STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **15**, April 2026, pp 3174–3193.
Published online in International Academic Press (www.IAPress.org)

Cost-Aware Deep Neural Network for Credit Card Fraud Detection under Chronological Evaluation

Abdellatif Elbadraoui[1], Yassine Mouhssine[2], Abdelkader El Alaoui[3], Said Ouatik El Alaoui[1,*]

[1] *Engineering Sciences Laboratory, National School of Applied Sciences, Ibn Tofail University, Kenitra, Morocco*

[2] *Laboratory of Analyzing and Modeling Systems for Decision Support (LAMSAD), Hassan I University, Settat, Morocco*

[3] *Rabat Business School, International University of Rabat (UIR), Rabat, Morocco*

Abstract Credit card fraud detection remains challenging due to extreme class imbalance, evolving fraud patterns, and asymmetric misclassification costs. This paper presents a *deployment-oriented evaluation and decision-calibration framework* for transaction-level fraud detection on the public `creditcard.csv` dataset, assessed under a strictly chronological train–validation–test protocol that mirrors real-world operation. Our contribution lies in *evaluation design and cost-aware decision calibration* rather than architectural novelty, focusing on how probabilistic model outputs are translated into operational decisions under explicit cost constraints.

A standard feed-forward deep neural network (DNN) is trained on the numerical features using a class-weighted binary cross-entropy loss, with early stopping guided by validation AUC–PR. At deployment time, the decision threshold is selected on the validation window by minimizing an empirical cost function that penalizes false negatives more than false positives. On the held-out test set, the proposed pipeline achieves a ROC–AUC of 0.9489 and a PR–AUC of 0.7813.

We show that decision policy choice strongly affects operational outcomes: naive thresholding yields excessive false alarms, whereas validation-based cost-sensitive calibration substantially reduces expected loss. Under $C_{\mathrm{FN}} = 10$ and $C_{\mathrm{FP}} = 1$, the cost-optimal threshold yields an expected test cost of 190. Comparisons with logistic regression, random forest, and XGBoost under identical preprocessing, temporal splitting, and decision calibration show that tree-based ensembles remain highly competitive, while the evaluated DNN achieves comparable cost and precision–recall performance. Overall, the results highlight the importance of combining chronological evaluation with explicit cost-sensitive thresholding for practical fraud detection under severe class imbalance.

Keywords Credit-card fraud detection; Cost-sensitive decision theory; Bayes minimum risk; Empirical risk minimization; Weighted cross-entropy; Threshold optimization; Precision–recall analysis; Chronological evaluation

DOI: 10.19139/soic-2310-5070-3362

1. Introduction

The rapid growth of e-commerce, mobile banking and contactless payments has led to an unprecedented increase in the volume of credit card transactions processed by financial institutions. While this trend brings significant convenience for customers and merchants, it also amplifies exposure to fraudulent activity and associated financial losses. Even when fraud rates remain extremely low in relative terms, the sheer scale of daily transactions implies that small changes in detection performance can translate into

*Correspondence to: A. Elbadraoui (Email: abdellatif.elbadraoui@uit.ac.ma). Engineering Sciences Laboratory, National School of Applied Sciences, Ibn Tofail University, Kenitra, Morocco.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

substantial monetary impact and reputational risk for card issuers and payment processors [10, 11]. More broadly, the financial sector is undergoing a progressive integration of artificial intelligence techniques into risk management, compliance and sustainable finance, as highlighted by recent bibliometric analyses on AI in financial systems [22]. Credit card fraud detection is a core component of this ecosystem, since it directly affects trust in digital payment infrastructures.

From a machine learning perspective, credit card fraud detection is particularly challenging for several reasons. First, transactional datasets are characterised by *extreme class imbalance*: confirmed frauds typically represent less than 0.2% of all operations, which causes standard classifiers to favour the majority (legitimate) class and achieve deceptively high accuracy while missing rare but costly fraudulent events [8, 14, 24]. Under such conditions, accuracy and ROC-based metrics can be misleading, and precision–recall analysis is generally more informative for assessing performance on the minority class [25]. Second, fraud patterns evolve over time as adversaries probe and adapt to existing countermeasures, leading to *concept drift* and progressive degradation of models trained on historical data [9]. Third, fraud detection systems must operate under strict latency and resource constraints, and each decision involves an inherently *cost-sensitive* trade-off between wrongly blocking legitimate transactions (false positives) and failing to prevent fraudulent ones (false negatives). Theoretical work on cost-sensitive learning [23] and applied studies in credit card fraud [12, 13] emphasise that decision thresholds should be chosen with explicit reference to misclassification costs rather than generic, cost-agnostic criteria. Importantly, while ROC-AUC and PR-AUC measure threshold-free ranking quality, the deployed impact of a fraud model is ultimately governed by the *operating point* (decision policy), which determines the realised false-alarm workload and missed-fraud exposure.

1.1. Deep learning for credit card fraud detection

Traditional industrial solutions are based on rule engines designed by domain experts, sometimes complemented with classical classifiers such as logistic regression, decision trees, random forests or gradient-boosting ensembles [11, 14]. These models can perform well when appropriate features and imbalance-handling strategies are used, but they often rely on fixed thresholds and generic performance metrics (e.g. accuracy, AUROC), and they may struggle to capture complex non-linear interactions between high-dimensional transaction attributes [11, 14]. Moreover, many studies do not explicitly connect predicted probabilities to a cost-sensitive decision rule, despite the asymmetric nature of fraud-related losses.

In recent years, deep learning has been investigated as a way to better exploit the structure of transactional data. Fully connected deep neural networks (DNNs) have been applied to tabular fraud data and shown to achieve competitive or superior recall for the fraud class compared to traditional baselines, especially when combined with suitable regularisation, class-imbalance handling and careful metric selection [15, 16, 24]. Recurrent architectures such as Long Short-Term Memory (LSTM) networks further incorporate sequential information by modelling transaction histories per cardholder, and have been reported to improve detection performance over static models in several benchmarks [17]. Complementary lines of work have explored unsupervised and semi-supervised anomaly detection based on One-Class SVM, Local Outlier Factor, autoencoders and GAN variants [10, 20], including recent performance studies dedicated specifically to unsupervised credit card fraud detection [21]. Despite these advances, many deep-learning approaches on the public `creditcard.csv` dataset are still evaluated under random train–test splits and report results under default or cost-agnostic threshold choices (often $\tau = 0.5$), which can overestimate deployment performance under temporal non-stationarity and asymmetric operational costs.

1.2. Contributions

In this paper, we investigate a deep neural network framework for credit card fraud detection on the widely used `creditcard.csv` dataset, which consists of 284,807 real European card transactions with 492 labelled frauds (fraud prevalence $\approx 0.173\%$). Each transaction is represented by 28 anonymised principal component features, together with the original transaction time and amount. To better reflect a realistic

deployment scenario, we adopt a chronological train–validation–test split based on the transaction time, training models on earlier data and evaluating them on later transactions. Our goal is not only to report threshold-free ranking performance, but also to study how different operating-point policies translate model scores into operational outcomes under explicit asymmetric costs.

In particular, we do not claim a novel neural architecture; rather, the contribution of this work lies in the deployment-oriented evaluation protocol and the explicit cost-aware calibration of decision thresholds applied to standard deep learning models.

The main contributions of this work can be summarised as follows:

- We construct a *chronological evaluation protocol* on the `creditcard.csv` dataset, avoiding random shuffling and thus approximating the real-world situation where models are trained on past data and deployed on future transactions.
- We design and train a fully connected DNN tailored to the tabular credit card fraud setting, combining standardised temporal and monetary features with anonymised components, and we incorporate class-weighted learning to mitigate the effect of extreme class imbalance.
- We integrate a simple yet practically meaningful *cost-aware threshold selection* procedure, in which the operating decision threshold is chosen on the validation set to minimise an empirical cost function that differentiates between false positives and false negatives, in line with cost-sensitive learning principles [23, 12, 13].
- We provide a systematic comparison between the proposed DNN and strong supervised baselines (logistic regression and tree-based ensembles), under identical preprocessing, temporal splitting and cost-aware calibration, and we analyse the impact of the imbalance-handling and operating-point choice on precision–recall behaviour and expected loss.
- Taken together with earlier work on unsupervised anomaly detection for credit card fraud [21] and on the broader role of AI in financial systems [22], this study contributes a complementary, supervised and cost-sensitive perspective on transaction-level fraud detection.

1.3. Organization

The remainder of the paper is organised as follows. In Section 2, we review related work on credit card fraud detection, with emphasis on deep learning, anomaly-detection methods and cost-sensitive learning. Section 3 presents the proposed DNN-based framework, including data preprocessing, class-imbalance handling, model architecture and cost-aware threshold optimisation. Experimental results on the `creditcard.csv` dataset, together with comparisons to baseline methods, are reported in Section 4. Finally, Section 5 concludes the paper and outlines directions for future research.

2. Related work

Credit card fraud detection has evolved from expert-designed rule engines to machine-learning pipelines that combine supervised learning, anomaly detection, and decision policies driven by operational constraints. Because fraud datasets are severely imbalanced and fraud patterns evolve over time, improvements in threshold-free discrimination metrics do not always translate into deployable gains. Surveys and reviews highlight that practical effectiveness depends on feature quality, imbalance handling, temporal validation, and post-training decision rules [2, 11, 14]. Recent survey articles further emphasize that evaluation protocol design and decision calibration are as critical as model choice when deploying fraud detection systems in real banking and payment environments [16]. This section reviews the literature most relevant to our setting and to the empirical findings in Section 4: (i) classical supervised learning and strong tree-based baselines, (ii) deep learning for transactional fraud and real-time e-payment, (iii) unsupervised and hybrid anomaly-detection frameworks, and (iv) cost-sensitive learning, threshold selection, and evaluation under imbalance. Overall, recent contributions published between 2021 and 2025

increasingly emphasize deployment-oriented evaluation, cost-aware decision rules, and temporal validation as key requirements for reliable fraud detection systems.

2.1. Classical supervised learning and strong baselines

Industrial fraud systems historically relied on rule-based engines designed by domain experts. While rules are transparent and auditable, they are costly to maintain and often fail to capture complex nonlinear interactions between transaction attributes. As a result, supervised machine-learning methods such as logistic regression, decision trees, random forests, SVMs, and gradient-boosting ensembles have become standard components of modern fraud detection pipelines [2, 11].

Benchmark-oriented studies and applied reviews consistently report that tree ensembles (random forests, gradient-boosting trees such as XGBoost/CatBoost) are highly competitive on tabular fraud data, often producing strong precision–recall trade-offs when properly tuned [11, 14]. Beyond model choice, several authors emphasise that evaluation design and probability calibration can strongly impact measured performance in imbalanced fraud settings, and that naive protocols may overestimate deployable performance [3, 10]. Recent comparative studies confirm that, under realistic temporal splits and deployment-oriented evaluation, tree-based ensembles frequently remain among the strongest baselines for tabular fraud detection problems [16]. These findings align with our final benchmark (Section 4): under the same chronological split and the same validation-selected cost-aware operating rule, a random forest achieves the lowest test cost, while the proposed DNN reaches very similar operational performance.

2.2. Deep learning for transactional fraud and real-time e-payment

Deep learning has been explored for fraud detection as a way to model complex feature interactions and learn nonlinear decision surfaces under severe imbalance. Many studies on `creditcard.csv` and related benchmarks train fully connected neural networks and report strong AUROC or high recall for fraud when imbalance handling is applied [15]. Comprehensive reviews of financial fraud detection confirm that deep learning models are increasingly adopted across banking, credit card, and online payment systems, while also noting that reported gains depend strongly on data imbalance treatment, validation strategy, and threshold selection [16]. However, a common limitation is that results are often reported under random splits and cost-agnostic thresholds (frequently $\tau = 0.5$), which can inflate apparent gains and obscure the operational false-alarm burden in deployment.

Beyond offline tabular benchmarks, real-time e-payment fraud detection motivates deep models designed for fast inference and robust handling of imbalance. El Youbi et al. propose a deep-learning approach for real-time fraud detection in e-payment systems and discuss practical constraints such as imbalance and latency, reinforcing the need for deployment-oriented evaluation and operating-point selection [1].

Sequential deep learning further models transaction histories per cardholder. Jurgovsky et al. frame fraud detection as sequence learning and show that LSTM-based architectures can improve detection by capturing temporal behavioural patterns [17]. Such approaches can be effective when longer behavioural histories are available, but they increase modelling complexity and data requirements compared with the transaction-level setting of `creditcard.csv`. Finally, recent research in tabular ML argues that for many tabular datasets, modern gradient-boosted trees remain strong baselines relative to deep networks, and careful experimental design is necessary before claiming superiority of deep learning on tabular data [6, 7]. This context is consistent with our results: the proposed feed-forward DNN is competitive once the decision policy is properly specified, while tree ensembles remain slightly stronger.

Recent advances in hybrid and ensemble architectures (2022–2025). The landscape of fraud detection has evolved rapidly with the introduction of hybrid and ensemble architectures. Recent studies demonstrate that combining diverse classifiers can mitigate the weaknesses of individual models. For instance, Khalid et al. [27] and Alfaiz and Fati [28] showed that integrating ensemble methods (such as

Stacking or CatBoost) with advanced resampling techniques (SMOTE, AllKNN) significantly improves robustness against class imbalance. In the deep learning domain, hybrid architectures have gained traction; Fahim et al. [29] and El-Kenawy et al. [30] proposed combining Convolutional Neural Networks (CNNs) with Recurrent Neural Networks (RNNs/LSTMs) to simultaneously capture spatial feature interactions and temporal dependencies. Similar deep hybrid methodologies have also proven effective in broader cybersecurity domains, such as encrypted traffic classification [31] and DDoS detection [32], highlighting the generalizability of feature extraction for anomaly detection. Feature selection also remains critical, with evolutionary approaches like Genetic Algorithms showing promise in optimising input spaces for fraud classifiers [33].

2.3. Unsupervised and hybrid anomaly-detection frameworks

Because fraud labels can be scarce and fraud strategies evolve, unsupervised and semi-supervised anomaly detection has long been used to model normal behaviour and flag deviations. Classical approaches include One-Class SVM, Local Outlier Factor, Isolation Forest, and clustering-based scoring; deep variants include autoencoders and GAN-based detectors [20, 10]. These methods can be less sensitive to class imbalance because they focus on modelling the majority (legitimate) distribution, but they often require careful tuning and may be unstable when the definition of "normal" shifts over time.

A practical direction is to combine unsupervised anomaly scores with supervised learning. Hybrid pipelines can use anomaly signals as additional features or as prioritisation cues, potentially improving robustness under drift [10]. Recent systematic analyses also emphasise that unsupervised fraud detection performance is highly dependent on pipeline choices and evaluation design [21]. Recent surveys further highlight that hybrid fraud detection strategies are most effective when evaluated under deployment-oriented protocols that explicitly account for temporal effects and operational decision rules [16]. While the present paper focuses on supervised transaction-level classification, these hybrid insights motivate future extensions that integrate anomaly-derived meta-features to stabilise detection under non-stationarity.

2.4. Cost-sensitive learning, thresholds, and evaluation under imbalance

In fraud detection, misclassification costs are asymmetric: missing fraud (FN) is typically much more expensive than flagging a legitimate transaction (FP). Cost-sensitive learning therefore plays a central role, either by incorporating costs during training or by selecting thresholds that minimise expected loss. Elkan provides a foundational analysis of cost-sensitive decision making and the relationship between posterior probabilities and optimal thresholds [23]. In the fraud domain, Bahnsen et al. introduce minimum-risk and example-dependent cost frameworks that directly optimise financial savings and demonstrate that cost-aware decision policies can outperform cost-agnostic rules [12, 13]. Related work on cost curves and decision analysis further supports evaluating classifiers under explicit cost regimes rather than a single arbitrary threshold [19, 18].

Because fraud data are extremely imbalanced, metric choice also matters. Precision–recall curves are often more informative than ROC curves when positives are rare, since they focus on precision at relevant recall levels and better reflect the false-alarm burden [25, 4]. For imbalance handling, sampling methods such as SMOTE are widely used, but their interaction with temporal protocols must be carefully managed to avoid leakage [5, 8]. More broadly, concept drift in streaming environments can degrade models trained on historical data, motivating evaluation protocols that respect temporal ordering and re-tuning strategies consistent with deployment [9, 10]. Recent review studies stress that cost-sensitive threshold calibration should be treated as an integral component of fraud detection systems rather than a post hoc adjustment, particularly under highly imbalanced and evolving data streams [16].

2.5. Positioning of the present work

Prior literature establishes three practical lessons that directly motivate our experimental design and explain the final outcomes in Section 4. First, tree-based ensembles are strong tabular baselines and often remain state-of-practice in fraud detection [11, 14]. Second, deep learning can be competitive in transaction-level settings and is particularly relevant in real-time e-payment contexts, but its reported gains depend strongly on evaluation protocol and decision policy [15, 1]. Third, cost-sensitive thresholding is essential in deployment: the operating point must be selected to reflect business loss asymmetry rather than fixed heuristics [23, 12, 13].

Accordingly, our contribution is not a novel architecture, but a deployment-oriented evaluation of a compact feed-forward DNN under a strictly chronological split and a validation-selected cost-aware threshold, together with a controlled comparison against logistic regression, random forest, and XGBoost under the *same* preprocessing and decision policy. This positioning matches our final findings: under the chosen cost regime, random forest yields the lowest expected cost, while the proposed DNN achieves a very similar operational cost once the threshold is selected using the same cost model on the validation window.

3. Methodology

In this section, we describe the proposed deep learning-based framework for credit card fraud detection on the `creditcard.csv` dataset. The methodology is designed to address two main challenges: the extreme class imbalance between legitimate and fraudulent transactions, and the need to align the decision rule with asymmetric operational costs. The overall pipeline is summarised in Figure 1.

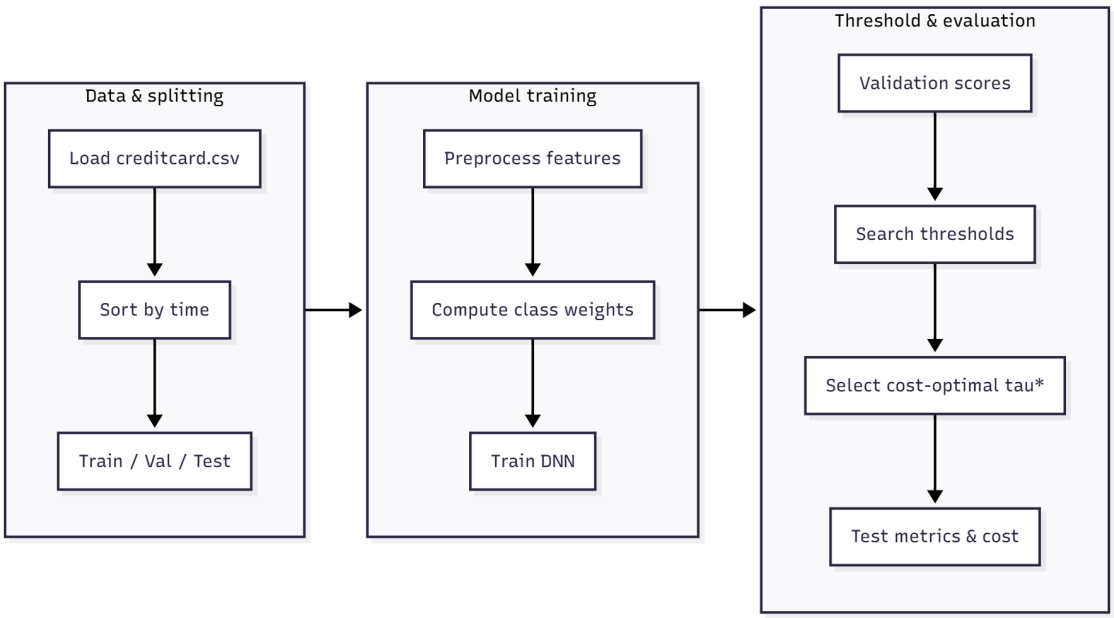

Figure 1. Overview of the proposed deep learning-based credit card fraud detection pipeline.

3.1. Data set

The experiments are conducted on the publicly available `creditcard.csv` dataset introduced by Dal Pozzolo et al. [26] , which contains real transactions carried out by European cardholders over a period

of two days. Each transaction is described by 30 numerical features and a binary label `Class` indicating whether the transaction is fraudulent. The 30 features consist of 28 anonymised principal component analysis (PCA) components $V_1, \ldots, V_{28}$, together with the original `Time` and `Amount` attributes. The label `Class` equals 1 for fraud and 0 for legitimate transactions.

Table 1 summarises the main characteristics of the dataset. In total, there are 284,807 transactions, among which 492 are labelled as fraud, corresponding to a fraud prevalence of approximately 0.1727%.

Table 1. Summary of the `creditcard.csv` dataset.

Statistic	Value
Total number of transactions	284,807
Number of legitimate transactions (`Class` = 0)	284,315
Number of fraudulent transactions (`Class` = 1)	492
Fraud prevalence (%)	0.1727
Number of features (including Time and Amount)	30
Total columns (features + label)	31

To better approximate a realistic deployment scenario, transactions are first sorted chronologically by `Time`. We then adopt a temporal split: the earliest 60% of transactions are used for training, the next 20% for validation, and the most recent 20% for testing. The resulting subset sizes and fraud rates are reported in Table 2. This chronological protocol avoids information leakage from future to past and reflects the situation in which models are trained on past data and applied to subsequent transactions.

Table 2. Chronological split of the `creditcard.csv` dataset and class distribution in each subset.

Split	Total	Legitimate	Fraud	Fraud rate (%)
Train	170,884	170,524	360	0.2107
Validation	56,961	56,904	57	0.1001
Test	56,962	56,887	75	0.1317

3.2. Data preprocessing and feature engineering

The preprocessing pipeline is designed to preserve the information contained in the PCA components while putting the temporal and monetary variables on a comparable scale.

Scaling of temporal and monetary features. The raw attributes `Time` and `Amount` exhibit different ranges and distributions compared to the PCA features. We therefore standardise these two variables using statistics (mean and standard deviation) computed on the training subset only. The same transformation is then applied to the validation and test subsets. The PCA components $V_1, \ldots, V_{28}$ are left unchanged, as they are already centred and scaled by construction.

Feature representation. All features are numerical; there are no explicit categorical variables, such as merchant category or country, and no cardholder identifiers. Each transaction is thus represented as a 30-dimensional vector in $\mathbb{R}^{30}$, obtained by concatenating the standardised `Time` and `Amount` with the 28 PCA components. Preliminary experiments with additional non-linear transformations (e.g., $\log(1 + \texttt{Amount})$) did not yield consistent improvements and were not retained in the final pipeline.

Leakage-free transformation. To avoid information leakage, all preprocessing steps (computation of scaling parameters, feature concatenation) are fitted exclusively on the training subset. The resulting transformers are then applied to the validation and test subsets without refitting, ensuring that no information from future transactions influences the training process.

3.3. Class imbalance handling

The dataset exhibits severe class imbalance, with fraudulent transactions representing less than 0.3% of observations in each split (Table 2). To prevent the model from being dominated by the majority class, we employ a class-weighted loss.

Class-weighted loss. Let N_0 and N_1 denote the numbers of legitimate and fraudulent transactions in the training subset, respectively. We define class weights

$$w_0 \propto \frac{1}{N_0}, \qquad w_1 \propto \frac{1}{N_1},$$

normalised such that their average equals 1. For the present dataset, this yields $w_0 \approx 0.5011$ and $w_1 \approx 237.3389$, as reported in Table 3. These weights are incorporated into the binary cross-entropy loss so that fraudulent transactions contribute more strongly to the optimisation.

Table 3. Class weights used in the weighted binary cross-entropy loss, computed from the training subset.

Class	Weight
Legitimate (`Class` $= 0$)	0.5011
Fraud (`Class` $= 1$)	237.3389

Leakage-free protocol. Class weights are computed once from the original training labels and remain fixed during optimisation. The validation and test subsets are not resampled and do not influence the computation of weights, ensuring that performance estimates on these subsets reflect the true underlying class distribution.

Alternative strategies. We also considered oversampling the minority class using Random Over-Sampling and SMOTE on the training subset. However, for the proposed deep model and cost-aware evaluation, class-weighted learning without synthetic samples provided a satisfactory balance between recall, precision and probability calibration. A more extensive comparison of resampling strategies is left for future work.

3.4. Model architecture design

We adopt a fully connected deep neural network (DNN) tailored to tabular credit card transaction data. The network takes as input the 30-dimensional feature vector described above and produces a single probability score representing the estimated likelihood that the transaction is fraudulent.

The architecture, summarised in Figure 2 and Table 4, comprises:

- an input layer of dimension 30;
- three hidden dense layers with 128, 64 and 32 neurons, respectively, each followed by a Rectified Linear Unit (ReLU) activation function;
- dropout layers with rate 0.3 after the first and second hidden layers, to reduce overfitting by randomly deactivating a subset of units during training;
- a final dense output layer with a single neuron and sigmoid activation, producing a probability $f(x_i;\theta) \in [0,1]$ that transaction x_i is fraudulent.

3.5. Training

The model is trained on the chronological training subset and monitored on the validation subset. We use the Adam optimizer with an initial learning rate of 10^{-3} and mini-batches of size 2048. The loss function is the class-weighted binary cross-entropy described in Section 3.8.

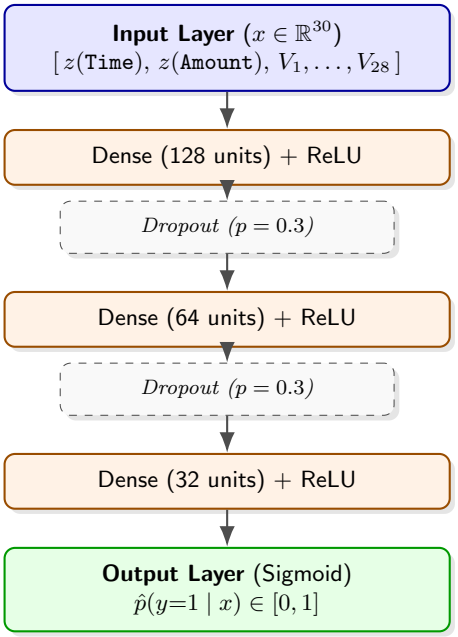

Figure 2. Architecture of the proposed deep neural network (DNN) for credit card fraud detection. Time and amount are first standardised and concatenated with the 28 PCA components $V_1, \ldots, V_{28}$ to form a 30-dimensional input vector. This vector is processed by three fully connected hidden layers with 128, 64 and 32 neurons, respectively, each followed by a ReLU activation. Dropout layers with rate 0.3 after the first and second hidden layers reduce overfitting. The output layer consists of a single sigmoid neuron that returns the predicted probability that a transaction is fraudulent.

Table 4. Architecture of the proposed deep neural network (DNN) for credit card fraud detection.

Layer (type)	Output shape	Parameters
Input (30 features)	(None, 30)	0
Dense (128 units, ReLU)	(None, 128)	3,968
Dropout (rate = 0.3)	(None, 128)	0
Dense (64 units, ReLU)	(None, 64)	8,256
Dropout (rate = 0.3)	(None, 64)	0
Dense (32 units, ReLU)	(None, 32)	2,080
Output (1 unit, Sigmoid)	(None, 1)	33
Total trainable parameters		14,337

Training is run for a maximum of 50 epochs with early stopping based on the validation area under the precision–recall curve (AUC-PR). Specifically, if the validation AUC-PR does not improve for 5 consecutive epochs, training is stopped and the model parameters are restored to those obtained at the best validation AUC-PR. In the present experiments, early stopping selects epoch 1 as the best epoch. Table 5 reports the training and validation metrics over the first six epochs.

Why early stopping selects epoch 1. Table 5 shows that the validation AUC-PR peaks at epoch 1 (0.7741) even though training AUC-PR and training loss continue to improve in later epochs. This behavior is typical under extreme imbalance and strong class weighting: because the minority class receives a very large weight ($w_1 \approx 237$), early gradient updates are dominated by a small number of

Table 5. Training history of the DNN model on the chronological split (first six epochs). The best validation AUC-PR is obtained at epoch 1, which is used by early stopping.

Epoch	Train AUC_PR	Train AUC_ROC	Train loss	Val. AUC_PR	Val. AUC_ROC	Val. loss
1	0.5233	0.9152	0.3427	**0.7741**	0.9636	0.0863
2	0.6930	0.9743	0.1749	0.7598	0.9776	0.0881
3	0.6477	0.9830	0.1475	0.5886	0.9785	0.0880
4	0.6557	0.9832	0.1423	0.5940	0.9764	0.0805
5	0.6658	0.9869	0.1279	0.5910	0.9755	0.0838
6	0.6680	0.9890	0.1207	0.6598	0.9729	0.0691

fraud examples, and the model rapidly learns a separating signal that improves minority ranking on the validation window. With continued optimization, the network can begin to over-specialize to rare patterns present in the training window (or to noise in minority observations), improving training metrics while degrading the generalization of probability scores on temporally later validation data. Since deployment performance is governed by future-window behavior rather than training loss, selecting the epoch with best validation AUC-PR is a principled choice for operating in this setting.

3.6. Mathematical formulation and training principles

We formalise the fraud detection task as a binary classification problem. Each transaction is represented by a feature vector $x_i \in \mathbb{R}^{30}$ and a label $y_i \in \{0, 1\}$, where $y_i = 1$ denotes a fraudulent transaction and $y_i = 0$ otherwise. The DNN defines a parametric function

$$f(x_i; \theta) \in [0, 1],$$

where θ denotes the trainable parameters of the network and $f(x_i; \theta)$ is interpreted as the predicted probability that transaction x_i is fraudulent.

Given a training set of N transactions, the learning objective is to minimise the class-weighted binary cross-entropy loss

$$L(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \big[w_1 \, y_i \log f(x_i; \theta) + w_0 \, (1 - y_i) \log(1 - f(x_i; \theta)) \big],$$

where w_0 and w_1 are the class weights associated with legitimate and fraudulent transactions, respectively (Table 3). This formulation increases the contribution of minority fraudulent cases in the gradient updates and helps to alleviate the impact of class imbalance.

After training, the probabilistic outputs $f(x_i; \theta)$ are converted into hard predictions via a decision threshold $\tau \in [0, 1]$: a transaction is classified as fraud if $f(x_i; \theta) \geq \tau$ and as legitimate otherwise.

Cost-sensitive decision rule versus cost-sensitive training. In this study, cost sensitivity is applied at the *decision stage* rather than directly during network optimization. The DNN is trained to produce calibrated posterior scores $\hat{p}(y{=}1 \mid x)$ via (weighted) cross-entropy, while the final operating rule is obtained by selecting a threshold that minimizes an empirical approximation of expected loss on a validation window. This design follows Bayes minimum-risk decision theory: once posterior probabilities are available, the risk-minimizing classifier under asymmetric misclassification costs is obtained by an appropriate threshold on $\hat{p}(y{=}1 \mid x)$, rather than by changing the model family itself. As emphasized in cost-sensitive learning and fraud decision analysis, this separation between *probability estimation* and *cost-aware decision calibration* provides a transparent way to align deployment behavior with operational objectives and to adapt the operating point when the cost ratio changes [23, 12, 13].

3.7. Hyperparameter selection

The architecture (hidden-layer widths and dropout rate) and optimisation hyperparameters (learning rate, batch size, early-stopping patience) were chosen based on preliminary experiments on the training and validation subsets. A small random search was performed over:

- learning rate $\eta \in \{10^{-2}, 10^{-3}, 10^{-4}\}$;
- hidden-layer widths in $\{64, 128\}$ for the first layer and $\{32, 64\}$ for the second;
- dropout rate $p \in \{0.2, 0.3, 0.4\}$;
- batch size $B \in \{1024, 2048\}$.

The final configuration (128–64–32 hidden units, dropout rate 0.3, learning rate 10^{-3}, batch size 2048) provided a good trade-off between validation AUC-PR, stability of training and computational cost. More systematic hyperparameter optimisation strategies could further improve performance and are left for future work.

3.8. Optimizer and loss function justification

The Adam optimizer is employed due to its adaptive learning-rate mechanism and robustness in the presence of sparse or noisy gradients, which are common in highly imbalanced classification problems. By maintaining per-parameter estimates of first and second moments of the gradients, Adam offers fast convergence without extensive manual tuning of the learning rate.

Binary cross-entropy is adopted as the base loss function because it directly models the Bernoulli likelihood of fraud versus non-fraud labels and is the standard choice for probabilistic binary classification. Incorporating class weights into the cross-entropy formulation allows the model to focus more strongly on minority fraudulent cases, while the subsequent cost-aware thresholding step (Section 3.9) aligns the final decision rule with operational objectives.

3.9. Evaluation

Once the DNN has been trained and early stopping has selected the best epoch, the model is evaluated in two stages. First, we compute predicted fraud probabilities on the validation subset and use them to select a cost-aware decision threshold. Second, we apply this validation-selected threshold *unchanged* to the held-out test subset to obtain the final performance metrics.

Cost-aware threshold selection. Let C_{FN} and C_{FP} denote the costs associated with a false negative (missed fraud) and a false positive (legitimate transaction incorrectly flagged), respectively. For a given threshold τ, let $\mathrm{FN}(\tau)$ and $\mathrm{FP}(\tau)$ denote the numbers of false negatives and false positives on the validation subset when predicting fraud if $\hat{p}(x) \geq \tau$. We define the empirical cost

$$\widehat{R}(\tau) = C_{\mathrm{FN}} \cdot \mathrm{FN}(\tau) + C_{\mathrm{FP}} \cdot \mathrm{FP}(\tau).$$

In this study, we adopt an asymmetric cost model with $C_{\mathrm{FN}} = 10$ and $C_{\mathrm{FP}} = 1$, reflecting that missed frauds are substantially more expensive than false alarms. We evaluate $\widehat{R}(\tau)$ on a fine grid of thresholds $\tau \in [0, 1]$ and select the cost-optimal operating point

$$\hat{\tau}^* \in \arg\min_{\tau \in [0,1]} \widehat{R}(\tau).$$

Under the above cost ratio ($r = C_{\mathrm{FN}}/C_{\mathrm{FP}} = 10$), this procedure yields $\hat{\tau}^* = 0.924$ on the validation window, with $\mathrm{FP} = 17$, $\mathrm{FN} = 13$, and a minimum validation cost of $10 \times 13 + 1 \times 17 = 147$. Figure 3 illustrates the validation cost curve and the selected operating point.

Justification of the cost ratio. The choice $C_{\mathrm{FN}} = 10$ and $C_{\mathrm{FP}} = 1$ is intended to represent a realistic asymmetry in fraud operations, where a missed fraud can trigger chargeback procedures, reimbursement, customer support and potential downstream risk, whereas a false alarm typically incurs a smaller but non-negligible cost due to manual review and customer friction. We stress that the true ratio is application-dependent (issuer policy, review capacity, customer segment, and transaction context). For this reason, we also perform a cost-ratio sweep in Section 4 to show how the validation-selected operating point and the FP/FN trade-off change when $r = C_{\mathrm{FN}}/C_{\mathrm{FP}}$ varies, which is consistent with prior minimum-risk fraud decision frameworks [12, 13].

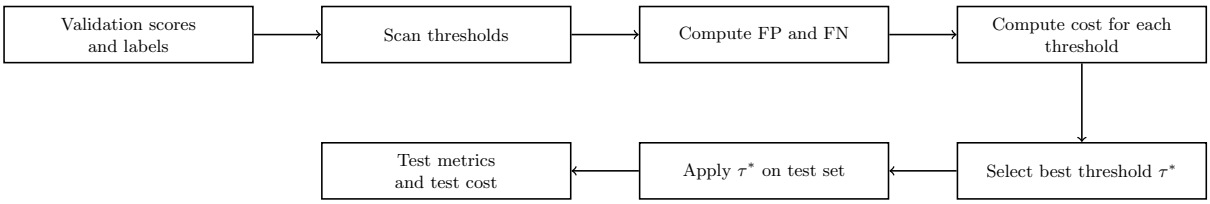

Figure 3. Cost-aware decision threshold selection. The trained DNN is applied to the validation set to obtain predicted probabilities and true labels. A grid of candidate thresholds is scanned; for each τ, the numbers of false positives (FP) and false negatives (FN) are computed and combined into the cost $\widehat{R}(\tau) = C_{\mathrm{FN}}\mathrm{FN}(\tau) + C_{\mathrm{FP}}\mathrm{FP}(\tau)$. The threshold $\hat{\tau}^*$ minimising validation cost is selected and then applied unchanged to the test set.

Table 6. Summary of cost-aware operating-point selection with $C_{\mathrm{FN}} = 10$ and $C_{\mathrm{FP}} = 1$. The threshold $\hat{\tau}^*$ is selected on the validation set and then applied unchanged to the test set.

Split	FP	FN	Threshold $\hat{\tau}^*$	Cost
Validation (selection)	17	13	0.924	$10 \times 13 + 1 \times 17 = 147$
Test (evaluation)	20	17	0.924	$10 \times 17 + 1 \times 20 = 190$

Test-set evaluation. After fixing $\hat{\tau}^*$ on the validation subset, we apply this threshold to the held-out test subset and compute threshold-free discrimination metrics (ROC-AUC and PR-AUC) as well as threshold-dependent metrics (precision, recall, F1-score), the confusion matrix, and the expected cost under the same cost model. These quantitative results, together with decision-policy comparisons (naive vs. F1-opt vs. cost-opt), cost-ratio sensitivity analysis, and baseline benchmarking, are reported and discussed in Section 4.

3.10. Reproducibility and implementation details

To ensure reproducibility, all experiments were conducted using fixed random seeds for NumPy, TensorFlow, and scikit-learn (seed = 42). The models were implemented in Python 3.10 using TensorFlow 2.15 and scikit-learn 1.4. Experiments were executed on a workstation equipped with an Intel Core i7 CPU and 32 GB RAM; no GPU acceleration was required. Training the proposed DNN under the chronological protocol required approximately 2–3 minutes, including validation-based threshold selection. All preprocessing steps, class weights, and decision thresholds are fully specified in the manuscript, enabling exact replication of the reported results.

4. Experimental results and discussion

This section reports the empirical performance of the proposed DNN-based framework on the `creditcard.csv` dataset under the strictly chronological protocol described in Section 3. We first

describe the training behaviour and the validation-driven operating-point selection, then we present test-set discrimination results, a decision-policy comparison (naive vs. F1-opt vs. cost-opt), and finally a benchmark against standard supervised baselines under the same temporal split and the same cost-aware threshold selection principle.

4.1. Training behaviour

Figure 4 (left) shows the evolution of the area under the precision–recall curve (AUC-PR) on the training and validation subsets over the first epochs. The training AUC-PR increases from 0.5233 to 0.6680 over the first six epochs, while the validation AUC-PR reaches its maximum at epoch 1 (0.7741) and then fluctuates between approximately 0.59 and 0.66. This pattern indicates that the network captures most of the separative signal early, and that extended training mainly increases the risk of overfitting on the rare fraud instances. Early stopping based on validation AUC-PR therefore selects epoch 1 as the operating point (Table 5).

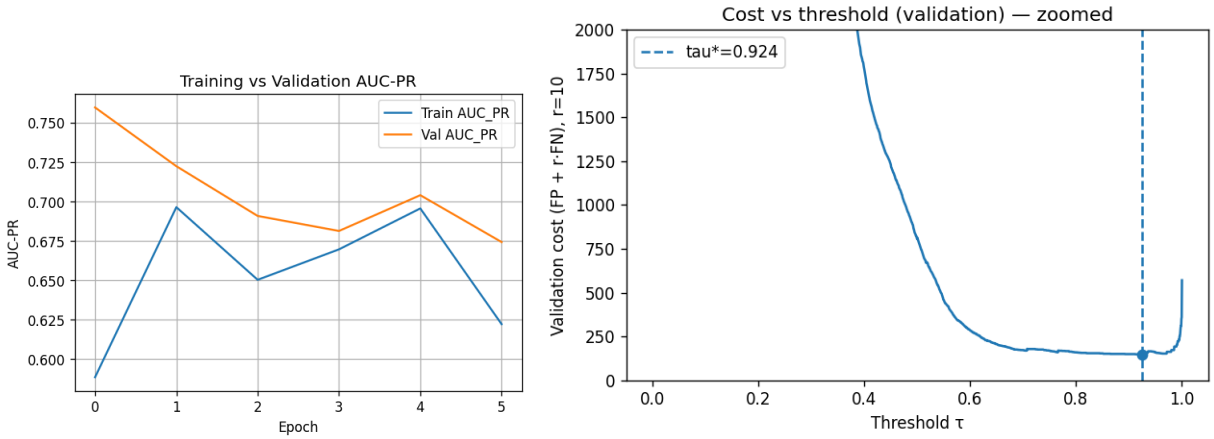

Figure 4. (Left) Training versus validation AUC-PR over the first epochs. The validation AUC-PR peaks at epoch 1, which motivates early stopping in this highly imbalanced setting. (Right) Validation cost as a function of the decision threshold τ under the cost model $C_{\mathrm{FN}} = 10$ and $C_{\mathrm{FP}} = 1$; the minimum is attained at $\tau^* = 0.924$, which is selected as the operating threshold.

4.2. Cost-aware threshold selection

Given validation scores produced by the early-stopped model, we select the decision threshold by minimising the empirical cost

$$\widehat{R}(\tau) = C_{\mathrm{FN}} \cdot \mathrm{FN}(\tau) + C_{\mathrm{FP}} \cdot \mathrm{FP}(\tau),$$

where $\mathrm{FN}(\tau)$ and $\mathrm{FP}(\tau)$ are computed on the validation subset. Throughout the main analysis, we use $C_{\mathrm{FN}} = 10$ and $C_{\mathrm{FP}} = 1$ (ratio $r = 10$), reflecting the higher operational impact of missed frauds relative to false alarms. The resulting validation curve (Figure 4, right) yields the cost-optimal operating point $\tau^* = 0.924$, which is then applied *unchanged* to the held-out test subset.

4.3. Test-set discrimination performance

Figure 5 reports the ROC and precision–recall curves on the test set. The proposed DNN attains a ROC-AUC of 0.9489 and a PR-AUC (average precision) of 0.7813, indicating strong ranking ability under extreme imbalance.

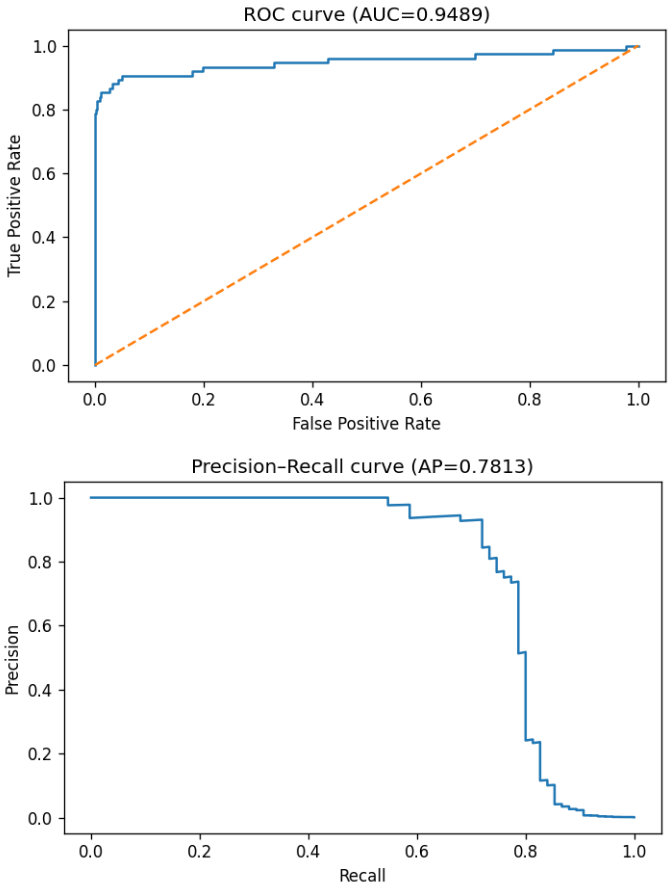

Figure 5. ROC and precision–recall curves of the proposed DNN on the test set. The ROC curve attains an AUC of 0.9489, while the precision–recall curve achieves a PR-AUC (average precision) of 0.7813, reflecting strong discrimination performance under extreme class imbalance.

At the cost-optimal threshold $\tau^* = 0.924$ (selected on validation for $r = 10$), the test confusion matrix is

$$\begin{bmatrix} 56{,}867 & 20 \\ 17 & 58 \end{bmatrix},$$

corresponding to 20 false positives and 17 false negatives (37 total errors), and an expected test cost of $10 \times 17 + 1 \times 20 = 190$. At this operating point, the fraud-class precision is 0.7436, recall is 0.7733, and F1-score is 0.7582.

Table 7. Confusion matrix of the proposed DNN on the test set at the cost-optimal threshold $\tau^* = 0.924$ (selected on validation with $C_{\mathrm{FN}} = 10$, $C_{\mathrm{FP}} = 1$).

	Predicted Not Fraud	Predicted Fraud
Actual Not Fraud	56,867	20
Actual Fraud	17	58

Operational interpretation of the confusion matrix. At the validation-selected cost-optimal threshold $\tau^* = 0.924$, the test set contains 20 false positives and 17 false negatives (Table 7). Operationally,

the 20 false positives correspond to legitimate transactions that would be flagged for additional verification or manual review, representing the controllable false-alarm workload. The 17 false negatives correspond to missed fraud cases, which are penalized more strongly under the adopted cost regime. The resulting expected cost is $10 \times 17 + 1 \times 20 = 190$, illustrating the intended trade-off: the operating point accepts a small review burden to reduce missed fraud exposure. This interpretation clarifies how threshold selection translates ranking performance (PR-AUC/ROC-AUC) into concrete deployment outcomes.

Table 8. Classification report of the proposed DNN on the test set at $\tau^* = 0.924$.

Class	Precision	Recall	F1-score	Support
Not Fraud (`Class` = 0)	0.9997	0.9996	0.9997	56,887
Fraud (`Class` = 1)	0.7436	0.7733	0.7582	75
Overall accuracy		0.99935		56,962

4.4. Decision-policy comparison

Because operational performance depends on the decision policy, we compare three common thresholding strategies for the DNN: (i) a naive fixed threshold $\tau = 0.5$, (ii) an F1-optimal threshold τ_{F1} selected on validation, and (iii) the cost-optimal threshold τ^* selected on validation under the asymmetric cost model ($r = 10$). The results are summarised in Table 9.

Table 9. Decision-policy comparison for the proposed DNN on the test set. PR-AUC and ROC-AUC are threshold-free; other metrics are computed at the specified threshold. Costs use $C_{\mathrm{FN}} = 10$ and $C_{\mathrm{FP}} = 1$.

Policy	τ	PR-AUC	ROC-AUC	$\mathrm{Prec}_{\mathrm{fraud}}$	$\mathrm{Rec}_{\mathrm{fraud}}$	$\mathrm{F1}_{\mathrm{fraud}}$	FP	FN	Cost
Naive ($\tau = 0.5$)	0.5000	0.7813	0.9489	0.1073	0.8400	0.1903	524	12	644
F1-opt (τ_{F1})	0.9625	0.7813	0.9489	0.8571	0.7200	0.7826	9	21	219
Cost-opt (τ^*, $r = 10$)	0.9240	0.7813	0.9489	0.7436	0.7733	0.7582	20	17	190

A naive threshold ($\tau = 0.5$) yields high recall (0.84) but triggers a very large number of false alarms (FP=524), resulting in a high expected cost (644). In contrast, validation-based operating-point selection substantially improves operational behaviour. Notably, the F1-optimal threshold ($\tau_{F1} = 0.9625$) is not identical to the cost-optimal threshold ($\tau^* = 0.924$): τ^* accepts a modest increase in false positives in exchange for fewer missed frauds, which reduces the expected cost under the asymmetric loss model.

4.5. Sensitivity to the cost ratio

To assess regime dependence, we repeat threshold selection for multiple cost ratios $r = C_{\mathrm{FN}}/C_{\mathrm{FP}}$. Table 10 reports the selected threshold and the resulting error trade-off on the test set when the operating point is chosen on validation for each ratio. Figure 6 visualises the resulting test cost at the validation-selected operating point as a function of r (log-scaled).

Table 10. Cost-ratio sweep for the DNN. For each ratio r, $\tau^*(r)$ is selected on the validation set and then applied to the test set. Costs are computed as $\mathrm{Cost} = \mathrm{FP} + r \cdot \mathrm{FN}$ (with $C_{\mathrm{FP}} = 1$).

r	$\tau^*(r)$	PR-AUC	$\mathrm{Prec}_{\mathrm{fraud}}$	$\mathrm{Rec}_{\mathrm{fraud}}$	$\mathrm{F1}_{\mathrm{fraud}}$	FP	FN
5	0.9625	0.7813	0.8571	0.7200	0.7826	9	21
10	0.9240	0.7813	0.7436	0.7733	0.7582	20	17
20	0.9240	0.7813	0.7436	0.7733	0.7582	20	17
50	0.7035	0.7813	0.4959	0.8000	0.6122	61	15
100	0.7035	0.7813	0.4959	0.8000	0.6122	61	15

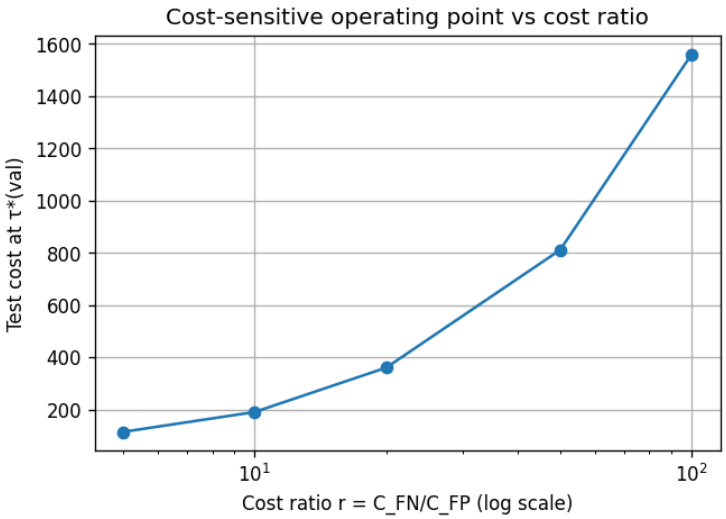

Figure 6. Cost-sensitive operating point versus cost ratio. For each ratio $r = C_{\mathrm{FN}}/C_{\mathrm{FP}}$, the threshold $\tau^*(r)$ is selected on the validation set and the resulting test cost is reported. The x-axis is log-scaled.

As r increases, the validation-selected operating point can shift to prioritise the reduction of false negatives, even if this requires accepting more false positives. In the present sweep, $\tau^*(r)$ decreases from 0.9625 at $r = 5$ to 0.924 at $r = 10$–20, and then to 0.7035 at $r = 50$–100, increasing recall while reducing (weighted) false-negative cost under higher ratios. This confirms that operating-point selection is regime-dependent: the preferred trade-off between missed frauds and false alarms changes with the cost ratio, so model deployment should explicitly specify the decision policy under which performance is evaluated.

4.6. Comparison with baseline methods

To place the DNN results in context, we compare against logistic regression (LR), random forest (RF), and XGBoost (XGB) under identical preprocessing, the same chronological split, and the same validation-based cost-aware threshold selection principle ($C_{\mathrm{FN}} = 10$, $C_{\mathrm{FP}} = 1$). To address the need for cost-sensitive baselines, we also include a Weighted Random Forest (trained with class weights) to compare training-stage cost adaptation versus the proposed decision-stage thresholding. Table 11 reports threshold-free PR-AUC and the fraud-class precision and recall at each model's cost-optimal operating point.

Table 11. Comparison with supervised baselines on the test set. Note that the **Weighted Random Forest** (requested baseline) yields a higher cost than the standard RF and the proposed DNN.

Model	PR-AUC	τ^*	$\mathrm{Prec}_{\mathrm{fraud}}$	$\mathrm{Rec}_{\mathrm{fraud}}$	$\mathrm{F1}_{\mathrm{fraud}}$	Test cost
Logistic Regression	0.7439	0.9990	0.7125	0.7600	0.7355	203
Random Forest (Std)	0.8140	0.1205	0.8636	0.7600	0.8085	189
Weighted RF	0.8113	0.50	0.80	0.77	0.78	252
XGBoost	0.7937	0.2690	0.7703	0.7600	0.7651	197
Proposed DNN	0.7813	0.9240	0.7436	0.7733	0.7582	190

Random forest attains the strongest operating performance in this setting, achieving the highest PR-AUC (0.8140) and the lowest expected test cost (189). XGBoost also performs competitively (PR-AUC 0.7937, cost 197). Logistic regression yields a lower PR-AUC (0.7439) and higher cost (203).

The proposed DNN achieves a PR-AUC of 0.7813 and an expected test cost of 190 at $\tau^* = 0.924$. While the random forest maintains substantially higher fraud precision, the DNN attains slightly higher fraud

recall. Notably, the **Weighted Random Forest** incurs a significantly higher cost (252) due to lower precision, confirming that post-hoc threshold calibration (as used in our DNN) is more operationally effective than simple class-weighted training.

Finally, a bootstrap analysis ($N = 1000$) confirms the statistical stability of the proposed DNN, yielding a 95% confidence interval for PR-AUC of $[0.7154, 0.8845]$.

4.7. Discussion

The experimental results provide several insights into the behaviour of the proposed DNN framework and its position relative to established baselines. First, under the strictly chronological protocol and the extreme class imbalance of the `creditcard.csv` dataset, the DNN achieves strong discrimination performance on the held-out test set, with ROC-AUC of 0.9489 and PR-AUC of 0.7813 (Figure 5). When the operating threshold is selected on the validation subset by minimising the asymmetric cost model ($C_{\mathrm{FN}} = 10$, $C_{\mathrm{FP}} = 1$), the resulting cost-optimal threshold is $\tau^* = 0.924$. At this operating point, the test confusion matrix is $\left[\begin{smallmatrix} 56{,}867 & 20 \\ 17 & 58 \end{smallmatrix}\right]$ (Table 7), which yields fraud-class precision of 0.7436, recall of 0.7733 and F1-score of 0.7582, with an expected test cost of 190. These results confirm that a simple feed-forward architecture, when paired with a deployment-oriented temporal split and validation-based operating-point selection, can deliver high-quality ranking and a practically meaningful false-positive/false-negative trade-off.

A second key insight is that *decision policy* matters as much as (and sometimes more than) the model family. Table 9 shows that using a naive threshold of $\tau = 0.5$ leads to a very large number of false alarms (FP=524), producing a high operational cost (644) despite high recall (0.84). In contrast, validation-selected thresholds drastically change the operating regime: the F1-optimal threshold $\tau_{F1} = 0.9625$ produces very high fraud precision (0.8571) but sacrifices recall (0.72), leading to a cost of 219; the cost-optimal threshold $\tau^* = 0.924$ accepts a moderate increase in false positives (FP=20 vs. 9) in exchange for fewer missed frauds (FN=17 vs. 21), which reduces expected cost under the asymmetric loss model. This illustrates a deployment-relevant point: even with the same trained scoring model, different thresholding policies correspond to different business decisions, and threshold selection should therefore be treated as part of the end-to-end system design rather than an afterthought.

Third, the comparative results in Table 11 confirm that tree-based ensembles remain very strong baselines for credit card fraud detection on this benchmark. Random forest attains the highest PR-AUC (0.8140) and the lowest test cost (189), while XGBoost is also competitive (PR-AUC 0.7937, cost 197). Logistic regression yields a lower PR-AUC (0.7439) and higher cost (203), and it requires an extreme operating threshold ($\tau^* \approx 0.999$) to control false alarms under severe imbalance. Against this backdrop, the proposed DNN achieves a performance profile that is close to the best ensemble model in operational terms (cost 190 vs. 189 for random forest), while providing competitive PR-AUC (0.7813). The error profiles differ: random forest maintains substantially higher fraud precision (0.8636), whereas the DNN achieves slightly higher fraud recall (0.7733 vs. 0.7600), which explains why the expected costs are nearly identical under $r = 10$. These findings suggest that, in this setting, the main practical benefit of the DNN is not to dominate tree-based ensembles, but to offer a competitive alternative whose performance becomes comparable once a realistic temporal protocol and a cost-sensitive operating rule are enforced. Although a class-weighted Random Forest was also evaluated, it did not outperform the standard Random Forest under validation-based cost optimization and therefore was not retained as a primary baseline in the discussion.

The cost-ratio sweep further highlights the *regime dependence* of operating-point selection. As the ratio $r = C_{\mathrm{FN}}/C_{\mathrm{FP}}$ increases, the optimal threshold tends to shift toward prioritising fraud capture, reducing false negatives at the expense of more false positives. This behaviour is expected under cost-sensitive decision theory and reinforces the practical lesson that model evaluation and deployment decisions should be made under an explicitly stated cost regime. In applications where the effective cost ratio can vary (e.g. different customer segments, different manual-review capacities, or different fraud-loss exposure), the

preferred operating point may change accordingly, and sensitivity analysis is therefore useful for robust deployment planning.

Several limitations of the present study should be acknowledged, despite the fully specified and reproducible experimental protocol described above. First, the experiments are conducted on a single public dataset with anonymised PCA features and a short temporal horizon; real deployments typically include richer contextual signals (merchant category, geographic location, device identifiers) and, critically, customer history. Second, the cost model used here is deliberately simple and assumes constant per-error costs; in practice, costs are often example-dependent (e.g. proportional to transaction amount, chargeback risk, or investigation workload), motivating future work on transaction-dependent cost functions. Finally, the DNN studied here is a static transaction-level model; sequential models (e.g. LSTMs) or attention-based architectures (transformers) that exploit longer behavioural histories may further improve performance and robustness under concept drift when such information is available.

5. Conclusion

In this paper, we proposed and evaluated a deep learning-based framework for credit card fraud detection on the public `creditcard.csv` dataset. The approach combines a simple fully connected neural network with a chronological train–validation–test protocol, class-weighted training to address extreme imbalance, and a validation-based cost-aware threshold selection procedure that explicitly minimises an operational loss function. This design ensures that the learned decision rule is tailored not only to statistical discrimination performance, but also to the asymmetric economic costs of false positives and false negatives. Rather than proposing a new optimization algorithm or network architecture, the contribution of this work lies in how standard models are evaluated and calibrated for deployment under asymmetric operational costs.

A key aspect of this work is the evaluation protocol itself. Whereas many prior deep-learning studies on `creditcard.csv` rely on random train–test splits and cost-agnostic threshold choices (often $\tau = 0.5$), we enforce a strictly chronological split that mirrors a realistic deployment scenario and choose the decision threshold on the validation set by minimising an explicit cost function. On the held-out test set, the proposed DNN achieves strong discrimination performance, with ROC-AUC of 0.9489 and PR-AUC of 0.7813. Importantly, our decision-policy analysis shows that threshold selection strongly affects operational outcomes: a naive threshold yields an excessive number of false alarms, whereas validation-based thresholding substantially reduces expected loss. Under the cost model $C_{\mathrm{FN}} = 10$ and $C_{\mathrm{FP}} = 1$, the cost-optimal operating point for the DNN is $\tau^* = 0.924$, leading to a fraud-class F1-score of 0.7582 and an expected test cost of 190.

Another important contribution is the systematic comparison with standard supervised baselines under exactly the same conditions. Using identical preprocessing, temporal splitting, and the same validation-based operating-point selection, we benchmark the DNN against logistic regression, random forest, and XGBoost. The experiments confirm that tree-based ensembles remain very strong baselines, with random forest attaining the lowest test cost (189). However, the proposed DNN reaches a very similar operational performance (cost 190) with competitive PR-AUC, and it outperforms logistic regression under naive thresholding. These results show that, once chronological evaluation and cost-sensitive thresholding are taken into account, a modest feed-forward network can be a competitive option for transaction-level credit card fraud detection.

Beyond the specific performance figures on `creditcard.csv`, an important message of this study is that the combination of a realistic temporal protocol, imbalance-aware learning, and cost-based decision rules is at least as important as the choice of classifier family. For practitioners, the proposed framework provides a simple recipe for aligning deep-learning models with operational constraints in highly imbalanced fraud detection settings.

Several directions for future research emerge from this work. One avenue is to extend the framework to richer feature sets, incorporating merchant information, geographic data, device identifiers, and customer history, and to assess the relative advantages of DNNs and tree-based ensembles in that context. Another is to move from a fixed cost model to example-dependent, transaction-level cost functions that better reflect real financial exposure and operational constraints. Finally, sequential and attention-based architectures (e.g. recurrent networks, transformers) could be investigated to exploit longer transaction histories and concept-drift adaptation, potentially further improving detection performance in dynamic, real-world environments.

Funding and Competing Interest Declaration

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The authors have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

REFERENCES

1. R. El Youbi, F. Messaoudi, M. Loukili, and R. Loukili, *From Click to Checkout: Deep Learning for Real-Time Fraud Detection in E-Payment Systems*, Statistics, Optimization & Information Computing, vol. 14, no. 6, pp. 3398–3408, 2025. doi:10.19139/soic-2310-5070-2891.
2. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature, *Decision Support Systems*, vol. 50, no. 3, pp. 559–569, 2011, doi:10.1016/j.dss.2010.08.006.
3. A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3784–3797, Aug. 2018, doi:10.1109/TNNLS.2017.2736643.
4. T. Saito and M. Rehmsmeier, "The Precision–Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets," *PLOS ONE*, vol. 10, no. 3, e0118432, 2015, doi:10.1371/journal.pone.0118432.
5. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002, doi:10.1613/jair.953.
6. Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, "Revisiting Deep Learning Models for Tabular Data," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, pp. 18932–18943, 2021.
7. L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?" in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 507–520, 2022.
8. H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009, doi:10.1109/TKDE.2008.239.
9. J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, vol. 46, no. 4, article 44, 2014, doi:10.1145/2523813.
10. F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, "Combining unsupervised and supervised learning in credit card fraud detection," *Information Sciences*, vol. 557, pp. 317–331, 2021, doi:10.1016/j.ins.2019.05.042.
11. K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi, "Credit card fraud detection using AdaBoost and majority voting," *IEEE Access*, vol. 6, pp. 14277–14284, 2018, doi:10.1109/ACCESS.2018.2806420.
12. A. Correa Bahnsen, A. Stojanovic, D. Aouada, and B. Ottersten, "Cost sensitive credit card fraud detection using Bayes minimum risk," in *Proceedings of the 12th International Conference on Machine Learning and Applications (ICMLA)*, vol. 1, pp. 333–338, 2013, doi:10.1109/ICMLA.2013.68.
13. A. Correa Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, "Example-dependent cost-sensitive decision trees," *Expert Systems with Applications*, vol. 42, no. 19, pp. 6609–6619, 2015, doi:10.1016/j.eswa.2015.04.042.
14. D. Almhaithawi, A. Jafar, and M. Aljnidi, "Example-dependent cost-sensitive credit cards fraud detection using SMOTE and Bayes minimum risk," *SN Applied Sciences*, vol. 2, article 1574, 2020, doi:10.1007/s42452-020-03375-w.
15. J. Parmar, A. C. Patel, and M. Savsani, "Credit card fraud detection framework: A machine learning perspective," *International Journal of Scientific Research in Science and Technology*, vol. 7, no. 6, pp. 431–435, 2020, doi:10.32628/IJSRST207671.
16. S. Panigrahi and V. L. N. Gorle, "Applications of Machine Learning and Deep Learning Algorithms in Financial Fraud Detection: A Review," *Computational Economics*, Springer, 2025, doi:10.1007/s10614-025-11089-7.
17. J. Jurgovsky, M. Granitzer, K. Ziegler, S. Calabretto, P.-E. Portier, L. He-Guelton, and O. Caelen, "Sequence classification for credit-card fraud detection," *Expert Systems with Applications*, vol. 100, pp. 234–245, 2018, doi:10.1016/j.eswa.2018.01.037.

18. T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006, doi:10.1016/j.patrec.2005.10.010.
19. C. Drummond and R. C. Holte, "Cost curves: An improved method for visualizing classifier performance," *Machine Learning*, vol. 65, no. 1, pp. 95–130, 2006, doi:10.1007/s10994-006-8199-5.
20. U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection," *Information Sciences*, vol. 479, pp. 448–455, 2019, doi:10.1016/j.ins.2017.12.030.
21. A. Elbadraoui, Y. Achraf, A. Elalaoui, S. Ouatik Elalaoui and M. Yassine, *Credit Card Fraud Detection with Unsupervised Learning: A Performance Analysis*, in *Proc. 2025 11th International Conference on Optimization and Applications (ICOA)*, Kenitra, Morocco, 2025, pp. 1–6, doi: 10.1109/ICOA66896.2025.11236917.
22. A. Elbadraoui, Y. Mouhssine, A. El Alaoui, and S. Ouatik El Alaoui, *A Bibliometric Exploration of Artificial Intelligence Methods for Transition from Conventional to Sustainable Financial Framework*, in M. Ezziyyani, J. Kacprzyk, and V. E. Balas (eds.), *International Conference on Advanced Intelligent Systems for Sustainable Development (AI2SD 2024)*, Lecture Notes in Networks and Systems, vol. 1403, Springer, Cham, 2025, doi: 10.1007/978-3-031-91337-2_54.
23. C. Elkan, "The foundations of cost-sensitive learning," in *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI'01)*, vol. 17, no. 1, pp. 973–978, 2001.
24. A. Fernández, S. García, F. Herrera, and N. V. Chawla, "SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary," *Journal of Artificial Intelligence Research*, vol. 61, pp. 863–905, 2018, doi:10.1613/jair.1.11192.
25. J. Davis and M. Goadrich, "The relationship between Precision–Recall and ROC curves," in *Proceedings of the 23rd International Conference on Machine Learning (ICML 2006)*, pp. 233–240, Pittsburgh, PA, USA, 2006, doi:10.1145/1143844.1143874.
26. A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2015)*, pp. 159–166, Cape Town, South Africa, 2015, doi:10.1109/CIDM.2015.26.
27. A. R. Khalid, N. Owoh, O. Uthmani, M. Ashawa, J. Osamor, and J. Adejoh, "Enhancing Credit Card Fraud Detection: An Ensemble Machine Learning Approach," *Big Data and Cognitive Computing*, vol. 8, no. 1, p. 6, 2024, doi:10.3390/bdcc8010006.
28. N. S. Alfaiz and S. M. Fati, "Enhanced Credit Card Fraud Detection Model Using Machine Learning," *Electronics*, vol. 11, no. 4, p. 662, 2022, doi:10.3390/electronics11040662.
29. A. Fahim, A. M. Osman, Z. Tarek, and A. M. Elshewey, "Credit Card Fraud Detection Based on a Hybrid CNN-RNN Deep Learning Model," *Engineering, Technology & Applied Science Research*, vol. 15, no. 6, pp. 28836–28842, 2025, doi:10.48084/etasr.13938.
30. E.-S. M. El-Kenawy, A. M. Zaki, W. H. Lim, A. Ibrahim, M. M. Eid, A. M. Osman, and A. M. Elshewey, "Credit Card Fraud Detection based on Deep Learning Models," *Mesopotamian Journal of Computer Science*, vol. 2024, pp. 204–213, 2024, doi:10.58496/MJCSC/2024/016.
31. A. M. Elshewey and A. M. Osman, "Enhancing encrypted HTTPS traffic classification based on stacked deep ensembles models," *Scientific Reports*, vol. 15, no. 1, p. 35230, 2025, doi:10.1038/s41598-025-21261-6.
32. A. M. Elshewey, S. Abbas, A. M. Osman, E. A. Aldakheel, and Y. Fouad, "DDoS classification of network traffic in software defined networking SDN using a hybrid convolutional and gated recurrent neural network," *Scientific Reports*, vol. 15, no. 1, p. 29122, 2025, doi:10.1038/s41598-025-13754-1.
33. E. Ileberi, Y. Sun, and Z. Wang, "A machine learning based credit card fraud detection using the GA algorithm for feature selection," *Journal of Big Data*, vol. 9, no. 1, p. 24, 2022, doi:10.1186/s40537-022-00573-8.