instance by the model [45]. Ref. [6] highlights AUC as the preferred metric for credit risk prediction. Additionally, in broader machine learning literature, AUC is also widely recognized as a leading metric for evaluating binary classifiers [49].

Appendix A.2. Five-Fold Cross Validation

Best practices in applying machine learning recommend conducting cross-validation on the training set [50]. Cross-validation helps ensure a balance between bias and variance and can be used to fine-tune hyperparameters or apply regularization techniques to prevent under-fitting or over-fitting before deploying the model on the testing set. Although cross-validation provides an estimate of the prediction error averaged over hypothetical datasets from the same distribution rather than the actual prediction error [51], it can still serve as a valuable benchmark for the primary evaluation conducted on the testing set which has not been encountered by the model.

Following [34,52], a stratified 5-fold cross-validation strategy is employed in this study using the scikit-learn package [53]. It is important to note that the 5-fold split is based on customer_ID, ensuring that all rows associated with the same customer remain within the same fold, thus preventing any risk of data leakage. Additionally, the folds are created to maintain the proportion of samples with positive and negative label classes. This iterative process repeatedly trains and validates the model on different data splits, resulting in more reliable and generalizable predictions. The training data pipeline generates 5 models, one for each fold, and these models are then used in the prediction pipeline. The final prediction is obtained by averaging the predictions of the 5 models.

Appendix A.3. Transformer

Ref. [46] introduced the transformer, the first sequence transduction model based entirely on attention mechanisms, which replaced the recurrent layers typically used in the encoder-decoder architectures with multi-headed self-attention.

Attention

Attention is a mechanism that determines the importance of different parts of input data for producing an output, by comparing a query to multiple keys and assigning weights to the corresponding values based on their similarity. By representing (queries, keys, values) as matrices (Q, K, V), and the dimension of the key and query vectors as $\sqrt{d_k}$, the attention calculation is given as follows [46]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Self-Attention

In particular, self-attention refers to where all keys, values and queries come from the same input sequence. This self-attention mechanism effectively identifies the latent relationships between the transactions, revealing the hidden patterns that enhance the prediction performance.

Multi-Head Attention

Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. The calculations are as follows [46]:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Appendix A.4. Experimental Setup

Both the proposed standalone and ensemble models were trained and tested on a machine equipped with an AMD Ryzen Threadripper 3960X 24-Core Processor and two Nvidia RTX 3090 GPUs.

Python version 3.9.19 was used for all the experiments. The main package versions are shown in Table A1.

Table A1. Python Packages.

Python Package	Version
LightGBM	3.3.5
scikit-learn	1.1.2
torch	1.10.1 + cu111
pandas	1.5.0
numpy	1.26.0

Appendix A.5. Hyperparameters

To optimize the model architectures and evaluate parameter sensitivity, hyperparameter tuning was conducted using an empirical grid search evaluated across the cross-validation folds.

Since the Taiwan dataset has significantly fewer records and features compared to the Amex dataset, a less complex neural network is used for the prediction task to prevent overfitting. Hence, we reduced the encoder layers, used a smaller hidden dimension and a different learning rate scheduler, namely cosine annealing, which are suited for shorter training runs. Table A2 shows the key hyperparameter values for the proposed transformer-based model which have achieved the best empirical performance on both datasets.

Table A2. Transformer hyperparameters for both datasets.

	Amex	Taiwan
hidden_dim	256	32
encoder_layer_num	3	2
attn_heads	32	2
ff_dim	256	4
encoder_dropout_rate	0.05	0.005
HF_dropout_rate	0.01	0.005
FNN_dropout_rate	0.05	0.025
Learning Rate	8.25×10^{-5}	9×10^{-4}
Learning Rate Scheduler	Step Function	Cosine Annealing
Optimizer	Adam	Adam
Training Epochs	12	15
Batch Size	256	512

To ensure transparency, fair comparison, and reproducibility, Table A3 shows the key hyperparameter values for the benchmark LightGBM model which have achieved the best empirical performance on both datasets.

Table A3. LightGBM hyperparameters for both datasets.

	Amex	Taiwan
learning_rate	0.035	0.035
bagging_freq	5	5
bagging_fraction	0.75	0.75
feature_fraction	0.05	0.05
metric	binary_logloss	binary_logloss
num_leaves	64	64
max_depth	−1	−1
lambda_l1	30	30
lambda_l2	24	24
rounds	4500	1500
early_stopping_rounds	100	100

Appendix A.6. Five-Fold Cross Validation Results for Amex Dataset

Because the ground truth labels for the AMEX test dataset are permanently withheld by the dataset providers, it is not possible to conduct paired statistical tests directly on the official test set. Therefore, we report fold-level validation scores across the 5 local cross-validation folds in Table A4. To assess whether the observed improvements are consistent across folds, we conducted paired *t*-tests on the fold-level validation scores. The standalone transformer without feature engineering was compared against LightGBM without feature engineering, while the final ensemble was compared against the standalone transformer without feature engineering. Both paired comparisons showed statistically significant improvements at the 0.05 level. These results provide additional evidence that the observed gains are consistent across local validation folds, although the official AMEX test set comparison remains based on the competition metric due to withheld labels.

Table A4. Fold-level validation scores for the AMEX dataset.

	LightGBM Without FE	Transformer Without FE	Ensemble
Fold 0	0.792345	0.794331	0.806416
Fold 1	0.783720	0.784336	0.796679
Fold 2	0.788226	0.789010	0.799645
Fold 3	0.782836	0.787296	0.796386
Fold 4	0.790248	0.792301	0.802113
Mean $\pm$ Std Dev	0.787475 $\pm$ 0.003677	0.789455 $\pm$ 0.003548	0.800248 $\pm$ 0.003729

Appendix A.7. Feature Engineering (Amex)

Feature engineering is needed to expand the raw data features to capture more dynamic dependency between the multiple variables. Several feature engineering methods have been adopted to extract features which can provide domain-specific insights which will then lead to a more robust and accurate model. These methods are detailed as follows:

Statistical Transformation

Minimum, maximum, sum, difference, rank (sorted by value), time-rank (sorted by time order), “last-n” (last n records) and standard deviation of the denoised series data (such as spending amount) within a defined time windows (e.g., 3 or 6 months) are calculated to capture the potential financial trends.

Knowledge Distillation

Knowledge distillation (KD) features are created by running a small-size LightGBM model [54] on only the series data and predicting the target variable for each month. In machine learning, knowledge distillation acts like a teacher tutoring a student. A complex, pre-trained model (the teacher) efficiently guides a smaller model (the student) to grasp intricate relationships between the input data and output categories, while the student remains nimble and efficient [55]. The prediction performance of the KD model also offers a baseline for subsequent model evaluations.

LGBM Greedy-Find-Bins Transformation

Ref. [54] introduced a groundbreaking GBDT algorithm named LightGBM, incorporating two innovative techniques: gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB). GOSS addresses the challenge of handling large datasets, while EFB is designed to manage a high volume of features. Experimental results align with theoretical expectations, demonstrating that LightGBM, leveraging GOSS and EFB, significantly surpasses XGBoost and stochastic gradient boosting (SGB) in both computational speed and memory efficiency. A crucial component of EFB is the greedy-find-bins transformation, which is a greedy algorithm that achieves a rather good approximation ratio in substitution of optimal bundling [56].

References

- Kim, H.; Cho, H. *An Empirical Study on Credit Card Loan Delinquency in Korea*; KAIST College of Business Working Paper Series; KAIST College of Business: Seoul, Republic of Korea, 2015.
- Basel Committee on Banking Supervision; Bank for International Settlements. *Principles for the Management of Credit Risk*; Bank for International Settlements: Basel, Switzerland, 2000.
- Syed, A.M.; Bawazir, H.S. Recent trends in business financial risk—A bibliometric analysis. *Cogent Econ. Financ.* **2021**, *9*, 1913877. [CrossRef]
- Dzelihodzic, A.; Donko, D. Data mining techniques for credit risk assessment task. *Recent Adv. Comput. Sci. Appl.* **2013**, *6*, 105–110.
- Thomas, L.; Crook, J.; Edelman, D. *Credit Scoring and Its Applications*; SIAM: Philadelphia, PA, USA, 2017.
- Noriega, J.P.; Rivera, L.A.; Herrera, J.A. Machine learning for credit risk prediction: A systematic literature review. *Data* **2023**, *8*, 169. [CrossRef]

7. Leo, M.; Sharma, S.; Maddulety, K. Machine learning in banking risk management: A literature review. *Risks* **2019**, *7*, 29. [CrossRef]
8. Berloco, C.; Argiento, R.; Montagna, S. Forecasting short-term defaults of firms in a commercial network via Bayesian spatial and spatio-temporal methods. *Int. J. Forecast.* **2023**, *39*, 1065–1077. [CrossRef]
9. Rosdi, N.F.S.; Ibrahim, N.S.; Shamsudin, I.H.; Mutalib, S.; Abdul-Rahman, S. A Provisional Study of Data Mining Classification Algorithms in Predicting Credit Card Defaulters Using Weka Tools. In *Proceedings of the 2023 IEEE 8th International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*; IEEE: New York, NY, USA, 2023; pp. 1–5.
10. Liu, X.; Li, Y.; Jiang, C.; Wang, Z.; Zhao, F.; Wang, J. Attentive feature fusion for credit default prediction. In *Proceedings of the 2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*; IEEE: New York, NY, USA, 2022; pp. 816–821.
11. Chen, R.; Ju, C.; Shen Tu, F. A Credit Scoring Ensemble Framework using Adaboost and Multi-layer Ensemble Classification. In *Proceedings of the 2022 International Conference on Pattern Recognition and Intelligent Systems, Hammamet, Tunisia, 24–26 March 2022*; pp. 72–79.
12. Abellán, J.; Castellano, J.G. A comparative study on base classifiers in ensemble methods for credit scoring. *Expert Syst. Appl.* **2017**, *73*, 1–10. [CrossRef]
13. Hu, Z.; Yeo, C.K. A Lightweight Neural Network with Transformer to Predict Credit Default. In *Proceedings of the 2024 IEEE Conference on Artificial Intelligence (CAI)*; IEEE: New York, NY, USA, 2024; pp. 29–30.
14. Butaru, F.; Chen, Q.; Clark, B.; Das, S.; Lo, A.W.; Siddique, A. Risk and risk management in the credit card industry. *J. Bank. Financ.* **2016**, *72*, 218–239. [CrossRef]
15. Crook, J.N.; Edelman, D.B.; Thomas, L.C. Recent developments in consumer credit risk assessment. *Eur. J. Oper. Res.* **2007**, *183*, 1447–1465. [CrossRef]
16. Alvi, J.; Arif, I.; Nizam, K. Advancing financial resilience: A systematic review of default prediction models and future directions in credit risk management. *Heliyon* **2024**, *10*, e39770. [CrossRef]
17. Altman, E.I. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *J. Financ.* **1968**, *23*, 589–609. [CrossRef]
18. Wiginton, J.C. A note on the comparison of logit and discriminant models of consumer credit behavior. *J. Financ. Quant. Anal.* **1980**, *15*, 757–770. [CrossRef]
19. Machado, M.R.; Karray, S. Assessing credit risk of commercial customers using hybrid machine learning algorithms. *Expert Syst. Appl.* **2022**, *200*, 116889. [CrossRef]
20. Moula, F.E.; Guotai, C.; Abedin, M.Z. Credit default prediction modeling: An application of support vector machine. *Risk Manag.* **2017**, *19*, 158–187. [CrossRef]
21. Yeh, I.C.; Lien, C. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Syst. Appl.* **2009**, *36*, 2473–2480. [CrossRef]
22. Lessmann, S.; Baesens, B.; Seow, H.V.; Thomas, L.C. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *Eur. J. Oper. Res.* **2015**, *247*, 124–136. [CrossRef]
23. Siphuma, E.; van Zyl, T. Enhancing Credit Risk Assessment Through Transformer-Based Machine Learning Models. In *Artificial Intelligence Research*; Springer: Cham, Switzerland, 2024; p. 124.
24. Wang, M.; Zhou, L.; Meng, Q.; Kong, Y.; Sun, J. Credit risk prediction network based on semantic feature transformer and CNN. In *Proceedings of the 2023 IEEE 6th International Conference on Electronic Information and Communication Technology (ICEICT)*; IEEE: New York, NY, USA, 2023; pp. 723–728.
25. Wang, Y.; Xu, Z.; Yao, Y.; Liu, J.; Lin, J. Leveraging Convolutional Neural Network-Transformer Synergy for Predictive Modeling in Risk-Based Applications. In *Proceedings of the 2024 4th International Conference on Electronic Information Engineering and Computer Communication (EIECC)*; IEEE: New York, NY, USA, 2024; pp. 1565–1570.
26. Hofmann, H. Statlog (German Credit Data). UCI Machine Learning Repository. 1994. Available online: <https://archive.ics.uci.edu/dataset/144/statlog+german+credit+data> (accessed on 5 April 2026). [CrossRef]
27. Yeh, I.C. Default of Credit Card Clients. UCI Machine Learning Repository. 2009. Available online: <https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients> (accessed on 5 April 2026). [CrossRef]
28. Grinsztajn, L.; Oyallon, E.; Varoquaux, G. Why do tree-based models still outperform deep learning on typical tabular data? *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 507–520.
29. Uddin, S.; Lu, H. Confirming the statistically significant superiority of tree-based machine learning algorithms over their counterparts for tabular data. *PLoS ONE* **2024**, *19*, e0301541. [CrossRef]
30. Morelia, A.L.; Inversion; KirillOdintsov; Kotek, M. Home Credit Default Risk. Kaggle. 2018. Available online: <https://kaggle.com/competitions/home-credit-default-risk> (accessed on 5 April 2026).
31. Howard, A.; AritraAmex; Xu, D.; Vashani, H.; Inversion; Negin; Dane, S. American Express-Default Prediction. Kaggle. 2022. Available online: <https://kaggle.com/competitions/amex-default-prediction> (accessed on 5 April 2026).

32. Hlongwane, R.; Ramaboa, K.K.; Mongwe, W. Enhancing credit scoring accuracy with a comprehensive evaluation of alternative data. *PLoS ONE* **2024**, *19*, e0303566. [CrossRef]
33. Guo, K.; Luo, S.; Liang, M.; Zhang, Z.; Yang, H.; Wang, Y.; Zhou, Y. Credit default prediction on time-series behavioral data using ensemble models. In *Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN)*; IEEE: New York, NY, USA, 2023; pp. 1–9.
34. Wang, H. Forecasting credit card defaults using light gradient boosting machine with dart algorithm. In *Proceedings of the 2022 5th Artificial Intelligence and Cloud Computing Conference*, Osaka, Japan, 7–19 December 2022; pp. 207–212.
35. Gan, Z.; Qiu, J.; Li, F.; Liang, Q. A light-gbm based default prediction method for american express. In *Proceedings of the 2nd International Conference on Information Economy, Data Modeling and Cloud Computing ICIDC*, Nanchang, China, 2–4 June 2023.
36. Babaev, D.; Savchenko, M.; Tuzhilin, A.; Umerenkov, D. Et-rnn: Applying deep learning to credit loan applications. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage, AK, USA, 4–8 August 2019; pp. 2183–2190.
37. Sukharev, I.; Shumovskaia, V.; Fedyanin, K.; Panov, M.; Berestnev, D. Ews-gcn: Edge weight-shared graph convolutional network for transactional banking data. In *Proceedings of the 2020 IEEE International Conference on Data Mining (ICDM)*; IEEE: New York, NY, USA, 2020; pp. 1268–1273.
38. Ky, S.; Lee, J.H.; Na, K. Incorporating BERT-based NLP and Transformer for An Ensemble Model and its Application to Personal Credit Prediction. *Smart Media J.* **2024**, *13*, 9–15.
39. Wang, C.; Xiao, Z. A deep learning approach for credit scoring using feature embedded transformer. *Appl. Sci.* **2022**, *12*, 10995. [CrossRef]
40. Yuan, Q.; Liu, Y.; Tang, Y.; Chen, X.; Zheng, X.; He, Q.; Ao, X. Dynamic Graph Learning with Static Relations for Credit Risk Assessment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Philadelphia, PA, USA, 25 February–4 March 2025; Volume 39, pp. 13133–13141.
41. Kumar, D.; Anitha, P.; Murugachandavel, J.; Jeevitha, S.; Bhuvanesh, A.; Pawar, P.P. Banking fraud detection using optimized enhanced stacked autoencoder approach. *Secur. Priv.* **2025**, *8*, e70054. [CrossRef]
42. Xie, Y.; Zhou, M.; Liu, G.; Wei, L.; Zhu, H.; De Meo, P. A transactional-behavior-based hierarchical gated network for credit card fraud detection. *IEEE/CAA J. Autom. Sin.* **2025**, *12*, 1489–1503. [CrossRef]
43. Padhi, I.; Schiff, Y.; Melnyk, I.; Rigotti, M.; Mroueh, Y.; Dognin, P.; Ross, J.; Nair, R.; Altman, E. Tabular transformers for modeling multivariate time series. In *Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE: New York, NY, USA, 2021; pp. 3565–3569.
44. Subasi, A.; Cankurt, S. Prediction of default payment of credit card clients using Data Mining Techniques. In *Proceedings of the 2019 International Engineering Conference (IEC)*; IEEE: New York, NY, USA, 2019; pp. 115–120.
45. Sharma, D.; Kang, S.S. Hybrid model for detection of frauds in credit cards. In *Proceedings of the 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*; IEEE: New York, NY, USA, 2022; pp. 70–77.
46. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
47. Upstart. Our Story. 2025. Available online: <https://www.upstart.com/our-story> (accessed on 23 July 2025).
48. Zest AI. Machine Learning 101 for Credit Underwriting. Technical Report, Zest AI. 2024. Available online: https://www.zest.ai/wp-content/uploads/2024/08/647926692388d720d00a4ed2_E-book-Machine-Learning-101.pdf (accessed on 23 July 2025).
49. Fawcett, T. ROC graphs: Notes and practical considerations for researchers. *Mach. Learn.* **2004**, *31*, 1–38.
50. Domingos, P. A few useful things to know about machine learning. *Commun. ACM* **2012**, *55*, 78–87. [CrossRef]
51. Bates, S.; Hastie, T.; Tibshirani, R. Cross-validation: What does it estimate and how well does it do it? *J. Am. Stat. Assoc.* **2024**, *119*, 1434–1445. [CrossRef]
52. Zhu, L.; Qiu, D.; Ergu, D.; Ying, C.; Liu, K. A study on predicting loan default based on the random forest algorithm. *Procedia Comput. Sci.* **2019**, *162*, 503–513. [CrossRef]
53. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
54. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. Lightgbm: A highly efficient gradient boosting decision tree. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 3146–3154.
55. Gou, J.; Yu, B.; Maybank, S.J.; Tao, D. Knowledge distillation: A survey. *Int. J. Comput. Vis.* **2021**, *129*, 1789–1819. [CrossRef]
56. Shi, Y.; Ke, G.; Soukhavong, D.; Lamb, J.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; et al. R Package, version 4.6.0; lightgbm: Light Gradient Boosting Machine, 2025. Available online: <https://github.com/Microsoft/LightGBM> (accessed on 5 April 2026).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.