INFORMATICA, 2026, Vol. 0, No. 0, 1–26

DOI: https://doi.org/10.15388/26-INFOR646

MTLFraudDetect: Multi-Task Learning with Complementary Model Evidence for Credit Card Fraud Detection

Yu-Chi CHUNG[1], Michaela ESPINO[1], I-Fang SU[2,*]

[1] *Department of Industrial Engineering and Management, National Kaohsiung University of Science and Technology, No. 415, Jiangong Rd., Sanmin Dist., Kaohsiung City 807618, Taiwan*
[2] *Department of Computer Science and Information Engineering, National University of Tainan, 3, Sec. 2, Shu-Lin St., Tainan 700301, Taiwan*
e-mail: ycchung@nkust.edu.tw, mtespino@bpsu.edu.ph, ifangsu@mail.nutn.edu.tw

Received: August 2026; accepted: August 2026

Abstract. Credit card fraud (CCF) poses a growing threat to the global economy, with payment card fraud losses reaching approximately USD 33.8 billion in 2023. While deep learning approaches such as CNNs, autoencoders, and Transformer-based models have improved detection accuracy, their "black box" nature restricts adoption in high-stakes financial domains where transparent and auditable reasoning is essential for trust and regulatory compliance. To address this challenge, we propose MTLFraudDetect, a multi-task framework that combines fraud detection with complementary model-analysis techniques. The model combines fraud classification with two auxiliary tasks—data reconstruction and amount prediction—while employing curriculum learning and PCGrad to stabilize multi-task optimization. For model analysis, the framework combines reconstruction-based anomaly-sensitivity signals with SHAP-based decision attribution. Reconstruction errors indicate transaction-level and feature-wise deviations from patterns learned by the reconstruction branch, whereas SHAP examines how the classifier's shared latent representation and transaction amount contribute to its final prediction. Under an explicitly separated development-test protocol with training-fold-only preprocessing and out-of-fold hyperparameter selection, MTLFraudDetect achieves competitive F1-score and AUC-PR performance while providing complementary evidence at the levels of global anomaly detection, feature-wise error localization, and post-hoc decision attribution. These results do not by themselves establish formal explanatory faithfulness or complete interpretability.

Key words: deep learning, multi-task learning, credit card fraud detection, explainable artificial intelligence.

1. Introduction

Credit card fraud (CCF) poses a significant and rapidly growing threat to the global economy, resulting in substantial financial losses for both consumers and financial institutions (Alarfaj *et al.*, 2022; Cheng *et al.*, 2020). As cashless and card-based payment sys-

*Corresponding author.

tems continue to expand worldwide, despite regional differences in the dominant payment methods, opportunities for fraudsters to exploit system vulnerabilities also increase (Bank for International Settlements, 2026; Nilson Report, 2025; Caporal, 2025). Labelled credit card transaction data contain discriminative patterns that distinguish fraudulent transactions from legitimate ones. These patterns enable machine learning (ML) and deep learning (DL) models to learn informative representations for automated transaction classification (Bhattacharyya *et al.*, 2011). In this context, this study investigates whether a multi-task learning (MTL) framework can improve representation learning and enhance fraud detection performance. In high-stakes financial environments, however, accurate detection alone is insufficient. Financial institutions, regulators, and other stakeholders also require transparent decision-making processes to facilitate model interpretation, auditing, and regulatory compliance. Consequently, the substantial losses associated with CCF highlight the need for fraud detection systems that are both accurate and interpretable. Interpretability techniques therefore play an important role in understanding model behaviour and communicating the rationale underlying model predictions in real-world applications (Du *et al.*, 2020; Hassija *et al.*, 2024).

Early fraud detection systems relied primarily on rule-based approaches, in which domain experts manually designed heuristic rules to identify suspicious transactions (Leonard, 1995). Although these systems offer high interpretability, they require extensive domain expertise and considerable manual effort to remain effective. Furthermore, they incur substantial maintenance costs and require continuous updates to adapt to evolving fraud patterns, limiting their scalability and long-term adaptability (Krishnavardhan *et al.*, 2024).

To reduce dependence on manually crafted rules, researchers increasingly adopted ML techniques such as decision trees, SVMs, and KNN. These methods learn discriminative patterns from large volumes of historical transaction data and have demonstrated strong performance in a variety of applications. Conventional ML approaches remain competitive for tabular problems such as credit card transaction analysis. However, when they rely on fixed or manually engineered features, their ability to capture complex interactions in high-dimensional, nonlinear data depends heavily on the quality of feature engineering (Goodfellow *et al.*, 2016; Bishop and Bishop, 2023).

The emergence of DL has further advanced fraud detection by enabling neural network-based models to learn hierarchical feature representations directly from data. These models have achieved impressive performance across numerous application domains (Fanai and Abbasimehr, 2023), including CCF detection (CCFD). Nevertheless, their effectiveness remains limited in certain scenarios, indicating the need for continued improvements in model architectures and training strategies. In addition, most DL models operate as "black boxes", limiting their adoption in financial environments that require transparent, explainable, and auditable decision-making processes. Although post-hoc explainable artificial intelligence (XAI) techniques (Kilickaya, 2025; Sowmya *et al.*, 2025) have been developed to address this limitation, explanations generated only after prediction may provide limited insight into whether an input deviates from the patterns learned during training. Reconstruction-based anomaly evidence can therefore provide comple-

mentary information for model analysis without replacing post-hoc attribution methods that explain the classifier's final decision.

To address these limitations, this study investigates the potential of MTL (Crawshaw, 2020; Liu *et al.*, 2019) for CCFD. MTL enables a model to learn multiple related tasks simultaneously by combining shared feature representations with task-specific output layers. During training, the loss functions from multiple tasks jointly update the shared representation, allowing the model to learn features that generalize more effectively across related objectives (Zhang and Yang, 2021).

Beyond its primary objective of fraud classification, the proposed framework uses MTL to generate model-derived evidence that characterizes deviations from learned transaction patterns. Specifically, the reconstruction task produces transaction-level and feature-wise reconstruction errors that serve as anomaly-sensitivity signals. These signals quantify deviations from patterns learned predominantly from legitimate transactions. However, they are not intended to serve as formal explanations of the classifier's decisions or to satisfy formal XAI faithfulness criteria. Instead, they complement SHAP, which provides post-hoc feature attribution for the classifier's final prediction.

To address the identified research gaps, this study proposes **MTLFraudDetect**, a comprehensive framework that systematically investigates and optimizes MTL for credit card fraud detection through the following methodological contributions.

1. **Application of MTL to fraud detection.** We introduce an MTL framework specifically designed for CCFD. Unlike most previous studies, which employ single-task learning, the proposed framework incorporates heterogeneous auxiliary tasks to learn more robust feature representations. By leveraging complementary information from transaction reconstruction and transaction amount prediction, the framework is designed to improve minority-class detection while providing additional information for model analysis.
2. **Leakage-controlled evaluation protocol.** We evaluate the proposed framework using a stratified development-test holdout, training-fold-only preprocessing and resampling, pooled out-of-fold objectives for hyperparameter optimization, and an independent test set reserved exclusively for final evaluation. This protocol explicitly defines the sources of model-selection information and clearly separates model development from final performance assessment.
3. **Hybrid model-analysis pipeline.** We propose a hybrid analysis framework that combines reconstruction-based anomaly evidence with post-hoc decision attribution. The reconstruction task generates transaction-level anomaly-sensitivity scores and feature-wise reconstruction errors that identify deviations from patterns learned predominantly from legitimate transactions. SHAP independently provides post-hoc attribution of the classifier's output to the shared latent representation and transaction amount supplied to the classifier. These two components serve complementary purposes: reconstruction errors characterize deviations under the reconstruction objective, whereas SHAP characterizes classifier-input contributions within the assumptions of the SHAP framework. We do not claim that either component alone provides complete interpretability or satisfies formal explanation-faithfulness criteria.

4. **Systematic evaluation of MTL loss-balancing strategies.** Because MTL requires simultaneous optimization of multiple objectives, we conduct a comprehensive evaluation of several loss-balancing strategies, including uniform weighting, dynamic weight averaging (DWA) (Liu *et al.*, 2019) and PCGrad (Yu *et al.*, 2020). Our analysis examines how these strategies influence gradient interactions, representation learning, and overall fraud detection performance, providing practical guidance for designing stable and effective MTL-based fraud detection systems.
5. **Curriculum-based training strategy.** Finally, we propose a curriculum-based training strategy to improve task scheduling during optimization. Training begins with joint optimization of all tasks and gradually transitions toward greater emphasis on the primary fraud detection objective. We evaluate the effectiveness of this strategy under a fixed training budget and analyse the resulting performance trade-offs.

The remainder of this paper is organized as follows: Section 2 reviews related research on credit card fraud detection, with particular emphasis on ML, DL, and MTL approaches, and identifies existing research gaps and unresolved challenges. Section 3 presents the proposed MTLFraudDetect framework. Section 4 describes the experimental design. Section 5 presents the model-derived anomaly evidence, classifier attribution analyses, and comparative performance evaluation. Finally, Section 6 concludes the paper and discusses directions for future research.

2. Related Works

CCFD remains a major challenge due to the severe class imbalance, the continuously evolving nature of fraud strategies, and the scarcity of labelled fraudulent transactions. To address these issues, a wide range of ML and DL methods have been proposed, each offering distinct advantages and limitations.

2.1. *ML for CCFD*

Several studies have applied traditional ML algorithms to CCFD. Sudhakar and Kaliyamurthie (2023) compared Random Forest (RF), Naïve Bayes, Logistic Regression, and XGBoost, reporting that XGBoost achieved superior performance. Leevy *et al.* (2023) evaluated binary and one-class classification (OCC) approaches and found that binary classifiers, such as CatBoost, consistently outperformed OCC models. El Bazi *et al.* (2024) proposed OptiStack, a single-task stacking ensemble that combines XGBoost, CatBoost, and LightGBM with Logistic Regression as a meta-learner, together with SMOTE and Bayesian hyperparameter optimization. Using the European Credit Card Fraud Detection dataset, they reported fraud-class Precision of 0.88, Recall of 0.86, F1-score of 0.87, and ROC-AUC of 0.9887. Their results provide a useful published ensemble reference; however, the reported ROC-AUC is not directly comparable with AUC-PR, and the differences in evaluation protocols are discussed in Section 5.1. Despite these successes, traditional ML methods exhibit notable limitations. As highlighted in Alarfaj *et al.* (2022), they often struggle with real-world class imbalance, adapt poorly to evolving fraud patterns, and

produce high false-positive rates. These limitations have driven an increasing interest in more expressive DL approaches.

2.2. *DL for CCFD*

DL models have demonstrated strong performance in CCFD not only because they can model complex and nonlinear relationships, but also because their layered architectures can learn task-relevant latent representations and feature interactions from transaction features (Goodfellow *et al.*, 2016; Bishop and Bishop, 2023). This automatic representation learning is particularly relevant when the model must discover compact combinations of correlated or transformed variables rather than depend entirely on manually engineered feature interactions. For instance, Alarfaj *et al.* (2022) achieved a detection accuracy of 99.72% using a 20-layer convolutional neural network (CNN), outperforming conventional ML models. By integrating RF with generative adversarial networks (GANs), Akouhar *et al.* (2024) further enhanced detection performance while effectively mitigating class imbalance. Deviation networks (DevNet), introduced by Pang *et al.* (2019), leverage limited labelled data to detect anomalies, making them particularly suitable for fraud detection scenarios in which genuine fraud cases are scarce.

Despite these advances, most DL-based approaches rely on single-task learning frameworks that prioritize classification accuracy alone and do not exploit the benefits of multi-task learning. Furthermore, model interpretability is often overlooked, even though it is essential in high-stakes financial environments that demand transparent and auditable decision-making (Hassija *et al.*, 2024; McDonnell *et al.*, 2023). These limitations motivate the exploration of explainable artificial intelligence (XAI) and multi-task learning techniques for CCFD.

2.3. *XAI for CCFD*

The increasing adoption of ML in high-stakes domains such as CCFD has heightened the demand for transparency and interpretability. XAI addresses this need by providing mechanisms that render complex model behaviour understandable to human stakeholders. Within XAI research, two dominant paradigms have emerged: *post-hoc explanations* and *intrinsic interpretability* (Du *et al.*, 2020). These paradigms differ fundamentally in how transparency is achieved.

2.3.1. *Post-Hoc Explanations*

Post hoc explanation methods generate interpretations after model training, typically by employing an external explainer to approximate a trained model's decision logic without altering its internal structure (Du *et al.*, 2020). SHAP is a family of additive feature-attribution methods whose explainer variants differ in their dependence on model structure. Some variants, such as KernelSHAP, are designed to be model-agnostic, whereas others, such as TreeSHAP and DeepExplainer, exploit model-specific structure. In the context of CCFD, SHAP-based post-hoc attribution methods and local interpretable model-agnostic explanations (LIME) (Ribeiro *et al.*, 2016) are commonly used approaches;

SHAP should therefore not be characterized as uniformly model-agnostic (Lundberg and Lee, 2017). Beyond SHAP and LIME, several studies report that layer-wise relevance propagation (LRP) can provide more efficient and informative attributions for CNN-based detectors applied to tabular fraud data (Ullah *et al.*, 2021). Another line of research integrates autoencoders with post hoc or surrogate-based explanation modules, enabling anomaly evidence extracted from latent representations to be complemented by human-interpretable feature rankings, decision rules, or representative exemplars (Sowmya *et al.*, 2025; Rao *et al.*, 2020).

2.3.2. *Intrinsic Interpretability*

Intrinsic interpretability is achieved by designing self-explanatory models in which interpretability is embedded directly in the model architecture, rather than approximated post hoc (Du *et al.*, 2020). Deep symbolic classification (DSC) (Visbeek *et al.*, 2024) exemplifies this paradigm by replacing opaque neural architectures with concise, closed-form mathematical expressions. This design yields inherently transparent decision and model-level explanations while maintaining competitive performance against strong baselines such as XGBoost. Moreover, by directly optimizing performance metrics such as the F1-score and DSC, DSC naturally mitigates class imbalance without relying on oversampling or undersampling techniques. SEFraud (Li *et al.*, 2024) extends this intrinsic approach by employing heterogeneous graph transformer networks with learnable feature and edge masks, enabling joint fraud detection and the generation of fine-grained self-explanations.

Although intrinsic XAI models represent important progress toward interpretable fraud detection, they are not universally interchangeable with the MTL setting considered here. Visbeek *et al.* (2024) demonstrate that intrinsic XAI can be applied to tabular transaction data through a concise, closed-form symbolic classifier trained on the PaySim dataset. However, their deep symbolic classification approach relies on a dedicated symbolic-regression search space and a single classification objective, whereas our framework uses a neural MTL architecture with reconstruction and transaction-amount prediction as auxiliary tasks. Our aim is therefore not to argue that intrinsic XAI is inapplicable to tabular fraud detection, but to study a different hybrid design that combines shared representation learning, reconstruction-based anomaly-sensitivity signals, and post-hoc classifier attribution.

2.4. *Multi-Task Learning (MTL)*

MTL is an ML paradigm in which multiple related tasks are learned jointly. A typical MTL architecture consists of shared layers whose parameters are updated by multiple task objectives, alongside task-specific layers that produce the outputs for individual tasks (Zhang and Yang, 2021; Chen *et al.*, 2024). This arrangement provides a common representation while keeping the task outputs and their corresponding objectives distinct. Its effect depends on the relationships among the tasks, the model architecture, and the optimization procedure.

To jointly optimize multiple objectives, MTL requires an effective strategy for combining task-specific loss functions. The simplest approach is uniform weighting, in which

all losses are summed equally. However, this strategy is often suboptimal, as tasks may converge at different rates and exhibit conflicting gradient directions. To overcome these issues, more advanced loss-balancing techniques have been proposed, including gradient conflict mitigation methods such as projecting conflicting gradients (PCGrad) (Yu *et al.*, 2020). A detailed discussion of these approaches is provided in Section 3.3.

Among existing MTL-based fraud detection approaches, MTCNN (Qu *et al.*, 2024) and HMTL (Chen *et al.*, 2022) are the most closely related to our work. However, their objectives and design choices differ from ours in two key aspects. First, they primarily employ MTL as a static performance enhancement mechanism, whereas we adopt a curriculum learning strategy to schedule tasks during training dynamically. Second, these studies focus primarily on predictive performance and do not analyse reconstruction-based anomaly evidence together with post-hoc classifier attribution, which is the complementary model-analysis setting investigated here.

3. Methodology

3.1. *Overview of the Proposed MTL Framework*

This study aims to improve the accuracy of CCFD and provide complementary evidence for model analysis. We propose MTLFraudDetect, a multi-task framework designed for CCFD. MTLFraudDetect jointly learns three tasks: data reconstruction (TASK A), fraud classification (TASK B), and transaction amount prediction (TASK C). TASK A employs an autoencoder to reconstruct input features. The resulting reconstruction error highlights atypical transaction patterns and provides model-derived anomaly evidence. TASK B estimates the probability of fraud, and its outputs can be further analysed using feature attribution methods, yielding a separate post-hoc decision-attribution layer. TASK C predicts transaction amounts and serves as an auxiliary objective through which spending-related information is included in the joint training process. All tasks share a common encoder while maintaining task-specific heads. Consequently, the task objectives update a common encoder, while each task retains its own output head. The following sections provide a detailed description of the proposed framework and its design choices.

3.2. *MTL Model Architecture*

The overall architecture of MTLFraudDetect is illustrated in Fig. 1. The architecture comprises an autoencoder (AE), a shared feature network, a classifier and a regressor. Within this design, fraud classification is treated as the primary task, while the shared representation supports both the primary classification objective and the auxiliary transaction-amount prediction objective.

For notational convenience, we use a common representation for the three tasks. Let $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,d}]^\top \in \mathbb{R}^d$ denote the vector of non-label transaction features available to the model after preprocessing. Let $y_i \in \{0, 1\}$ denote the class label of transaction i, where $y_i = 1$ and $y_i = 0$ represent fraudulent and legitimate transactions, respectively.

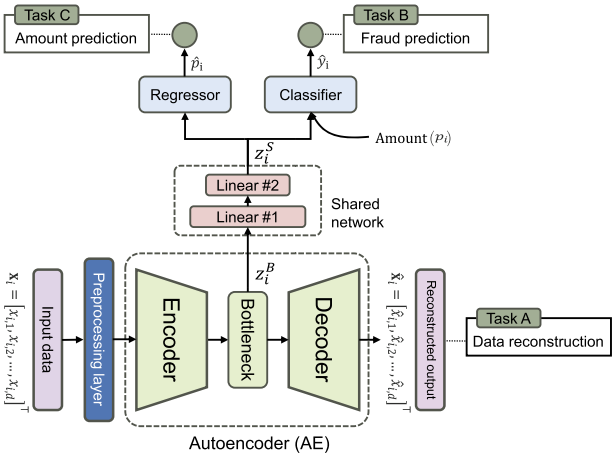


Fig. 1. MTL model architecture.

Let p_i denote the payment amount associated with transaction i, and write the observed transaction record as $t_i = (\mathbf{x}_i, p_i)$. In the present credit-card dataset, $\mathbf{x}_i$ contains the pre-processed *Time* and V1–V28 features, p_i is *Amount*, and y_i is *Class*.

3.2.1. *AE for Data Reconstruction (Task A)*

In Task A, an AE is employed to reconstruct input data and learn richer feature relationships that support fraud detection. Both the encoder and decoder are implemented as multilayer perceptrons (MLPs). The encoder progressively compresses the input into a low-dimensional representation, while the decoder reconstructs the original features from this compressed representation. Reconstruction is trained using the mean squared error (MSE) loss; accurate reconstruction indicates that the bottleneck layer captures essential information.

To prevent data leakage, the payment amount p_i and class label y_i are excluded from the AE input. This ensures that the bottleneck representation cannot directly encode transaction amounts or class labels, which would otherwise render downstream prediction tasks trivial. The AE also functions as a reconstruction-based anomaly detector, learning the manifold of normal transactions. Consequently, elevated reconstruction errors for fraudulent samples serve as a warning sign. This behaviour provides model-derived global and feature-level anomaly evidence, but does not replace attribution of the classifier's black-box decision (see Section 3.5).

Let f_E and f_D denote the encoder and decoder, respectively. For transaction i, the encoder produces a bottleneck representation $z_i^B = f_E(\mathbf{x}_i; \theta_E)$, where θ_E represents the encoder parameters. The decoder reconstructs the input as $\hat{\mathbf{x}}_i = f_D(z_i^B; \theta_D)$, where $\hat{\mathbf{x}}_i$ is the reconstructed feature vector and θ_D denotes the decoder parameters.

The reconstruction loss for Task A is denoted by $\mathcal{L}_1$. Specifically, we adopt MSE to quantify the discrepancy between the original feature vector $\mathbf{x}_i$ and its reconstruction $\hat{\mathbf{x}}_i$.

For one transaction, the reconstruction loss is defined as

$$\ell_{A,i} = \frac{1}{d}\sum_{j=1}^{d}(x_{i,j} - \hat{x}_{i,j})^2,$$

where d is the number of input features, and $x_{i,j}$ and $\hat{x}_{i,j}$ are the original and reconstructed values of feature j for transaction i, respectively. For n training transactions, the Task A objective is the average per-transaction loss,

$$\mathcal{L}_1 = \frac{1}{n}\sum_{i=1}^{n}\ell_{A,i}.$$

The parameters θ_E and θ_D are optimized by minimizing $\mathcal{L}_1$.

3.2.2. *Shared Feature Network*

The shared network transforms the bottleneck representation z_i^B produced by the AE. Let f_S denote the shared feature network and θ_S its parameters. The shared representation for transaction i is defined as $z_i^S = f_S(z_i^B; \theta_S)$. The shared network consists of a two-layer MLP, with each layer using ReLU activation and dropout applied between layers to improve training stability. The same shared-network parameters generate the representation supplied to both task-specific heads. Thus, z_i^S simultaneously supports fraud classification (Task B) and transaction amount prediction (Task C).

3.2.3. *Fraud Classification (Task B)*

The classifier C performs fraud detection by taking the shared latent representation z_i^S and concatenating it with the transaction amount p_i. For transaction i, the classifier output is defined as $\hat{y}_i = \sigma(C(z_i^S \oplus p_i; \theta_C))$, where $\oplus$ denotes concatenation, σ is the sigmoid function, and θ_C represents the classifier parameters. The output $\hat{y}_i \in (0, 1)$ is the predicted probability that transaction i is fraudulent. When a hard class decision is required, this probability can be converted to a binary decision by applying a selected threshold. The per-transaction binary cross-entropy (BCE) loss for Task B is

$$\ell_{B,i} = -\big[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\big].$$

For n training transactions, the Task B objective is

$$\mathcal{L}_2 = \frac{1}{n}\sum_{i=1}^{n}\ell_{B,i}.$$

3.2.4. *Transaction Amount Prediction (Task C)*

The regressor R predicts the transaction amount and follows a structure similar to that of the classifier. It receives the shared representation z_i^S as input. The regressor output is

expressed as $\hat{p}_i = R(z_i^S; \theta_R)$, where θ_R denotes the trainable parameters and $\hat{p}_i$ is the predicted transaction amount. The per-transaction MSE loss for Task C is

$$\ell_{C,i} = (p_i - \hat{p}_i)^2,$$

where p_i is the ground-truth transaction amount. For n training transactions, the Task C objective is

$$\mathcal{L}_3 = \frac{1}{n} \sum_{i=1}^{n} \ell_{C,i}.$$

3.3. *Joint Optimization of Multiple Loss Functions*

Jointly learning the three tasks within a single framework is nontrivial, particularly due to potential conflicts among task objectives. This section introduces three strategies for combining task-specific losses to enable stable and effective joint optimization.

3.3.1. *Uniform Weighting*

Uniform weighting is the most straightforward strategy for combining multiple task losses. Under this approach, the loss functions of all tasks are summed into a single aggregated objective. Formally, the total loss is defined as $\mathcal{L}_{tot} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$. This strategy assigns equal importance to all tasks during optimization, making it simple and easy to implement. However, in practice, tasks that converge faster or exhibit larger loss magnitudes tend to generate stronger gradients and dominate the training process. As a result, tasks that learn more slowly may be under-optimized. To mitigate this imbalance, we adopt the DWA method, which is described in the following subsection.

3.3.2. *DWA*

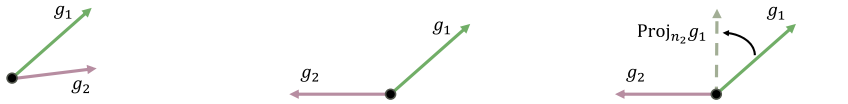
DWA (Liu *et al.*, 2019) adaptively balances the contributions of multiple tasks during training, ensuring that no single task dominates the optimization process. At training step t, let λ_i $(i = 1, 2, 3)$ denote the weight associated with the loss of the i-th task. The total loss at step t is then expressed as a weighted sum. That is, $\mathcal{L}_{tot} = \sum_{i=1}^{3} \lambda_i(t)\mathcal{L}_i(t)$.

The core principle of DWA is to assign weights inversely proportional to their learning speed. Tasks that exhibit slower loss reduction are assigned higher weights, while tasks that learn faster receive lower weights. Specifically, the weight $\lambda_k(t)$ for the k-th task computed as:

$$\lambda_k(t) = \frac{3 \times \exp(w_k(t-1)/T)}{\sum_{i=1}^{3} \exp(w_i(t-1)/T)},$$

here, $w_k(t-1)$ quantifies the relative change in loss for task k between two consecutive training steps:

$$w_k(t-1) = \frac{\mathcal{L}_k(t-1)}{\mathcal{L}_k(t-2)}.$$



(a) Gradients from two non-conflicting tasks.

(b) Gradients from two conflicting tasks.

(c) The projection of g_1 onto the normal vector of g_2.

Fig. 2. The idea of PCGrad.

The terms $\mathcal{L}_k(t-1)$ and $\mathcal{L}_k(t-2)$ represent the average loss values of task k at steps $t-1$ and $t-2$, respectively. A larger value of $w_k(t-1)$ indicates slower convergence for task k. The temperature parameter T controls the smoothness of the weighting scheme. Larger values of T yield more uniform task weights, whereas smaller values accentuate differences among tasks. The constant factor 3 corresponds to the total number of tasks and ensures that the sum of task weights satisfies $\sum_{i=1}^{3} \lambda_i(t) = 3$. Overall, DWA discourages excessive emphasis on simpler tasks and reallocates learning capacity toward more challenging objectives.

3.3.3. *Projecting Conflicting Gradients (PCGrad)*

Conflicting gradients are a common challenge in MTL and arise when a model is optimized for multiple objectives simultaneously. Specifically, gradient conflict occurs when gradients associated with different tasks lose point in substantially different or opposing directions in parameter space. In such cases, parameter updates that improve one task may degrade the performance of others, thereby limiting overall learning efficiency. To address this issue, we employ a gradient surgery technique known as PCGrad (Yu *et al.*, 2020). This method modifies task-specific gradients during backpropagation to reduce inter-task interference, enabling more aligned parameter updates and more stable joint optimization.

One effective gradient surgery technique is PCGrad. Figure 2 illustrates the working mechanism of PCGrad. In this framework, two tasks i and j are considered conflicting when the cosine similarity between their corresponding gradient vectors g_i and g_j is negative. As shown in Fig. 2(a), g_1 and g_2 represent gradients from two non-conflicting tasks. In contrast, Fig. 2(b) illustrates a conflicting scenario, where the cosine similarity between g_1 and g_2 is negative.

When tasks i and j are in conflict, PCGrad resolves this issue by modifying the gradient g_i by projection onto the normal plane of g_j, as illustrated in Fig. 2(c). In this figure, $Proj_{n2}$ denotes the projection operator, and $Proj_{n2}\ g_1$ represents the projection of g_1 onto the normal vector of g_2. This operation effectively removes the component of g_1 that conflicts with g_2, thereby reducing the gradient interference between tasks.

3.4. *Model Training*

To train the proposed MTLFraudDetect model, we adopt a curriculum learning strategy governed by a transition ratio (TR). Unlike conventional curriculum learning, where models progress from easier to harder tasks, our approach starts with a holistic training phase in which the model is jointly optimized on multiple objectives (i.e. fraud detection, reconstruction, and regression). This phase enables the model to learn shared and generalized

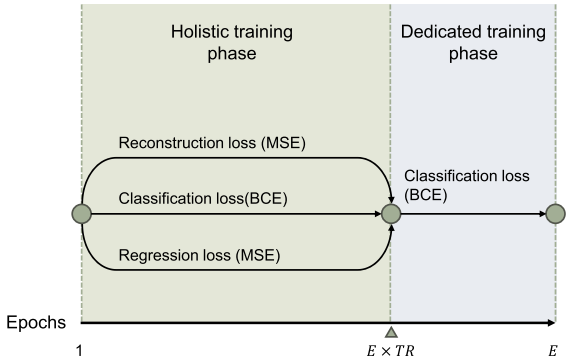


Fig. 3. Two-phase training mechanism.

representations across tasks. Subsequently, training transitions to a dedicated phase that focuses exclusively on the primary fraud detection task. This two-phase strategy balances representation learning with task-specific optimization. Importantly, we do not assume that multi-task learning is universally superior. Instead, we hypothesize that both the timing and duration of the holistic training phase play a critical role in determining final performance. By tuning the transition ratio, we flexibly control the balance between holistic and dedicated training, allowing us to empirically identify when and how MTL most effectively enhances fraud detection performance.

Figure 3 illustrates the two-phase training strategy employed in this study, comprising a holistic training phase followed by a dedicated training phase. During the holistic phase, all three tasks are jointly trained and their respective loss functions are combined to update the model parameters. The duration of this phase is controlled by the TR, which specifies the proportion of the total number of training epochs E allocated to holistic learning. Consequently, the holistic phase lasts for $E \times TR$ epochs. After completing $E \times TR$ epochs, training transitions to the dedicated phase.

In the dedicated phase, the model is optimized exclusively for fraud transaction detection using only the classification loss based on BCE. This phase refines the model for fraud detection by adjusting decision boundaries and optimizing feature utilization based on the representations learned during the holistic phase. Overall, the two-phase training scheme integrates the representational benefits of multi-task learning with task-specific fine-tuning, thereby improving robustness and effectiveness in addressing fraud detection challenges.

3.5. *Three Levels of Complementary Model Evidence in MTLFraudDetect*

The architecture of MTLFraudDetect provides a three-level model-analysis framework that combines complementary evidence about fraud detection behaviour. By leveraging its multi-task learning structure, we organize the analysis into three complementary levels. Specifically, the framework provides three complementary levels of model evidence: (1) transaction-level reconstruction-based anomaly evidence, (2) feature-level localization

based on reconstruction errors, and (3) decision-level post-hoc attribution for examining the classifier's decision function. The first two outputs are generated by the auxiliary AE (Task A) and quantify deviations under the reconstruction objective. They are not claimed to be formal explanations or faithful attributions of the classifier's fraud decision.

Transaction-level reconstruction-based anomaly evidence. At the transaction level, we use the reconstruction error $\mathrm{RE}(x_i)$ produced by Task A as an anomaly-sensitivity score. This behaviour is expected in cases of severe class imbalance, where legitimate transactions substantially outnumber fraudulent transactions in the training data. As a result, the AE primarily minimizes reconstruction loss for normal transactions and learns their underlying data manifold more accurately. Consequently, fraudulent transactions, being less frequent and potentially structurally different, may produce higher $\mathrm{RE}(x_i)$ than normal ones. This score measures deviation under the reconstruction objective and is not treated as a direct explanation of the classifier's final decision. Formally, for a transaction $x_i \in \mathbb{R}^d$, the AE generates a reconstructed output $\hat{x}_i$, where d denotes the number of input features and $x_{i,j}$ represents the value of the j-th feature of transaction i. $\mathrm{RE}(x_i)$ is then defined as $\frac{1}{d}\sum_{j=1}^{d}(x_{i,j} - \hat{x}_{i,j})^2$.

Feature-level reconstruction-error localization. In addition to transaction-level anomaly evidence, Task A enables finer-grained localization of reconstruction error. This layer identifies which input dimensions contribute most strongly to the reconstruction-based anomaly score; it does not establish a causal explanation of fraud or of the classifier's decision. For the j-th feature of transaction x_i, the **feature-wise reconstruction error** is defined as $\mathrm{FRE}(x_{i,j}) = (x_{i,j} - \hat{x}_{i,j})^2$.

A larger $\mathrm{FRE}(x_{i,j})$ indicates that the j-th feature of transaction x_i is poorly reconstructed by the AE. Under severe class imbalance, the AE is expected to fit the dominant patterns of normal transactions. Consequently, features that deviate from these normal patterns tend to produce larger feature-wise reconstruction errors. This mechanism provides model-derived feature-level anomaly evidence by identifying which input dimensions contribute most strongly to the overall anomaly score $\mathrm{RE}(x_i)$.

Decision-level post-hoc attribution. While the first two levels characterize deviations under **TASK A**, they do not fully explain the final fraud decision produced by the classifier in **TASK B**. This limitation motivates the addition of a third, decision-level attribution layer. To this end, we employ SHAP's DeepExplainer, which uses model-specific information for neural-network models, to generate post-hoc attributions for the fraud classification task.

This decision-level post-hoc attribution complements the two reconstruction-based evidence levels. The reconstruction outputs characterize deviation under the reconstruction objective, while the SHAP layer attributes the classifier's output under the fraud-detection objective. Together, these components form a three-level model-analysis framework linking transaction-level anomaly evidence, feature-wise error localization, and classifier decision attribution.

4. Experiments

This section presents a comprehensive experimental evaluation of the proposed MTL-FraudDetect framework and examines the contributions of its key design components. The experiments are organized to address the following research questions:

1. How does MTLFraudDetect compare with representative published CCFD methods under Precision, Recall, F1-score, and AUC-PR, while accounting for differences in evaluation protocols?
2. How does the TR in the proposed two-phase training strategy affect fraud detection performance, and which TR yields the best results under a fixed training budget?
3. Among uniform weighting, DWA, and PCGrad, which multi-task optimization strategy delivers the strongest fraud detection performance within the proposed framework?
4. How does bottleneck dimensionality affect overall model performance?
5. Can the proposed framework provide a multi-level model-analysis pipeline that integrates reconstruction-based anomaly evidence with post-hoc decision attribution?

We first describe the experimental setup, including the dataset, preprocessing procedures, model configuration, and evaluation metrics. We then present the experimental results and discuss the contribution of each component of the proposed framework in addressing the above research questions.

4.1. *Experimental Setup*

4.1.1. *Dataset Overview*

The European Credit Card Fraud Dataset[1] is a publicly available collection of real-world credit card transactions from a European financial institution. Owing to its realistic characteristics, it has been widely used in credit card fraud and anomaly detection research (Fanai and Abbasimehr, 2023; Forough and Momtazi, 2021). The dataset contains 284 807 transactions, of which only 492 (0.172%) are fraudulent, resulting in severe class imbalance that presents substantial challenges for both model training and evaluation.

The dataset comprises 30 numerical features, including 28 anonymized principal component analysis (PCA) features (V1–V28) and two original variables: **Time**, which represents the elapsed time in seconds since the first recorded transaction, and **Amount**, which denotes the transaction value in euros. The dataset is publicly available through Kaggle (i.e. https://www.kaggle.com/mlg-ulb/creditcardfraud).

For all experiments, the dataset is first partitioned using a stratified holdout split, with 25% of the observations reserved as an independent test set and the remaining 75% used as the development set. Hyperparameter optimization is performed on the development set using shuffled stratified five-fold cross-validation. The resulting validation folds generate out-of-fold (OOF) predictions, which are pooled for model development and hyperparameter selection, whereas the independent test set is reserved exclusively for the final evaluation of the selected model configuration.

[1] The dataset is available at https://www.kaggle.com/mlg-ulb/creditcardfraud

4.1.2. *Dataset Preprocessing*

Following established practice, the dataset is preprocessed using a sequence of transformations performed exclusively on the training data within each cross-validation fold. All data-dependent preprocessing parameters are estimated only after the initial holdout split and separately within each training fold. Specifically, for fold k, each preprocessing component is fitted solely on $train_k$ and subsequently applied, without refitting, to $train_k$, val_k, and the held-out test set. Consequently, neither validation nor test samples contribute to estimating scaling parameters or outlier thresholds.

The preprocessing pipeline begins by applying a $\log(1+\text{Amount})$ transformation to the *Amount* feature. The continuous features are then scaled according to the preprocessing strategy selected during hyperparameter optimization (HPO). This approach reduces the influence of extreme values, which are common in financial transaction data.

The distribution of transaction amounts is highly skewed, with a relatively small number of transactions exhibiting exceptionally large values. Because both distance-based and gradient-based learning algorithms are sensitive to extreme values, the logarithmic transformation compresses the dynamic range of the *Amount* feature, reducing skewness while preserving the relative ordering of transaction values.

For the primary preprocessing configuration explored during HPO, RobustScaler is applied to *Time* and the log-transformed *Amount*, whereas StandardScaler is applied to the PCA features (V1–V28). RobustScaler estimates scaling parameters from the median and interquartile range (IQR), making it less sensitive to extreme values than StandardScaler, which relies on the mean and standard deviation. The HPO search space also includes an outlier-aware preprocessing option that estimates IQR-based thresholds using only legitimate transactions in $train_k$, clips values outside the estimated bounds, and generates corresponding outlier indicator variables.

To address the severe class imbalance, Borderline-SMOTE (Han *et al.*, 2005) is applied only to the transformed training subset after all training-derived preprocessing steps have been completed. Unlike conventional oversampling methods that primarily replicate minority-class examples, Borderline-SMOTE generates synthetic minority samples through interpolation, focusing on instances located near the decision boundary. The validation and test sets retain their original class distributions and are never resampled.

Each fold-specific preprocessing pipeline is retained together with its corresponding trained model. During HPO, pooled out-of-fold AUC-PR serves as the optimization objective for selecting both the preprocessing strategy and the model hyperparameters. This metric is chosen because previous studies have shown that precision-recall-based evaluation is informative for highly imbalanced datasets, as precision more directly reflects the impact of false-positive predictions (Davis and Goadrich, 2006; Gray *et al.*, 2011; Gaudreault *et al.*, 2021). During the final evaluation, the preprocessing pipeline fitted for the k-th fold is applied to the held-out test set before the transformed samples are evaluated by the corresponding k-th model. The final prediction for each test instance is obtained by aggregating the outputs of the five fold-specific models using soft voting.

Table 1
Experimental parameters.

Parameter	Value
Batch size	256
Optimizer	AdamW
Learning rate	0.002 (with cosine learning rate scheduler)
Dropout rate	0.2
Bottleneck dimension	256, 512, 1024, 2048
Epochs	80
Transition Ratio (*TR*)	0.0, 0.25, 0.33, 0.50, 0.66, 0.75, 1.00

4.2. *Performance Metrics*

Given the binary classification nature of CCFD, model performance is evaluated using Precision, Recall, F1-score, and AUC-PR. Precision measures the reliability of fraud predictions, whereas Recall measures the model's ability to identify fraudulent transactions. The F1-score provides a balanced assessment of the trade-off between these two metrics (Akouhar *et al.*, 2024). Because the fraud class is highly imbalanced, AUC-PR is emphasized as the primary performance metric, as it summarizes the precision-recall trade-off across all classification thresholds and is more informative than ROC-based metrics for highly imbalanced datasets (Davis and Goadrich, 2006; Gray *et al.*, 2011; Gaudreault *et al.*, 2021).

4.3. *Experimental Settings*

All experiments were conducted on a workstation equipped with an NVIDIA Tesla V100 GPU and 32 GB of RAM. The proposed model was trained for 80 epochs using the hyperparameter configuration summarized in Table 1, including a batch size of 256, the AdamW optimizer with weight decay, an initial learning rate of 0.002, and a cosine learning rate scheduler to promote rapid convergence and stable training. A dropout rate of 0.2 was used to reduce overfitting. Unless otherwise specified, the default autoencoder bottleneck dimension was set to 1024. This parameter determines the size of the latent representation and directly affects the performance of both the classifier and the regressor. The alternative bottleneck dimensions considered in the experiments are also listed in Table 1; their corresponding performance results are presented in Section 5.

5. Results and Discussion

5.1. *Comparison with Existing Works*

Table 2 presents a contextual comparison between MTLFraudDetect and representative published CCFD methods. Missing entries, denoted by "–", indicate performance metrics that were not reported in the original studies. For example, El Bazi *et al.* (2024) reported

Table 2
Contextual comparison with published CCFD results under heterogeneous evaluation protocols.

Method	Precision	Recall	F1	AUC-PR
MTLFraudDetect (ours; AUC-PR-tuned)	0.9591	0.7642	0.8506	0.8814
XGBoost (Sudhakar and Kaliyamurthie, 2023)	0.8784	0.7420	0.8235	–
SVM (Fanai and Abbasimehr, 2023)	0.8650	0.7103	0.7800	0.6333
AE-DNN (Fanai and Abbasimehr, 2023)	0.9670	0.7213	0.8263	0.7746
Autoencoder (Fanai and Abbasimehr, 2023)	0.9456	0.7131	0.8263	0.7746
DevNet (Pang *et al.*, 2019)	–	–	–	0.6900
UAAD-FDNet (Jiang *et al.*, 2023)	0.9795	0.7553	0.8529	–
CatBoost (Leevy *et al.*, 2023)	–	–	–	0.8567
GRU (Mienye and Jere, 2024)	0.8571	0.7959	0.8254	–
OptiStack (El Bazi *et al.*, 2024)	0.8800	0.8600	0.8700	–

ROC-AUC but not AUC-PR, whereas several other studies omitted one or more of the comparison metrics shown in the table.

Recent studies have achieved strong fraud detection performance using a variety of carefully designed approaches. OptiStack employs a stacking ensemble that combines XGBoost, CatBoost, and LightGBM with logistic regression as the meta-learner and achieves the highest reported F1-score (0.8700) among the methods summarized in Table 2. At the reported operating points, OptiStack also reports higher Recall (0.8600 versus 0.7642) and F1-score (0.8700 versus 0.8506) than MTLFraudDetect, whereas MTLFraudDetect reports higher Precision (0.9591 versus 0.8800). UAAD-FDNet incorporates channel-wise feature attention into an autoencoder-based adversarial framework and reports the highest Precision (0.9795). Overall, the results show that no single method consistently achieves the best performance across all evaluation metrics; these cross-study differences remain contextual because the evaluation protocols are heterogeneous.

Among the methods reporting AUC-PR, MTLFraudDetect achieves the highest value (0.8814). It also attains the third-highest Recall (0.7642), F1-score (0.8506), and Precision (0.9591) among the methods for which the corresponding metrics are available. These results are obtained through joint optimization of data reconstruction, fraud classification, and transaction amount prediction during the multi-task training stage. The subsequent curriculum-based training phase progressively shifts the optimization focus toward fraud classification, enabling further refinement of the learned representation and classification boundary for the primary task. Collectively, these findings demonstrate that MTLFraudDetect is a competitive multi-task framework for CCFD, although the comparison remains contextual because the published studies employ heterogeneous evaluation protocols.

5.2. *Effect of Bottleneck Dimensionality*

In **MTLFraudDetect**, the bottleneck layer is intended to learn compact latent representations that capture informative patterns distinguishing legitimate transactions from fraudulent ones. To evaluate its influence, we vary the bottleneck dimensionality and summarize the corresponding results in Table 3.

As shown in Table 3, a bottleneck dimension of **1024** achieves the highest F1-score (0.8506) and AUC-PR (0.8814), while maintaining a Precision of 0.9591 and a Recall

Table 3
Effect of bottleneck dimensionality.

Bottleneck Dim.	Precision	Recall	F1	AUC-PR
256	0.9890	0.7317	0.8411	0.8721
512	1.0000	0.6991	0.8229	0.8762
1024	0.9591	0.7642	0.8506	0.8814
2048	0.8174	0.8373	0.8273	0.8800

Table 4
Comparison of different MTL optimization strategies.

Optimization strategies	Precision	Recall	F1	AUC-PR
Uniform	0.8833	0.8211	0.8511	0.8472
DWA	0.9367	0.8021	0.8642	0.8554
PCGrad	0.9591	0.7642	0.8506	0.8814

of 0.7642. Smaller bottleneck dimensions produce higher Precision but lower Recall, resulting in reduced F1-score and AUC-PR. This pattern suggests that more compact latent representations improve prediction confidence but reduce the model's ability to identify fraudulent transactions. Increasing the bottleneck dimension to **2048** yields the highest Recall (0.8373), although Precision and F1-score decrease, while AUC-PR remains comparable to that of the 1024-dimensional model. Overall, these results indicate a trade-off among the evaluation metrics, with a bottleneck dimension of **1024** providing the best overall balance and the strongest ranking performance under the AUC-PR-based model selection criterion.

5.3. *MTL Optimization Strategies*

Table 4 compares the performance of different loss-balancing strategies within the proposed framework. PCGrad achieves the highest Precision (0.9591) and AUC-PR (0.8814), whereas DWA produces the highest F1-score (0.8642), and uniform weighting yields the highest Recall (0.8211). These findings indicate that the choice of loss-balancing strategy involves a trade-off between threshold-dependent classification metrics and threshold-independent ranking performance under severe class imbalance. Under the AUC-PR-based model selection criterion, PCGrad is preferred despite its lower Recall, while the uniform and DWA results show that improvements at a fixed classification threshold do not necessarily translate into a higher AUC-PR.

5.4. *Impact of TR*

As discussed in Section 3.4, the proposed training strategy consists of two phases: a holistic training phase followed by a dedicated training phase, with the transition controlled by the transition ratio (TR). The value of TR determines the proportion of the total training budget allocated to holistic learning. Specifically, $\mathrm{TR} = 1$ corresponds to training entirely in the holistic phase, whereas $\mathrm{TR} = 0$ represents training exclusively in the dedicated phase

Table 5
Post-selection sensitivity analysis of TR with all other HPO-selected hyperparameters fixed.

TR	Precision	Recall	F1	AUC-PR
0.00	0.9878	0.6585	0.7902	0.8634
0.25	0.9208	0.7886	0.8496	0.8498
0.33	0.9000	0.8048	0.8497	0.8823
0.50	0.9489	0.7560	0.8416	0.8789
0.66	0.9591	0.7642	0.8506	0.8814
0.75	0.9565	0.7154	0.8186	0.8713
1.00	0.8862	0.8021	0.8421	0.8490

without auxiliary tasks. During HPO, TR = 0.66 produced the highest pooled out-of-fold AUC-PR on the development set and was therefore retained in the selected primary configuration. After selection, a one-factor sensitivity analysis was conducted by varying TR while holding the other HPO-selected hyperparameters fixed; the corresponding held-out test results are summarized in Table 5.

In this post-selection sensitivity analysis, TR = 0.33 attains the highest held-out test AUC-PR (0.8823), marginally exceeding the 0.8814 obtained by the HPO-selected TR = 0.66 configuration by 0.0009. The TR = 0.66 configuration nevertheless yields higher Precision (0.9591 versus 0.9000) and F1-score (0.8506 versus 0.8497), whereas TR = 0.33 yields higher Recall (0.8048 versus 0.7642). The small reversal in test-set AUC-PR does not invalidate the development-set selection: hyperparameters were selected using pooled out-of-fold AUC-PR, and their ordering need not be identical on an independent finite test sample. Moreover, because the sensitivity analysis changes only TR without jointly re-optimizing the remaining hyperparameters, it does not establish TR = 0.33 as a globally superior configuration. Instead, the close AUC-PR values indicate that performance is relatively stable across these two transition schedules while also suggesting possible interactions between TR and the other hyperparameters. Accordingly, the TR = 0.66 model remains the prespecified primary configuration, and the TR = 0.33 result is reported as a post-selection sensitivity finding rather than used to revise the main benchmark result.

5.5. *Complementary Reconstruction Evidence and Post-hoc Attribution*

This section evaluates the proposed hybrid model-analysis pipeline of MTLFraudDetect and examines three complementary levels of evidence for fraud detection. Within this pipeline, Task A provides reconstruction-based anomaly evidence, while SHAP provides post-hoc decision attribution for Task B. To ensure consistency with the five-fold soft-voting evaluation framework, reconstruction evidence is aggregated across the five fold-specific Task A models. For each transaction, every model generates a reconstruction, from which a fold-specific reconstruction error is computed. The five reconstruction errors are then averaged to obtain an ensemble reconstruction score. This ensemble-level reconstruction evidence can be analysed alongside the soft-voting classifier's prediction, providing complementary information for model analysis. Although Task C contributes to learning a shared latent representation, it is not treated as an explanatory output. The following

Table 6
Five-fold ensemble reconstruction error statistics.

Class	Mean RE	Median RE	Std.	Min.	Max.
Non-Fraud	0.4895	0.3608	1.2611	0.0818	118.5810
Fraud	2.7157	1.5975	6.1562	0.1454	58.6811
Ratio	5.55	4.43	–	–	–

subsections present results obtained on the held-out test set and illustrate the complementary roles of ensemble reconstruction evidence and SHAP-based feature attribution.

5.5.1. *Transaction-Level Reconstruction Evidence*
Consistent with the framework described in Section 3.5, we evaluate the autoencoder (AE) reconstruction error (RE) as a global, model-derived anomaly-sensitivity indicator. The underlying premise is that the AE, trained predominantly on legitimate transactions, learns the distribution of normal transaction patterns but reconstructs the relatively rare and irregular patterns associated with fraudulent transactions less accurately. To evaluate this behaviour at the ensemble level, a fold-specific RE was computed for every transaction using each of the five Task A models. For transaction i in fold k, the RE was calculated as the mean squared reconstruction error across all input features. The ensemble RE was then defined as the arithmetic mean of the five fold-specific RE values. This procedure produced one ensemble RE for each of the 71 202 transactions in the held-out test set, comprising 71 079 legitimate transactions and 123 fraudulent transactions.

Table 6 summarizes the reconstruction error statistics for the two classes and reveals a clear difference in reconstruction quality. Legitimate transactions exhibit substantially lower ensemble reconstruction errors (median RE = 0.3608), whereas fraudulent transactions show markedly higher errors (median RE = 1.5975), corresponding to a 4.43-fold increase in the median reconstruction error. The fraudulent class also exhibits considerably greater variability (standard deviation = 6.1562 versus 1.2611). Although the two distributions overlap and both classes contain high-error observations, the upward shift in the fraud distribution indicates that fraudulent transactions are generally more difficult for the ensemble of fold-specific autoencoders to reconstruct.

To assess whether this distributional shift is statistically significant, we performed a one-sided Mann–Whitney U test using the transaction-level ensemble reconstruction errors. The hypotheses were defined as follows:

$\mathbf{H}_0$: Fraudulent transactions do not exhibit stochastically larger ensemble reconstruction errors than legitimate transactions.

$\mathbf{H}_1$: Fraudulent transactions exhibit stochastically larger ensemble reconstruction errors than legitimate transactions.

The test yielded $U = 7\,860\,113$ and a p-value of 2.92×10^{-53}, providing overwhelming evidence against $\mathbf{H}_0$. These results indicate that fraudulent transactions generally produce larger ensemble reconstruction errors than legitimate transactions. Accordingly, the five-fold ensemble reconstruction error serves as a useful model-derived anomaly-sensitivity indicator within the proposed hybrid model-analysis pipeline.

Table 7
Top 10 features ranked by fraud-to-non-fraud median FRE ratio.

Feature	Legitimate Med	Fraud Med	Ratio
V8	0.0983	2.5818	26.26
V14	0.2660	3.3660	12.66
V17	0.2759	3.2792	11.89
V21	0.2080	2.3559	11.33
V12	0.2575	2.1441	8.33
V10	0.1501	0.8021	5.34
V7	0.3075	1.5762	5.13
V3	0.2871	1.3605	4.74
V27	0.1959	0.9279	4.74
V11	0.2782	1.1643	4.19

5.5.2. *Feature-Level Reconstruction-Error Localization*
Building on the framework described in Section 3.5, we extend the analysis to the feature level by examining the feature-wise reconstruction error, denoted as $\mathrm{FRE}(x_{i,j})$. A higher value of $\mathrm{FRE}(x_{i,j})$ indicates that feature j of transaction x_i is reconstructed less accurately by the AE and therefore contributes more to that transaction's reconstruction error.

For each transaction–feature pair, the FRE values produced by the five fold-specific models were first averaged to obtain an ensemble feature-wise reconstruction error. The 29 input features were then ranked according to the ratio of the median FRE for fraudulent transactions to that for legitimate transactions. For brevity, Table 7 reports only the ten features with the largest median ratios. This descriptive ranking highlights the largest class-wise differences in reconstruction error and should not be interpreted as implying that the remaining features are uninformative.

All ten reported features exhibit higher median FRE values for fraudulent transactions than for legitimate transactions, with fraud-to-non-fraud median ratios ranging from 4.19 to 26.26. V8 shows the largest disparity, followed by V14, V17, V21, and V12. These findings provide feature-level evidence of deviations under the reconstruction objective and complement the transaction-level reconstruction-error analysis presented in Section 5.5.1.

In summary, feature-wise RE analysis provides model-derived, feature-level anomaly evidence by identifying the input variables that exhibit the largest class-wise differences in reconstruction error. This localized analysis complements the transaction-level ensemble RE analysis described in Section 5.5.1.

5.5.3. *Decision-Level Post-Hoc Attribution via SHAP*
The final analysis examines the decision function of the fraud classifier (Task B) using SHAP. Unlike the reconstruction-based outputs, which characterize deviations from learned transaction patterns, SHAP provides post-hoc attribution of the classifier's output to its actual inputs: the shared latent representation (Z^S) and the **Amount** feature, i.e. [Z^S; **Amount**]. SHAP values quantify input contributions to the classifier's fraud logit relative to the selected background data. Because the logit is monotonically related to the predicted fraud probability, positive SHAP values indicate contributions that increase the predicted fraud probability, whereas negative SHAP values indicate contributions that de-

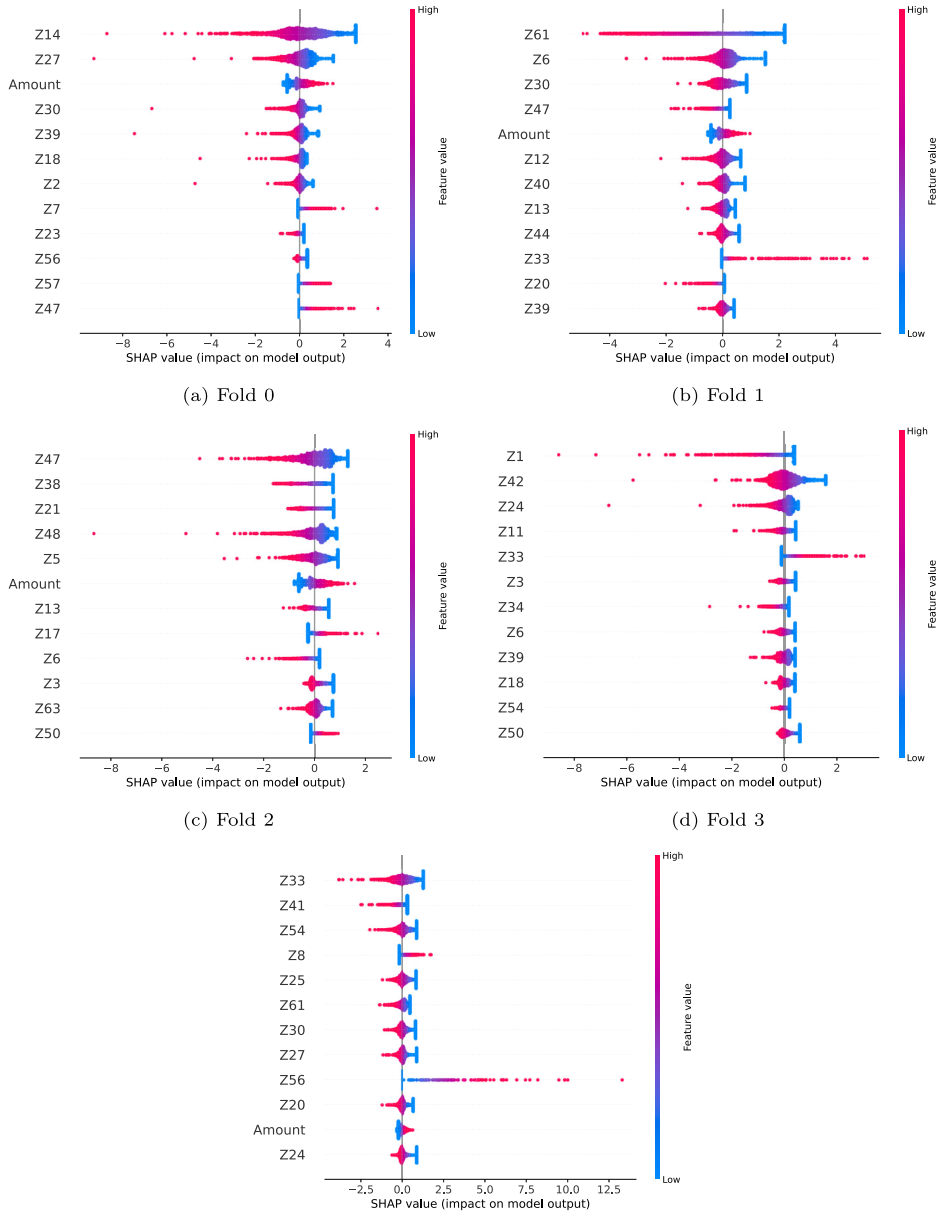
crease it. Figure 4 presents global SHAP beeswarm summaries for the five fold-specific classifiers. Each panel displays the 12 classifier inputs with the largest mean absolute SHAP values. To ensure a consistent evaluation protocol across folds, each analysis includes all 123 fraudulent transactions together with a randomly selected sample of 1 877 legitimate transactions, while 500 legitimate training transactions are used as the SHAP background set.

Across the five folds, the mean absolute SHAP values indicate that the classifier relies primarily on a subset of the learned latent components, with the ten most influential classifier inputs accounting for 49.4%–64.9% of the total mean absolute attribution. The identities of the highest-ranked latent components vary across folds because the fold-specific networks are trained independently and their latent spaces are not intrinsically aligned. Consequently, the same latent dimension should not be interpreted as representing the same semantic factor across different folds. Despite this variation, the beeswarm summaries exhibit a consistent directional pattern. The dominant latent components generally show negative SHAP values at high latent values and positive SHAP values at low latent values. Across the complete SHAP results, the **Amount** feature also exhibits a consistent directional effect in all five folds: larger transaction amounts tend to increase the fraud logit, whereas smaller amounts tend to decrease it. This directional relationship should be distinguished from the unconditional mean SHAP value. Although the mean SHAP value for **Amount** is negative in every fold (approximately -0.017 to -0.001), this statistic reflects the average contribution of the evaluated transactions relative to the background prediction rather than the conditional effect of high versus low transaction amounts. In Fold 3, **Amount** ranks 18th in terms of mean absolute SHAP value and therefore does not appear among the top 12 features shown in the beeswarm plot. Nevertheless, its directional effect remains evident in the complete SHAP results. Overall, the learned latent representation provides the primary fold-specific decision cues, whereas **Amount** contributes complementary and directionally consistent information to the classifier.

Overall, the five fold-specific SHAP summaries show that the classifier's decisions are driven primarily by the learned latent representation, with transaction amount providing an additional, directionally consistent contribution. This analysis complements the reconstruction-based anomaly evidence presented in Sections 5.5.1 and 5.5.2 by providing a post-hoc attribution of the classifier's prediction while remaining within the scope and assumptions of the SHAP framework.

6. Conclusions

This study proposed MTLFraudDetect, an evidence-augmented deep learning framework for credit card fraud detection that integrates multi-task learning (MTL) with complementary model-analysis techniques. Previous studies have largely focused on single-task models or have not fully explored how reconstruction-based anomaly evidence and post-hoc classifier attribution can be analysed together. To address this gap, the proposed framework jointly improves fraud detection performance while providing reconstruction-based



(a) Fold 0 (b) Fold 1

(c) Fold 2 (d) Fold 3

(e) Fold 4

Fig. 4. Global SHAP beeswarm summaries for the five fold-specific Task B classifiers. Each panel displays the 12 inputs with the largest mean absolute SHAP values.

anomaly evidence alongside post-hoc attribution of the classifier's predictions. We do not claim that these components alone provide formal interpretability or satisfy explanation-faithfulness criteria.

The experimental results demonstrate the effectiveness of the proposed framework. MTLFraudDetect achieves competitive fraud detection performance under the proposed evaluation protocol while providing complementary model-derived evidence for model analysis. In addition, we systematically investigated the effects of bottleneck dimensionality, curriculum-based training, and alternative multi-task optimization strategies, highlighting the performance trade-offs associated with these design choices.

Future work will extend the proposed framework to additional credit card fraud datasets and more general class-imbalanced learning problems. Evaluating MTLFraudDetect across a broader range of datasets will provide a more comprehensive assessment of its generalizability and applicability to diverse real-world fraud detection scenarios.

Concept drift also remains an important challenge in operational fraud detection, as changes in customer behaviour, payment environments, and fraud strategies can degrade model performance by increasing missed fraud and false alarms. Future work will therefore investigate the robustness of MTLFraudDetect under concept drift, including the integration of drift monitoring and periodic or adaptive retraining strategies.

7. Declaration of Generative AI and AI-Assisted Technologies in the Writing Process

During the preparation of this work, the authors used ChatGPT to assist in verifying specific research data points and to enhance the linguistic quality and grammatical structure of the manuscript. To ensure academic integrity, the authors critically reviewed and manually verified all AI-assisted outputs for factual accuracy. Subsequently, the final manuscript underwent professional human proofreading to ensure linguistic precision and clarity. The authors take full responsibility for the entire content of the publication.

Acknowledgements

The authors also thank the National Center for High-Performance Computing (NCHC) for providing computational and storage resources.

Funding

This work is supported by the Taiwan National Science and Technology Council (R.O.C.) under grants NSTC 115-2221-E-992-105 and NSTC 114-2221-E-024-017.

References

Akouhar, M., Abarda, A., Elfatini, M., Ouhssini, M. (2024). A deep learning and resampling approach to credit card fraud detection. In: *2024 11th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, IEEE, pp. 1–6.

Alarfaj, F.K., Malik, I., Khan, H.U., Almusallam, N., Ramzan, M., Ahmed, M. (2022). Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms. *IEEE Access*, 10, 39700–39715.

Bank for International Settlements (2026). Tap a card, pay by phone, but cash still holds its own. CPMI Brief No. 12. Accessed: 2026-07-11. https://www.bis.org/statistics/payment_stats/commentary2604.htm.

Bhattacharyya, S., Jha, S., Thanoon, K., Westland, J.C. (2011). Data mining for credit card fraud: a comparative study. *Decision Support Systems*, 50(3), 602–613.

Bishop, C.M., Bishop, H. (2023). *Deep Learning: Foundations and Concepts*. Springer Nature.

Caporal, J. (2025). Credit and Debit Card Market Share by Network and Issuer. Motley Fool Money. Accessed: 2026-07-11. https://www.fool.com/money/research/credit-debit-card-market-share-network-issuer/.

Chen, L., Jia, N., Zhao, H., Kang, Y., Deng, J., Ma, S. (2022). Refined analysis and a hierarchical multi-task learning approach for loan fraud detection. *Journal of Management Science and Engineering*, 7(4), 589–607.

Chen, S., Zhang, Y., Yang, Q. (2024). Multi-task learning in natural language processing: an overview. *ACM Computing Surveys*, 56(12), 1–32.

Cheng, D., Xiang, S., Shang, C., Zhang, Y., Yang, F., Zhang, L. (2020). Spatio-temporal attention-based neural network for credit card fraud detection. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, pp. 362–369.

Crawshaw, M. (2020). Multi-task learning with deep neural networks: a survey. arXiv preprint arXiv:2009.09796.

Davis, J., Goadrich, M. (2006). The relationship between precision-recall and ROC curves. In: *Proceedings of the 23rd International Conference on Machine Learning*, pp. 233–240.

Du, M., Liu, N., Hu, X. (2020). Techniques for interpretable machine learning. *Communications of the ACM*, 63(1), 68–77.

El Bazi, A., Chrayah, M., Aknin, N., Bouzidi, A. (2024). Enhancing credit card fraud detection using a stacking model approach and hyperparameter optimization. *International Journal of Advanced Computer Science and Applications*, 15(10). https://doi.org/10.14569/IJACSA.2024.01510110.

Fanai, H., Abbasimehr, H. (2023). A novel combined approach based on deep Autoencoder and deep classifiers for credit card fraud detection. *Expert Systems with Applications*, 217, 119562.

Forough, J., Momtazi, S. (2021). Ensemble of deep sequential models for credit card fraud detection. *Applied Soft Computing*, 99, 106883.

Gaudreault, J.-G., Branco, P., Gama, J. (2021). An analysis of performance metrics for imbalanced classification. In: *International Conference on Discovery Science*, Springer, pp. 67–77.

Goodfellow, I., Bengio, Y., Courville, A. (2016). *Deep Learning*. MIT Press.

Gray, D., Bowes, D., Davey, N., Sun, Y., Christianson, B. (2011). Further thoughts on precision. In: *15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE 2011)*, pp. 129–133.

Han, H., Wang, W.-Y., Mao, B.-H. (2005). Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: *International Conference on Intelligent Computing*, Springer, pp. 878–887.

Hassija, V., Chamola, V., Mahapatra, A., Singal, A., Goel, D., Huang, K., Scardapane, S., Spinelli, I., Mahmud, M., Hussain, A. (2024). Interpreting black-box models: a review on explainable artificial intelligence. *Cognitive Computation*, 16(1), 45–74.

Jiang, S., Dong, R., Wang, J., Xia, M. (2023). Credit card fraud detection based on unsupervised attentional anomaly detection network. *Systems*, 11(6), 305.

Kilickaya, O. (2025). Hybrid explainable autoencoders for credit card fraud detection: integrating deep latent representations with model-agnostic interpretability. In: *2025 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA)*, IEEE, pp. 1–7.

Krishnavardhan, N., Govindarajan, M., Rao, S.A. (2024). An intelligent credit card fraudulent activity detection using hybrid deep learning algorithm. *Multimedia Tools and Applications*, 83(40), 87621–87646.

Leevy, J.L., Hancock, J., Khoshgoftaar, T.M. (2023). Comparative analysis of binary and one-class classification techniques for credit card fraud data. *Journal of Big Data*, 10(1), 118.

Leonard, K.J. (1995). The development of a rule based expert system model for fraud alert in consumer credit. *European Journal of Operational Research*, 80(2), 350–356.

Li, K., Yang, T., Zhou, M., Meng, J., Wang, S., Wu, Y., Tan, B., Song, H., Pan, L., Yu, F., Sheng, Z., Tong, Y. (2024). SEFraud: graph-based self-explainable fraud detection via interpretative mask learning. In: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5329–5338.

Liu, S., Johns, E., Davison, A.J. (2019). End-to-end multi-task learning with attention. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1871–1880.

Lundberg, S.M., Lee, S.-I. (2017). A unified approach to interpreting model predictions. In: *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 4768–4777.

McDonnell, K., Murphy, F., Sheehan, B., Masello, L., Castignani, G. (2023). Deep learning in insurance: accuracy and model interpretability using TabNet. *Expert Systems with Applications*, 217, 119543.

Mienye, I.D., Jere, N. (2024). Deep learning for credit card fraud detection: a review of algorithms, challenges, and solutions. *IEEE Access*, 12, 96893–96910.

Nilson Report (2025). Global Card Payment Statistics and Market Size (2025–2030). Accessed: 2026-07-11. https://nilsonreport.com/global-card-payment-statistics/.

Pang, G., Shen, C., Van Den Hengel, A. (2019). Deep anomaly detection with deviation networks. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 353–362.

Qu, B., Wang, Z., Gu, M., Yagi, D., Zhao, Y., Shan, Y., Zahradnik, F. (2024). Multi-task CNN behavioral embedding model for transaction fraud detection. In: *2024 IEEE International Conference on Data Mining Workshops (ICDMW)*, IEEE, pp. 286–292.

Rao, S.X., Zhang, S., Han, Z., Zhang, Z., Min, W., Chen, Z., Shan, Y., Zhao, Y., Zhang, C. (2020). xFraud: explainable fraud transaction detection. arXiv preprint arXiv:2011.12193.

Ribeiro, M.T., Singh, S., Guestrin, C. (2016). "Why should i trust you?" Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144.

Sowmya, P., Shetty, V., Astekar, S.S., Smitha, T., Sougandhika, T., Yashashwini, H. (2025). Credit card fraud detection using XAI techniques. In: *2025 Third International Conference on Networks, Multimedia and Information Technology (NMITCON)*, IEEE, pp. 1–5.

Sudhakar, M., Kaliyamurthie, K. (2023). A novel machine learning algorithms used to detect credit card fraud transactions. *International Journal on Recent and Innovation Trends in Computing and Communication*, 11(2), 163–168.

Ullah, I., Rios, A., Gala, V., Mckeever, S. (2021). Explaining deep learning models for tabular data using layer-wise relevance propagation. *Applied Sciences*, 12(1), 136.

Visbeek, S., Acar, E., den Hengst, F. (2024). Explainable fraud detection with deep symbolic classification. In: *World Conference on Explainable Artificial Intelligence*, Springer, pp. 350–373.

Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., Finn, C. (2020). Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33, 5824–5836.

Zhang, Y., Yang, Q. (2021). A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12), 5586–5609.

Y.-C. Chung received his PhD degree in the Department of Computer Science and Information Engineering at the National Cheng-Kung University, Taiwan, in 2007. Currently, he is an associate professor of the Department of Industrial Engineering and Management, National Kaohsiung University of Science and Technology, Taiwan. His research interests include cloud computing, sensor networks, query processing, spatio-temporal databases, and natural language processing.

M. Espinoa received her master degree in Bataan Peninsula State University, Philippines. She is studying her PhD degree in the Department of Industrial Engineering and Management at National Kaohsiung University of Science and Technology, Taiwan.

I-F. Su received her PhD degree in computer science in the Department of Computer Science and Information Engineering at the National Cheng-Kung University in 2010. Currently, she is an associate professor in the Department of Computer Science and Information Engineering at the National University of Tainan, Taiwan. Her research interests include databases, mobile computing, as well as data applications in deep learning and machine learning.