big data and cognitive computing

MDPI

Article

Leveraging Mixture of Experts and Deep Learning-Based Data Rebalancing to Improve Credit Fraud Detection

Zeyuan Yang [1], Yixuan Wang [2], Haokun Shi [3] and Qiang Qiu [1,*]

[1] College of Economics and Management, Nanjing Forestry University, Nanjing 210037, China; yangzeyuan@njfu.edu.cn
[2] Department of Computer Science, New York University, New York, NY 10012, USA
[3] School of Computer Science, University of Sheffield, Sheffield S1 4DP, UK
[*] Correspondence: qiuqiang@njfu.edu.cn

Abstract: Credit card fraud detection is a critical challenge in the financial sector due to the rapidly evolving tactics of fraudsters and the significant class imbalance betweenegitimate and fraudulent transactions. Traditional models, while effective to some extent, often suffer from high false positive rates and fail to generalize well to emerging fraud patterns. In this paper, we propose a novel approach that integrates a Mixture of Experts (MoE) model with a Deep Neural Network-based Synthetic Minority Over-sampling Technique (DNN-SMOTE) to enhance fraud detection performance. The MoE modeleverages multiple specialized expert networks, each trained to detect specific types of fraud, while the DNN-SMOTE generates high-quality synthetic samples to address the class imbalance. Our experimental results on a publicly available dataset demonstrate that the proposed method achieves a classification accuracy of 99.93%, a true positive rate of 84.69%, and a true negative rate of 99.95%. The Matthews Correlation Coefficient (MCC) of 0.7883 further highlights the model's balanced performance in detecting fraudulent transactions. These results underscore the effectiveness of combining MoE with DNN-SMOTE, offering a robust solution for real-world credit card fraud detection scenarios.

Keywords: credit card fraud detection; financial security; mixture of experts; ensembleearning; synthetic data generation

check for updates

Citation: Yang, Z.; Wang, Y.; Shi, H.; Qiu, Q. Leveraging Mixture of Experts and Deep Learning-Based Data Rebalancing to Improve Credit Fraud Detection. *Big Data Cogn. Comput.* **2024**, *8*, 151. https://doi.org/10.3390/bdcc8110151

Academic Editor: Enrico Corradini, Francesco Cauteruccio and Luca Virgili

Received: 13 August 2024
Revised: 19 October 2024
Accepted: 29 October 2024
Published: 5 November 2024

1. Introduction

In the rapidly evolving digital era, credit card fraud has emerged as a formidable challenge, posing significant threats to the financial security of individuals and institutions [1,2]. This is because the prevalence of credit card fraud has escalated with the advent of online banking and e-commerce, making it a critical concern for the financial industry [3]. The increasing reliance on electronic transactions has been paralleled by a surge in fraudulent activities. According to recent studies [4–6], globalosses due to credit card fraud have been increasing exponentially. These fraudulent activities not onlyead to substantial financialosses but also erode consumer trust in digital financial transactions [7]. For example, according to the Consumer Sentinel Network Data Book [8] from the Federal Trade Commission (FTC), in 2023 alone, consumers reportedosses exceeding USD 10 billion due to fraud, marking a significant 14% increase from the previous year. This figure represents the highest reported fraudosses on record. Investment scams were particularly devastating, accounting for more than USD 4.6 billion of the totalosses, while imposter scams contributed nearly USD 2.7 billion.

Therefore, credit card fraud detection has become an increasingly critical issue in the financial industry as the proliferation of online transactions hased to a corresponding rise in fraudulent activities. However, there are two major challenges for credit card fraud detection tasks:

1. First, the rapid evolution of fraud patterns, driven by the continuous adaptation and sophistication of fraudsters, presents a significant challenge to existing detection systems. Traditional fraud detection methods, primarily based on supervisedearning techniques, rely on historical data to identify fraudulent behavior. While these methods can be effective, they are oftenimited by their inability to detect new or emerging fraud patterns that do not conform to previously observed behaviors.
2. The second challenge is the dataset imbalance issue. The credit card transactions, coupled with the imbalanced nature of fraud datasets, where fraudulent transactions constitute a tiny fraction of the total, further complicates the detection process. This imbalance ofteneads to a high rate of false positives, whereegitimate transactions are incorrectly flagged as fraudulent, causing inconvenience to customers and potential financialoss to merchants. Moreover, the constantly changing tactics of fraudsters require detection systems to adapt quickly, a demand that traditional methods struggle to meet.

In response to these challenges, recent research has explored various advanced machineearning techniques to improve the accuracy and robustness of fraud detection models. These methods include autoencoders [9–12], ensemble methods [9,13,14], and hybrid models combining these approaches. While these methods have demonstrated improved accuracy and robustness, they often struggle with significant class imbalance and the dynamic nature of fraud patterns. For instance, traditional models tend to overfit the majority class or fail to generalize well to new, unseen fraud cases.

Recently, Mixture of Experts (MoE) models [15,16] have emerged as a promising approach and have been employed extensively across different fields such as naturalanguage processing, computer vision, and robotics. In naturalanguage processing, Mixture of Experts (MoE) models have proven effective in addressing tasks such as machine translation and speech recognition. These models excel because they can accommodate the diverseinguistic features and subtleties inherent inanguage, which are often better processed by specialized sub-models rather than a single, all-encompassing model. Similarly, in computer vision, MoE models have been employed in image recognition endeavors. Here, distinct experts within the MoE framework can focus on identifying different categories of objects, each with its unique set of characteristics, thereby enhancing the overall accuracy and efficiency of the recognition process. MoE models utilize a collection of specialized sub-models, referred to as "experts", each of which is trained to address distinct facets of the fraud detection challenge. By integrating the predictions generated by these experts, the MoE model is capable of delivering more precise forecasts, especially in intricate and volatile settings where fraudulent behaviors are not straightforward to discern. This approach allows the model to capture a broader range of fraud patterns and adapt to the ever-changing nature of fraudulent activities.

Traditional oversampling techniques, such as the synthetic minority over-sampling technique (SMOTE) [17], have proven effective in classical machineearning models but struggle to integrate seamlessly with deepearning architectures, especially when working with complex, high-dimensional data. The recent DeepSMOTE [18] addresses theseimitations by combining the strengths of SMOTE with a convolutional architecture. Inspired by this, we propose an encoder–decoder to handle the challenge of data imbalance for credit card fraud detection. Our proposed DNN-SMOTE allows for the generation of high-quality synthetic instances that enhance the minority class representation. This approach not only balances the dataset but also enhances the model's ability to discriminate between classes.

In this work, we propose a novel framework consisting of MoE and DNN-SMOTE to address theseimitations. The integration of MoE with DNN-SMOTE offers a promising solution. The MoE model, with its ability toeverage multiple specialized expert models, enhances the model's adaptability and precision in detecting complex fraud patterns. Meanwhile, DNN-SMOTE effectively mitigates the class imbalance by generating high-quality synthetic samples for the minority class, ensuring that the model is trained on a more representative dataset. This combination allows for a more nuanced and accurate detection system, capable of handling the challenges that have hindered previous approaches.

2. Related Works

2.1. Traditional Fraud Detection Methods

Traditional approaches to fraud detection were predominantly rule-based, relying on fixed parameters and thresholds [19,20]. These systems, effective in identifying overt anomalies, often struggled with the subtleties and complexities of sophisticated fraud tactics [21]. As digital transactions advanced, these methods began showingimitations, primarily in scalability and adaptability [22]. The introduction of decision trees and basic statistical models, such asogistic regression, marked an evolution, offering more nuanced detection capabilities [23–25]. However, these methods still faced challenges in handling the dynamic nature of fraud, often resulting in high false-positive rates and the inability to adapt quickly to new fraud patterns [26]. Recent developments have seen the integration of hybrid models that combine rule-based systems with early machineearning techniques, aiming to improve accuracy and reduce false positives [27]. Nevertheless, the inherentimitations of traditional fraud detection frameworks became increasingly apparent. They were often unable to keep pace with the rapid evolution of fraud strategies, as they relied on historical trends that could become quickly outdated. The manual creation and updating of rules presented a significant operational burden, making it difficult to respond to fraud in real-time. Additionally, these systems did not account for the complex and evolving patterns ofegitimate user behavior,eading to a high rate of false alarms, which could alienate customers and strain resources. This recognition ofimitations prompted a shift towards more advanced analytics and machineearning-based approaches, setting the stage for a new era in fraud detection methodologies.

2.2. Deep Learning and Ensemble Methods

Recent advancements in credit card fraud detection haveargely focused on the development of machineearning and deepearning techniques to improve accuracy and resilience against evolving fraud patterns.

Deepearning approaches have also gained traction, with modelsike autoencoders [10] combined with Restricted Boltzmann Machines (RBMs) [12] being used to detect anomalies in transaction data. These workseveraged the unsupervisedearning capabilities of autoencoders to identify suspicious transactions, offering a solution to the challenge of evolving fraud patterns without relying heavily onabeled datasets. The integration of variational autoencoders with generative adversarial networks (VAE-GANs) has been introduced to generate synthetic data [11,28], thereby improving model training and addressing class imbalance. This approach enhances the representation of the minority class,eading to more effective fraud detection.

Additionally, enhanced autoencoder-based models combined with SMOTE [10,29] have shown promise in reducing false positives while improving the detection of fraudulent transactions. This hybrid approach not only mitigates data imbalance but also enhances overall detection performance. Ensemble methods [9,13,14] have proven effective by combining multiple classifiers to enhance detection rates. The integration of ensemble methods with deepearning models, as seen in hybrid systems, continues to push the boundaries of fraud detection. These systemseverage the strengths of both approaches, offering superior performance in identifying complex fraud patterns compared to single-model techniques.

2.3. Utilizing Data Augmentation Techniques

Data augmentation has become increasingly vital, especially in scenarios whereabeled data are scarce [30,31]. This approach marks a departure from the reliance on extensive-abeled datasets common in traditional machineearning paradigms [32,33]. By artificially enhancing data through various augmentation techniques, such as introducing controlled noise [34], rotation [35], or distortion, data augmentation helps unveil hidden patterns and anomalies within the dataset [36,37]. This method has shown significant potential in diverse areasike image processing, naturalanguage processing, and cybersecurity, enabling the detection of anomalies in complex and evolving datasets [38]. The flexibility and effectiveness of data augmentation make it a promising tool for addressing the intricate and dynamic nature of credit card fraud detection. Moreover, data augmentation techniques have been instrumental in mitigating the problem of overfitting in machineearning models, particularly those that are prone toearning noise in the training data. By expanding the diversity of the training examples, models areessikely to memorize specific instances and insteadearn to generalize better to unseen data. This results in improved model robustness and reliability, essential qualities for systems deployed in critical sectors such as finance. As theandscape of fraud continues to evolve, data augmentation stands as a critical component in the arsenal of tools available to data scientists and fraud analysts, ensuring that machineearning models remain effective in the ongoing battle against financial fraud.

2.4. Limitations of Existing Works

Traditional oversampling techniques have been widely used to address class imbalance by generating synthetic samples for the minority class [36–38]. However, these methods often fall short when applied to high-dimensional or complex datasets, as they are unable to capture the non-linear relationships and feature interactions in real-world data. SMOTE [17], while effective in classical machineearning, can also suffer from the issue of generating synthetic samples that are overly simplistic and prone to overlapping with outliers, which introduces noise into the training process. This not only degrades model performance but also results in overfitting to the majority class in highly imbalanced datasets. To overcome theseimitations, we proposed the DNN-SMOTE model thateverages an encoder–decoder structure to learn a more nuanced representation of the minority class. By generating synthetic samples from aatent spaceearned by the network, DNN-SMOTE produces more diverse and realistic data points that better capture the complexity of the minority class, ultimatelyeading to improved generalization and model robustness.

On the other hand, existing fraud detection models and classifiers [9,10,13,14,29] often struggle with the challenge of capturing diverse patterns in highly imbalanced datasets, where fraudulent activities exhibit wide variability. Traditional classifiers such as decision trees, random forests, and even deep neural networks treat the entire dataset uniformly, which canead to poor performance when faced with new, previously unseen fraud patterns. These models areimited by their inability to specialize in different regions of the data space, often resulting in high false positives or missed fraud detections. In this work, we exploit the Mixture of Experts (MoE) model, which addresses theseimitations by using multiple specialized expert networks, each trained to capture distinct aspects of the data. Its hierarchical structure allows the MoE classifier to balance precision and recall more effectively, reducing false positives while improving detection rates for minority class instances such as fraudulent transactions.

3. Proposed MoE with DNN-SMOTE Model for Fraud Detection with Class Imbalance

The core aim of this paper is to precisely identify and categorize credit card fraud events using the proposed algorithm. Figure 1 illustrates the overall framework of the proposed Mixture of Experts (MoE) model integrated with the DNN-SMOTE oversampling method for credit card fraud detection. This framework effectively addresses the challenges posed by imbalanced datasets in fraud detection, combining the strengths of DNN-SMOTE and MoE to create a more accurate and reliable detection model.

Figure 1 is divided into three phases to handle a dataset with a class imbalance during training and inference:

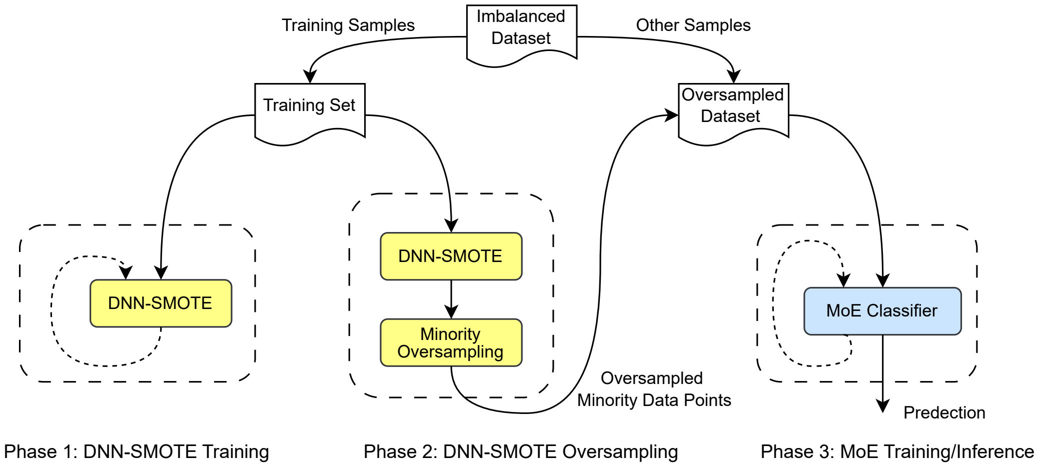


Figure 1. The overall framework and algorithmic flow of the proposed MoE model with the DNN-SMOTE oversampling method.

- **Phase 1: DNN-SMOTE Training:** In this phase, the DNN-SMOTE module is trained using the training set derived from the imbalanced dataset. The DNN-SMOTE module is responsible forearning the characteristics of the minority class (in this case, fraudulent transactions) and generating high-quality synthetic samples for better representation of the minority class during training.
- **Phase 2: DNN-SMOTE Oversampling:** Once the DNN-SMOTE module is trained, it is used to perform minority oversampling on the imbalanced dataset. This process generates oversampled minority data points, which results in a more balanced training dataset. The oversampled dataset combines both real and synthetic data, ensuring the minority class (fraudulent transactions) is better represented.
- **Phase 3: MoE Training/Inference:** In this phase, the MoE (Mixture of Experts) classifier is trained using the newly oversampled dataset. During inference, the MoE classifier makes predictions using the trained expert networks, with the goal of accurately classifying both majority and minority class instances, particularly in the presence of imbalanced datasets.

3.1. Mixture of Experts (MoE) Model for Credit Fraud Detection

Figure 2 illustrates the architecture of the proposed MoE model for credit card fraud detection. The MoE model is an ensembleearning technique that combines the predictions of multiple expert models, each specialized in different aspects of the input space. The key components of this architecture are the expert models, the gating network, and the weighted combination of expert outputs to produce the final prediction.

This MoE framework is particularly effective for handling complex and high-dimensional data, such as in credit card fraud detection, where different experts can focus on capturing different types of fraud patterns. The flexibility and adaptability of the MoE model allow it to provide more accurate and reliable predictions by dynamically weighting the contribution of each expert based on the specific characteristics of the input data.

3.1.1. Expert Networks

The MoE model in Figure 2 consists of N expert models, denoted as Expert 1, Expert 2, $\ldots$, Expert N. Each expert model $E_i(x)$ receives the same input x but generates a different output. Each expert in the MoE model is a neural network trained on the training data. The experts are designed to become specialists, each excelling in a specific region of the input space. Mathematically, the output of each expert i for an input $\mathbf{x}$ can be represented as:

$$E_i(\mathbf{x}) = f_i(\mathbf{x}) = \sigma(\theta_i^T \mathbf{x} + b), \tag{1}$$

where f_i is the function modeled by the i-th expert, and θ_i denotes the parameters of the expert. Here, we use ainearayer with non-linear activation $\sigma(\cdot)$ as the expert function.

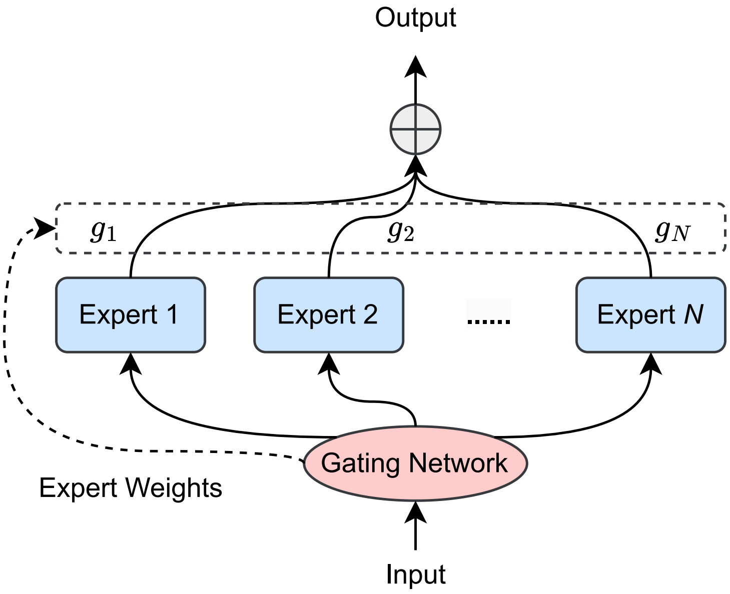


Figure 2. Diagram of the MoE model.

3.1.2. Gating Network

The gating network determines the weight or contribution of each expert for a given input. For a given input x, the gating network computes a set of weights $g_1(x), g_2(x), \ldots, g_N(x)$, where each $g_i(x)$ represents the weight assigned to the output of the corresponding expert $E_i(x)$. These weights are typically determined by a Softmax function, ensuring that the sum of all weights equals one:

$$g_i(\mathbf{x}) = \frac{\exp(w_i^\top \mathbf{x} + b_i)}{\sum_{j=1}^{N} \exp(w_j^\top \mathbf{x} + b_j)} \tag{2}$$

where w_i and b_i are the weights and bias for the i-th expert within the gating network, and N is the total number of experts.

3.1.3. Weighted Model Output

The final output of the MoE model is a weighted sum of the outputs from all expert networks, where the weights are provided by the gating network. For a given input $\mathbf{x}$, the model output Y is computed as:

$$Y(\mathbf{x}) = \sum_{i=1}^{N} g_i(\mathbf{x}) \cdot E_i(\mathbf{x}) \tag{3}$$

Equation (3) allows the MoE model to adaptively choose which experts to rely on based on the input, making it highly flexible and capable of handling diverse and complex data distributions effectively. The weighted combination allows the MoE model toeverage the strengths of each expert, thereby improving the overall model performance.

3.1.4. Weighted Cross Entropy

Using a standard binary cross-entropyoss function may cause the model to be biased towards the majority class. To address the issue of class imbalance in binary classification tasks, we use the weighted binary cross-entropyoss, which is a variation in the standard binary cross-entropyoss as follows:

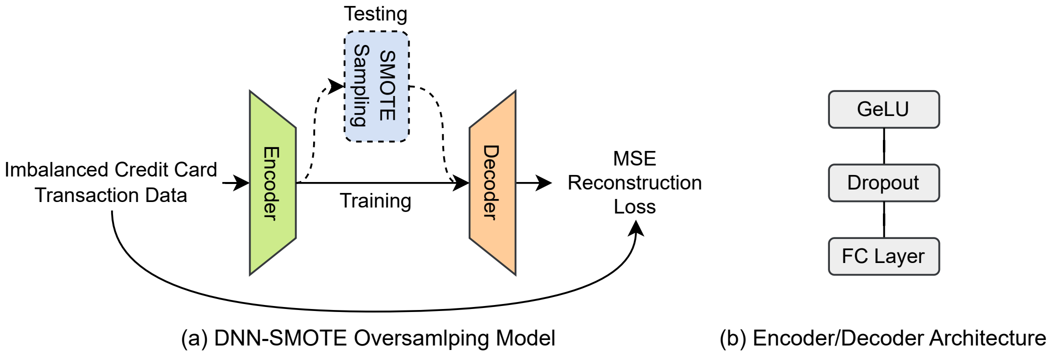
$$\text{Weighted BCE}(y, \hat{y}) = -[w_1 \cdot y \cdot \log(\hat{y}) + w_0 \cdot (1 - y) \cdot \log(1 - \hat{y})], \tag{4}$$

where weights w_0 and w_1 are introduced for the negative class (label 0) and positive class (label 1), respectively. The weighted binary cross-entropyoss assigns different weights to the positive and negative classes to ensure that the model does not become biased towards the majority class, thus maintaining a better balance between precision and recall.

3.2. Oversampling with DNN-SMOTE

Data augmentation in credit card fraud detection plays a crucial role in addressing the challenge ofimitedabeled anomaly data. SMOTE [17] has been widely adopted in previous works [9–11] where the imbalance in class distribution is prevalent. Its capability to enhance the representativeness of the minority class helps in building more robust classifiers. While SMOTE is effective in creating a balanced dataset, it may also introduce noise, especially when the minority class samples are outliers or when the feature spaceacks coherence. Hence, it is crucial to apply SMOTE in conjunction with appropriate outlier detection and feature selection techniques to maximize its effectiveness.

By introducing artificial variations that mimic fraudulent behavior, we can enhance the diversity of the training dataset. In this work, we propose the DNN-SMOTE model, whicheverages deepearning techniques to enhance the traditional SMOTE approach [17], generating high-quality synthetic samples that improve the performance of fraud detection models on imbalanced datasets. Figure 3 illustrates the proposed DNN-SMOTE model designed for data oversampling in the context of credit card fraud detection. This model addresses the issue of imbalanced datasets, where fraudulent transactions are significantlyess frequent than non-fraudulent ones. The figure is divided into two parts: (a) the DNN-SMOTE oversampling model and (b) the GeLU-based encoder/decoder architecture.



(a) DNN-SMOTE Oversamlping Model

(b) Encoder/Decoder Architecture

Figure 3. (**a**) Diagram of proposed DNN-SMOTE oversampling method. (**b**) GeLU-based encoder/decoder architecture.

The DNN-SMOTE model integrates deepearning with the SMOTE to generate synthetic samples for the minority class (fraudulent transactions). The process begins with the imbalanced credit card transaction data being fed into an encoder, which compresses the data into aower-dimensional representation. The encoder encodes the input data x into aatent space representation z:

$$z = f_{\text{encoder}}(x) = \text{GeLU}(\text{FC}(x)) \tag{5}$$

This compressed representation is then passed to a decoder, which reconstructs the data. The decoder reconstructs the data from theatent space representation:

$$\hat{x} = f_{\text{decoder}}(z) = \text{GeLU}(\text{FC}(z)) \tag{6}$$

The encoder and decoder are both built using GeLU (Gaussian Error Linear Unit) activation functions, dropoutayers, and fully connected (FC)ayers. The GeLU activation function is particularly effective in deepearning models as it allows for smoother and more effectiveearning, compared to traditional ReLU activations. The fully connectedayers are responsible forearning the complex patterns and features in the data that are essential for the effective reconstruction and synthesis of minority class samples. Dropoutayers are included to prevent overfitting by randomly disabling certain neurons during training, thereby improving the generalization of the model.

The Mean Squared Error (MSE) reconstructionoss is used to evaluate the difference between the original data and the reconstructed data, guiding the training process to minimize thisoss.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 \tag{7}$$

This oss function ensures that the reconstructed data closely matches the original input, thus preserving the essential characteristics of the minority class during oversampling.

During testing, the trained model applies SMOTE to the encoded representations to generate synthetic samples, effectively balancing the dataset. The SMOTE for the encoded representations involves the following steps to generate synthetic data:

1. **Selecting the *k*-nearest neighbors:** For each instance in the minority class, identify the *k*-nearest neighbors in encoded space for the minority class using the Euclidean distance metric.
2. **Synthetic sample generation by interpolation:** For each minority class sample, the synthetic samples are generated by choosing one of the *k*-nearest neighbors and interpolating between the two points. The synthetic samples in the encoded representation are created as follows:

   $$\mathbf{z}_{new} = \mathbf{z} + (\mathbf{z}_{nn} - \mathbf{z}) \times \delta \tag{8}$$

   where $\mathbf{z}$ is a vector representing the encoded vectors of minority data points, $\mathbf{z}_{nn}$ is one of its nearest neighbors, and δ is a random number between 0 and 1. This interpolation approach ensures that the synthetic samples are a variation within the feature space between the existing minority instances, thereby contributing to a more general and diversified representation of the minority class.
3. **Decoding:** The generated encoded samples $\mathbf{z}_{new}$ are then decoded back into the original data space and combined with the original dataset, providing a more balanced and representative set of training data for the classifier.

4. Experiments

This section delves into a comprehensive discussion and analysis of the following aspects: key hyperparameter selection and performance comparison of various credit card fraud detection models using a publicly available dataset. The goal is to assess the effectiveness of these models in identifying fraudulent transactions from an imbalanced dataset, where fraudulent transactions are significantlyess frequent than non-fraudulent ones.

4.1. Experimental Environment

The experiments were conducted using a computing environment equipped with NVIDIA GeForce RTX 4060 Ti with Intel Xeon Platinum 8276 CPU to accelerate model training. To implement the proposed DNN-SMOTE algorithm, the deepearning frameworks used for this research included PyTorch v2.1.0, which enabled efficient implementation of neural networks and flexible model configurations. We used scikit-learn for the other traditional machineearning baselines and preprocessing tasks such as data scaling and model evaluation.

4.2. Dataset Introduction and Preprocessing

The dataset utilized in this study is accessible on Kaggle [39] and was contributed by the Machine Learning Group at Université Libre de Bruxelles (ULB). The dataset consists of 30 features. It encompasses transactions conducted by European credit card holders within a two-day period in September 2013, totaling 284,807 transactions. Notably, only 492 of these transactions are identified as fraudulent, representing a mere 0.172% of the total dataset. This significant class imbalance renders the dataset particularly suitable for evaluating the efficacy of sophisticated machineearning approaches, including the MoE model enhanced with DNN-SMOTE, in identifying infrequent fraudulent activities.

The raw data are first preprocessed to handle missing values, normalize the features, and encode categorical variables if necessary. Due to confidentiality issues, all feature names except for "Time" and "Amount" have been anonymized. Feature scaling is applied to standardize the range of independent variables or features of data. This step is crucial for models sensitive to the scale of data. The dataset is split into training, validation, and testing subsets, following a distribution of 80%, 10%, and 10%, respectively. This partitioning ensures a comprehensive approach to model training and performance evaluation.

4.3. Model Configuration

The encoder and decoder in the DNN-SMOTE oversampling method are both 3-layer models. The k value for kNN, SMOTE and DNN-SMOTE is $k = 5$. The hidden dimension of each expert network is 64. The training for DNN-SMOTE and MoE models was conducted over 300 epochs using the Adam optimizer with a weight decay of 0.02 and an initialearning rate of 1×10^{-4}. A cosine decayearning rate scheduler, coupled with ainear warm-up phase spanning the first 10 epochs, was employed to facilitate faster convergence.

Table 1 provides a summary of the evaluated algorithms used in this paper for credit card fraud detection. The table categorizes the algorithms into two main groups: conventional machineearning and deepearning or ensemble methods. Including both conventional and advanced algorithms allows for a comprehensive evaluation of different approaches, providing insights into the strengths and weaknesses of each method. The first category includes widely-used conventional machineearning algorithms such asogistic regression, random forest, AdaBoost, bagging, gradient boosting, and k-Nearest Neighbors (kNN). These models are often employed in various classification tasks due to their simplicity, interpretability, and relatively fast training times. The second category consists of more advanced deepearning and ensemble-based methods, which are designed to handle the intricacies of complex data distributions. This group includes models such as Autoencoder (AE), Support Vector Machine (SVM) combined with AdaBoost, Autoencoder with Probabilistic Random Forest (PRF), and Autoencoder combined with LightGBM (AE-LGB) using SMOTE for oversampling. This category also includes the proposed models in this research: MoE combined with SMOTE and MoE combined with DNN-SMOTE.

Table 1. Summary of evaluated algorithms for credit card fraud detection.

Category	Algorithm
Conventional Machine Learning	Logistic Regression Random Forest AdaBoost Bagging Gradient Boosting kNN [40]
Deep Learning or Ensemble	AE [12] SVM with AdaBoost [13] AE with PRF [9] AE-LGB with SMOTE [10] MoE with SMOTE (This Work) MoE with DNN-SMOTE (This Work)

4.4. Evaluation Metrics

The effectiveness of the credit card fraud detection algorithms is assessed using a test dataset comprising unseen transactions. Several key metrics are employed to assess the model performance. These metrics include accuracy, the true positive rate, the true negative rate, the false positive rate, the confusion matrix, the Matthews correlation coefficient (MCC) [41], and the receiver operating characteristic (ROC) curve [42]. These metrics enable a comprehensive evaluation of its effectiveness in identifying fraudulent activities.

Accuracy (ACC) measures the proportion of correctly classified instances, both fraudulent andegitimate, out of the total instances. The accuracy is calculated as:

$$\text{ACC} = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

where TP (True Positives) represents the number of correctly identified fraudulent transactions, TN (True Negatives) represents the number of correctly identifiedegitimate transactions, FP (False Positives) denotesegitimate transactions incorrectly classified as fraudulent, and FN (False Negatives) denotes fraudulent transactions incorrectly classified as legitimate. True Positive Rate (TPR), also known as sensitivity or recall, measures the proportion of actual fraudulent transactions that are correctly identified by the model. It reflects the model's ability to detect fraud when it is present. The TPR is calculated as follows:

$$\text{TPR} = \frac{TP}{TP + FN} \tag{10}$$

True Negative Rate (TNR), or specificity, quantifies the proportion of actualegitimate transactions that are correctly identified as such by the model. It complements the TPR by focusing on the model's performance in recognizing non-fraudulent activities. The TNR is given by:

$$\text{TNR} = \frac{TN}{TN + FP} \tag{11}$$

False Positive Rate (FPR) is the proportion ofegitimate transactions that are incorrectly classified as fraudulent. A high FPR canead to unnecessary alerts and customer dissatisfaction. The FPR is calculated as:

$$\text{FPR} = \frac{FP}{FP + TN} \tag{12}$$

The Matthews Correlation Coefficient (MCC) [41] is a more comprehensive metric that takes into account the true and false positives and negatives, providing a balanced measure of the model's performance even when the classes are imbalanced. The MCC value ranges from -1 to +1, where +1 indicates perfect prediction, while -1 indicates total disagreement between prediction and observation. The MCC is calculated as:

$$\text{MCC} = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{13}$$

The confusion matrix is a table that summarizes the performance of a classification algorithm by displaying the counts of true positives, true negatives, false positives, and false negatives. It provides a detailed breakdown of how well the model is performing and helps in calculating other metrics. The confusion matrix is typically structured as follows:

	Predicted Negative	Predicted Positive
Actual Negative	FP	TN
Actual Positive	TP	FN

A Receiver Operating Characteristic (ROC) curve [42] is considered to provide a graphical representation that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The ROC curve is plotted by comparing the TPR against the FPR at various threshold settings.

These metrics collectively provide a comprehensive view of the model's performance, ensuring that both the detection of fraud and the minimization of false positives are adequately balanced in the evaluation of the fraud detection model.

4.5. Evaluation for MoE Models and Oversampling Algorithms

Figure 4 presents three plots showing the performance of the DNN-SMOTE module and the MoE classification module across 300 training epochs. In Figure 4a, the trainingoss initially starts higher above 20, and sharply decreases in the first 100 epochs. After the initial drop, the loss continues to decrease gradually, eventually flattening out, suggesting that the model converges as training progresses. The test accuracy plot demonstrates strong overall performance, with average accuracy nearing 1.0 and balanced accuracy remaining high, indicating the model's robustness in the presence of class imbalance, thanks to the DNN-SMOTE module. The smooth convergence in bothoss plots and the stabilization in accuracy suggest that the models are well-optimized and trained effectively. Figure 4b represents the trainingoss of the MoE classification module. The loss starts around 0.35 and shows a steady, significant decrease over the first 100 epochs. After 300 training epochs, the loss plateaus at approximately 0.175, indicating convergence similar to the DNN-SMOTE module. Figure 4c shows the test accuracy of the MoE classification module, where two metrics are plotted: the overall accuracy across all classes (average accuracy), and balanced accuracy that accounts for class imbalance. Balanced accuracy (orangeine) improves sharply but stabilizes at around 0.95, showing good performance in accounting for both classes in the imbalanced dataset. The slight gap between the two accuracy metrics indicates the effectiveness of the proposed DNN-SMOTE and MoE models in handling class imbalance.

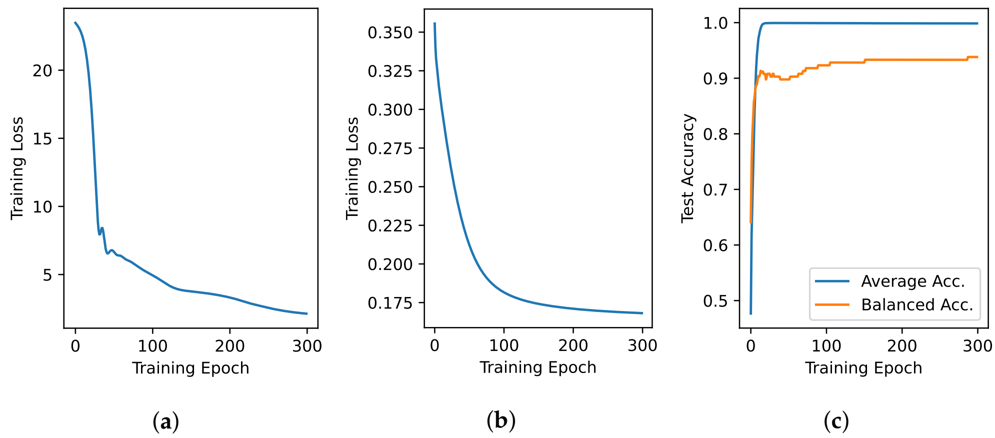


Figure 4. Training oss and testing accuracy for the DNN-SMOTE module and MoE classification module. (**a**) DNN-SMOTE training. (**b**) MoE training loss. (**c**) MoE testing accuracy.

In Figures 5 and 6, we first compare the impact of different oversampling techniques, SMOTE and DNN-SMOTE, on the distribution of training data points in the context of credit card fraud detection. We used t-Distributed Stochastic Neighbor Embedding (t-SNE) [43] to visualize high-dimensional data. The technique first reduces the data to two dimensions, enabling easier exploration and interpretation. Figures 5 and 6 depict how the data points are distributed across two dimensions under various oversampling ratios: 0.1, 0.3, 0.5, and 0.7. The oversampling ratio is defined as the ratio of the minority class (fraudulent transactions) with respect to the majority class (non-fraudulent transactions). The blue points represent the majority class, and the orange points represent the minority class.

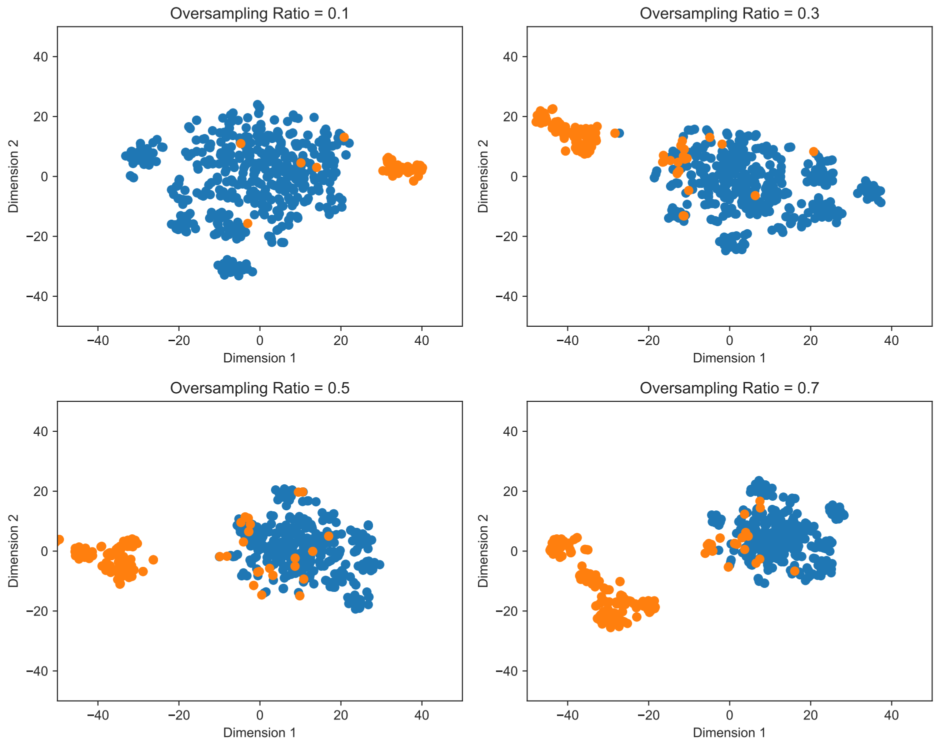


Figure 5. Visualization of sampled training data points for fraud detection under various oversampling ratios. The oversampling method is SMOTE. Blue: majority class. Orange: minority class.

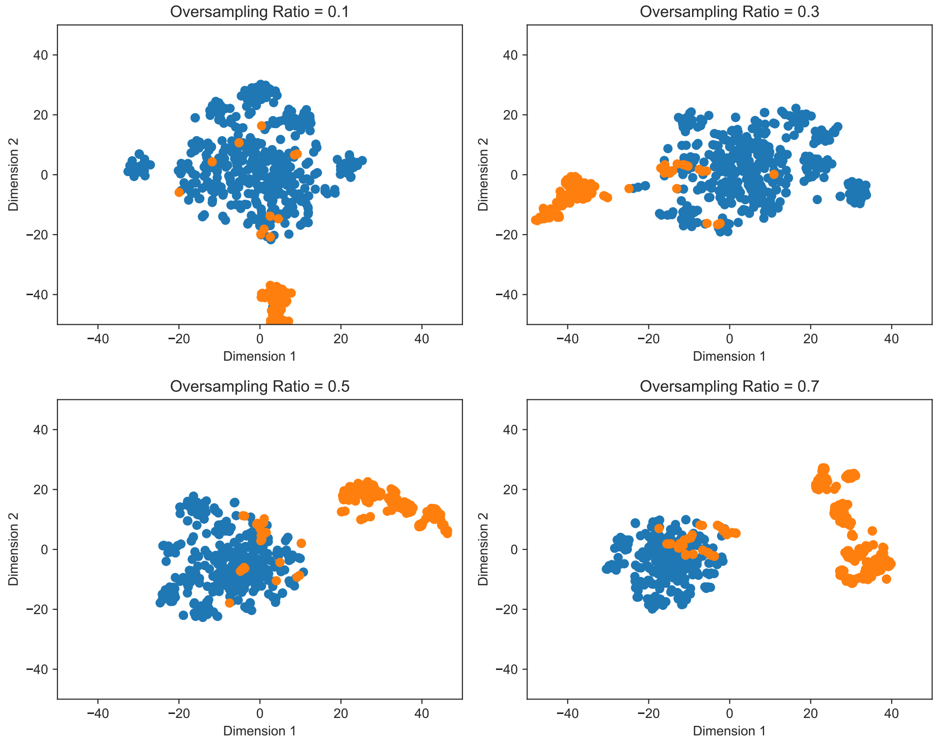
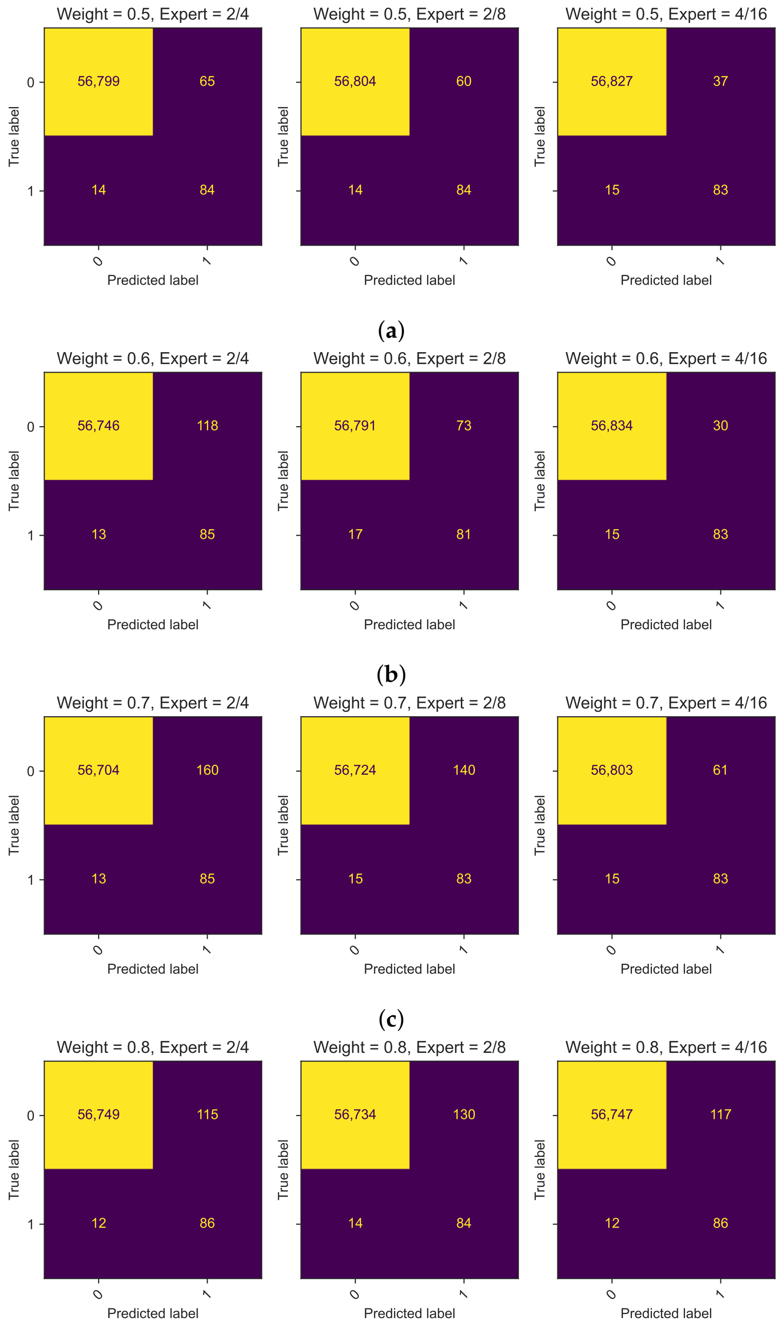


Figure 6. Visualization of sampled training data points for fraud detection under various oversampling ratios. The oversampling method is the proposed method DNN-SMOTE. Blue: majority class. Orange: minority class.

SMOTE generates synthetic samples by performinginear interpolation between existing minority class samples in the feature space. SMOTE works reasonably well for simple datasets. However, it hasimitations in high-dimensional or complex data. A common issue with SMOTE is: if there are outliers in the minority class, SMOTE may generate synthetic samples close to these outliers, thereby introducing noise into the training data. In comparison, DNN-SMOTEeverages an encoder–decoder architecture toearn aatent representation of the data. This network compresses the minority class data into aower-dimensional space using the encoder and then reconstructs it using the decoder. DNN-SMOTE mitigates this issue byearning aatent space representation of the minority data, which helps the model smooth out outliers.

In Figure 5 with SMOTE oversampling, as the oversampling ratio increases, the minority data points become more spread out and integrated with the majority class. At lower oversampling ratios, the minority data points are relatively sparse and mostly clustered near the majority class points, indicatingimited oversampling. As the ratio increases to 0.3 and 0.5, the orange points representing the minority class begin to fill in the space, but they remain somewhat close to the majority class, suggesting that SMOTE is interpolating between existing minority class samples. In Figure 6 with the proposed DNN-SMOTE method, at theower oversampling ratios, the minority data points are dispersed compared to SMOTE, indicating that DNN-SMOTE is generating more diverse synthetic samples even with minimal oversampling. As the ratio increases to 0.3 and 0.5, the distribution of the minority data points becomes more varied andess clustered than in the SMOTE figure, suggesting that DNN-SMOTE is better at creating synthetic samples that span a broader region of the feature space. DNN-SMOTE provides a more diversified and better-distributed set of minority data samples across different oversampling ratios. This improved distributionikely contributes to better generalization in the model, as it allows the classifier toearn more varied examples of fraudulent transactions, reducing the risk of overfitting to a narrow set of minority class features.

We first study the impact of the MoE model and training configurations on the classification performance. We plot a series of confusion matrices in Figure 7 that depict the performance of the proposed MoE model using DNN-SMOTE with an oversampling ratio of 0.2 across various configurations ofoss weights and expert configurations (2/4, 2/8, 4/16). Each subfigure (a) through (d) represents differentoss weights (0.5 to 0.8), and within each subfigure, the performance is evaluated for different expert configurations. Withoweross weights (e.g., 0.5), the model maintains a relatively high number of true negatives and true positives, as seen by theower misclassification rates in both the positive and negative classes. However, as the weight increases, the number of false negatives and false positives begins to rise. This indicates a potential trade-off between precision and recall as the model is adjusted to focus more on one aspect, possiblyeading to increased misclassification in another area. The expert configuration also plays a significant role. For example, in subfigure (d) with weight 0.8, the confusion matrix with a 2/4 expert configuration shows relatively balanced performance, but as the expert configuration increases to 2/8 and 4/16, the number of false positives and false negatives rises, suggesting that higher complexity in the model mightead to overfitting or misclassification in this specific setup. This analysis suggests that while increasing model complexity and adjusting weights can potentially enhance certain aspects of performance, it also introduces risks of misclassification, particularly in a highly imbalanced context such as fraud detection. Therefore, fine-tuning these parameters is crucial for achieving a balanced and effective fraud detection model.



(**a**)

(**b**)

(**c**)

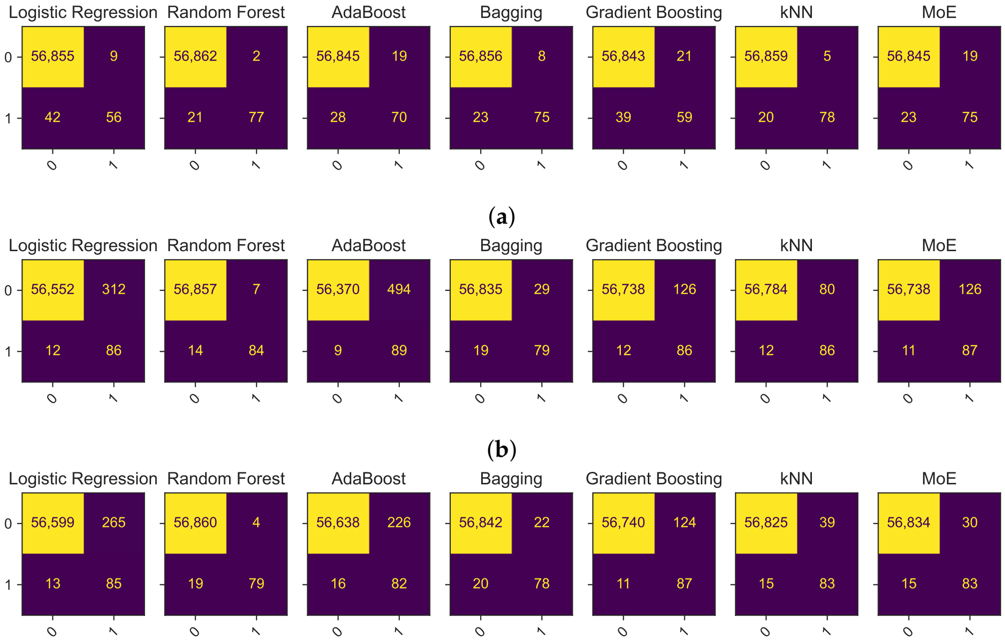
(**d**)

Figure 7. Confusion matrix for weighted cross-entropyoss with DNN-SMOTE oversampling ratio = 0.2.

4.6. Performance Evaluation

4.6.1. Performance Comparison with Other Machine Learning Models

Figure 8 presents confusion matrices comparing the performance of several machineearning algorithms (logistic regression, random forest, AdaBoost, bagging, gradient boosting, kNN, and MoE) across three scenarios: (a) no oversampling, (b) SMOTE oversampling, and (c) DNN-SMOTE oversampling. For the no oversampling scenario, random forest andogistic regression perform well, with random forest achieving the highest number of true positives and theowest number of false positives. AdaBoost and gradient boosting show a higher number of false positives, suggesting that they may overfit the majority class. When SMOTE oversampling is applied, the performance of most algorithms improves in terms of reducing false negatives. Notably, the AdaBoost model, which previously had high false negatives, now shows a marked improvement, although it still suffers from a relatively high number of false positives (494). Random forest continues to show balanced performance with aow false positive rate (7), but logistic regression and bagging show an increase in false positives compared to the no oversampling scenario. MoE balances well between false negatives and false positives. DNN-SMOTE oversampling further enhances the model's performance, with random forest and MoE showing a significant reduction in both false positives and false negatives. MoE, in particular, shows a substantial improvement with a more balanced confusion matrix, indicating that it effectively handles the minority class with DNN-SMOTE. The reduction in false positives across most models indicates that DNN-SMOTE provides better generalization and a more nuanced understanding of the minority class than SMOTE.



(**a**)

(**b**)

(**c**)

Figure 8. Confusion matrix of different algorithms. (**a**) No oversampling. (**b**) SMOTE oversampling. (**c**) DNN-SMOTE oversampling.

Figure 9 provides the ROC curve of various machineearning algorithms (logistic regression, random forest, AdaBoost, bagging, gradient boosting, kNN, and MoE) in detecting credit card fraud without oversampling techniques. Random forest, AdaBoost, and logistic regression demonstrate the best performance, with curves that closely hug the top-left corner of the plot, indicating high true positive rates andow false positive rates across various thresholds. On the other hand, kNN and MoE exhibit suboptimal performance, as indicated by their curves, which are closer to the diagonaline representing random guessing. The comparative analysis highlights that ensemble methodsike random forest and AdaBoost generally outperform other models in this imbalanced dataset scenario, achieving higher true positive rates while maintainingower false positive rates. The underperformance of kNN and MoE suggests that these models may require additional techniques, such as oversampling, to handle the imbalance in the data effectively.

Figure 10a shows the ROC curve with SMOTE oversampling. The ROC curves for all the algorithms show significant improvement, closely hugging the top-left corner of the plot. This indicates that SMOTE effectively addresses the class imbalance by enhancing the models' ability to correctly identify fraudulent transactions while minimizing false positives. The kNN and MoE models, which showedower performance without oversampling, exhibit marked improvement with SMOTE, as evidenced by their ROC curves moving closer to the ideal top-left region. This indicates that SMOTE has effectively helped these models better handle the minority class, improving their overall classification performance. The ROC curve analysis with SMOTE oversampling highlights that the application of this technique significantly enhances the performance of all the models.

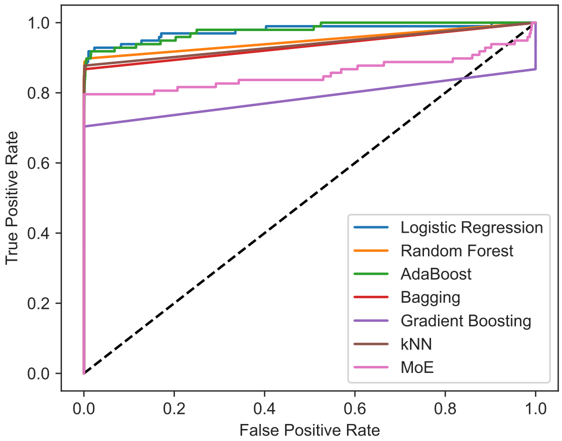
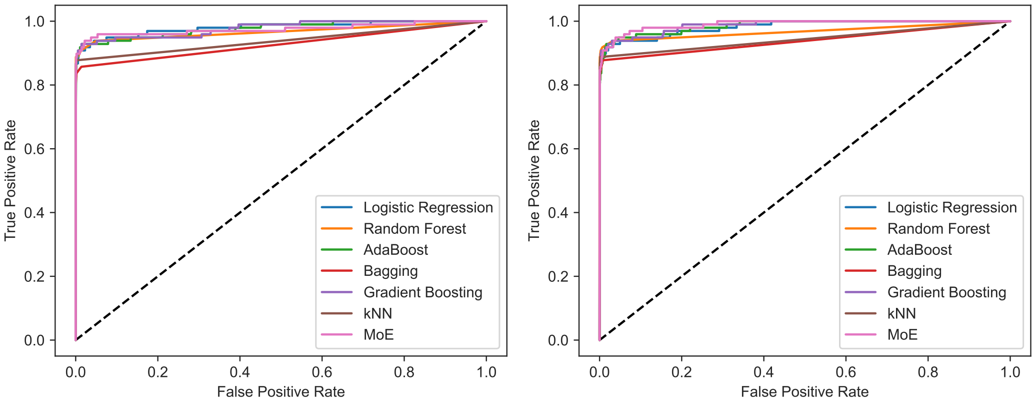


Figure 9. ROC curve without oversampling.



(**a**) SMOTE oversampling (**b**) DNN-SMOTE oversampling

Figure 10. ROC curve with (**a**) SMOTE and (**b**) DNN-SMOTE oversampling and sampling ratio = 0.2.

Figure 10b depicts the ROC curve with proposed DNN-SMOTE oversampling. The ROC curves for all algorithms show further improvement compared to standard SMOTE. The curves are tightly clustered near the top-left corner of the plot, indicating that all models achieve high true positive rates while maintainingow false positive rates. Notably, the MoE model, which had shown considerable improvement with SMOTE, now exhibits even better performance with DNN-SMOTE, placing it closer to the top-tier modelsike random forest and AdaBoost. This indicates that the advanced synthetic sampling provided by DNN-SMOTE helps MoE and other models generalize better and improve their ability to correctly classify both fraudulent and non-fraudulent transactions. Overall, this analysis suggests that DNN-SMOTE is a powerful tool for improving fraud detection models, offering superior performance and making previouslyess effective models more competitive in highly imbalanced datasets.

4.6.2. Performance Comparison with Existing Algorithms

We compare the proposed MoE-based model to various algorithms for credit card fraud detection, including kNN [40], AE [12], SVM with AdaBoost [13], AE with PRF [9], AE-LGB with SMOTE [10]. Table 2 provides an insightful comparison of these fraud detection algorithms, highlighting performance metrics such as ACC, TPR, TNR, and MCC. Our proposed MoE with DNN-SMOTE model achieved the highest accuracy of 0.9993, indicating an almost perfect classification. However, as mentioned earlier, it can be misleading in highly imbalanced datasets. TPR measures the ability of the model to correctly identify fraudulent transactions. The AE with PRF model achieved the highest TPR of 0.8910. Conversely, while the SVM with AdaBoost has an excellent TNR of 0.9995, its TPR isower at 0.8231, indicating that it might miss some fraudulent transactions despite its high overall accuracy. The MoE with DNN-SMOTE, developed in this work, strikes a balance with a TPR of 0.8469 and a TNR of 0.9995, suggesting it provides a more robust detection capability by minimizing false negatives while maintaining high specificity.

Table 2. Model performance comparison for different algorithms. **Bold** font indicates the best one.

Algorithm	ACC	TPR	TNR	MCC
kNN [40]	0.9691	0.8835	0.9711	0.5903
AE [12]	0.9705	0.8367	0.9707	0.1942
SVM with AdaBoost [13]	0.9992	0.8231	**0.9995**	**0.7960**
AE with PRF [9]	0.9973	**0.8910**	0.9975	0.5921
AE-LGB with SMOTE [10]	0.9970	0.8275	0.9973	0.5574
MoE with SMOTE (This Work)	0.9976	0.8877	0.9978	0.6012
MoE with DNN-SMOTE (This Work)	**0.9993**	0.8469	**0.9995**	0.7883

MCC is a balanced measure that takes into account all four quadrants of the confusion matrix (TP, TN, FP, FN), making it particularly useful in imbalanced datasets. The SVM with AdaBoosteads with an MCC of 0.7960, indicating a strong overall performance that considers both the true and false classifications. The MoE with DNN-SMOTE model closely follows with an MCC of 0.7883, demonstrating that it offers a robust performance close to that of the SVM with AdaBoost, while also providing a higher TPR. This suggests that the MoE with DNN-SMOTE model offers a better trade-off between identifying frauds and minimizing false positives compared to other models.

In summary, Table 2 illustrates that while traditional modelsike SVM with AdaBoost achieve high accuracy and MCC, the newly proposed MoE models, particularly MoE with DNN-SMOTE, provide a strong balance between sensitivity and specificity. The slightlyower MCC for MoE models compared to SVM with AdaBoost may be attributed to their focus on enhancing TPR without compromising too much on TNR. This balanced performance is crucial for real-world applications where missing a fraudulent transaction could have significant financial repercussions. Hence, the MoE with DNN-SMOTE model appears to be a promising approach in credit card fraud detection, providing a more reliable detection system that mitigates the weaknesses of high-accuracy butow-sensitivity models.

4.7. Complexity and Efficiency Analysis

The complexity of the proposed MoE with DNN-SMOTE model includes two parts: complexity for the DNN-SMOTE module (Phase 1 and 2 in Figure 1) and complexity for the MoE classifier (Phase 3 in Figure 1). Compared to the naive SMOTE oversampling method [17], our proposed DNN-SMOTE model needs an additional training step for the encoder and decoder model, as in Figure 3.

DNN-SMOTE Complexity: Suppose the DNN-SMOTE model has Layers with dimensions d_i for eachayer. The complexity of forward and backward passes through the network is approximately: $\mathcal{O}\left(\sum_{i=1}^{L} d_{i-1} \cdot d_i \cdot N_{\text{samples}}\right)$, where d_0 is the feature dimension, N_{samples} is the number of samples in the imbalanced dataset. The summation accounts for the number of weight updates in eachayer during backpropagation. If the training involves E epochs, the total training complexity becomes $\mathcal{O}\left(E \cdot \sum_{i=1}^{L} d_{i-1} \cdot d_i \cdot N_{\text{samples}}\right)$. We can see that DNN-SMOTE complexity growsinearly with the number of samples and epochs, as well as the size of the encoder–decoder network. Based on our experimental results, the training and oversampling on GPU can be performed within 15s. Meanwhile, the DNN-SMOTE training and oversampling steps only need to be performed once to generate the balanced dataset. We believe the additional complexity due to DNN-SMOTE is acceptable.

MoE Complexity: The MoE model can consist of the gating network and a few experts. The gating network is a single-layer feedforward neural network, which takes the input features and outputs a Softmax distribution over the experts. Suppose the gating network has h_j neurons in the j-thayer. The complexity of a forward pass through the gating network is $\mathcal{O}\left(\sum_{j=1}^{g} h_{j-1} \cdot h_j\right)$. Each expert network in MoE is a traditional neural network. Suppose there are N experts, each with Layers and d_i neurons in the i-thayer. If all experts were trained, the total complexity of training the expert networks is $\mathcal{O}\left(N \cdot \sum_{i=1}^{L} d_{i-1} \cdot d_i \cdot N_{\text{samples}}\right)$. The proposed MoE complexity is about $N\times$ of neural network-based classifiers [9,10,12]. However, in practice, we found MoE execution on GPU is very fast (less than 10s for training and inference on the tested GPU) since the number of experts is small.

5. Conclusions

In the realm of financial transactions, credit card fraud detection is a critical aspect of ensuring transactional security and consumer trust. Fraudulent activities canead to substantial financialosses and damage the reputation of financial institutions. Credit card fraud often unfolds over time, making its detection a challenging yet essential task. Effective fraud detection methods are crucial for early identification and prevention of such activities, thereby ensuring the smooth operation of financial systems.

In this paper, we have presented a novel approach to credit card fraud detection that integrates a Mixture of Experts (MoE) model with a Deep Neural Network-based Synthetic Minority Over-sampling Technique (DNN-SMOTE). This combination is specifically designed to address the inherent challenges in fraud detection, particularly the class imbalance and the dynamic nature of fraudulent behaviors. The proposed methodeverages the strengths of MoE, which uses multiple specialized experts to capture complex fraud patterns, and DNN-SMOTE, which generates high-quality synthetic samples to improve the representation of the minority class.

Our experimental results, conducted on a publicly available credit card transaction dataset, demonstrate the significant improvements offered by our approach. The MoE with DNN-SMOTE method achieved an impressive classification accuracy of 99.93%, a true positive rate of 84.69%, and a true negative rate of 99.95%. Furthermore, the model attained a Matthews Correlation Coefficient (MCC) of 0.7883, underscoring its balanced performance in detecting both fraudulent and non-fraudulent transactions. These metrics clearly indicate that our proposed method outperforms traditional models, which often struggle with either high false positive rates or poor generalization to new fraud patterns.

In conclusion, the integration of MoE with DNN-SMOTE provides a robust and adaptable solution to the ongoing challenge of credit card fraud detection. The success of this approach in our experiments suggests that it has significant potential for application in real-world financial systems, where the accurate and timely detection of fraud is crucial. Future work could explore the extension of this framework to other domains with similar challenges, such as insurance fraud or cyber intrusion detection, and further refine the model to improve its scalability and efficiency in processingarge-scale transaction data.

Author Contributions: Z.Y. contributed to Conceptualization, Formal analysis, Investigation, Supervision, Writing—original draft. Y.W. contributed to Data curation, Resources, Methodology, Validation. H.S. contributed to Software, Visualization, Writing—review & editing. Q.Q. contributed to Funding acquisition, Project administration, Supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The used dataset is available at https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud (accessed on 12 August 2024).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Patel, K. Credit Card Analytics: A Review of Fraud Detection and Risk Assessment Techniques. *Int. J. Comput. Trends Technol.* **2023**, *71*, 69–79. [CrossRef]
2. Madhuri, T.S.; Babu, E.R.; Uma, B.; Lakshmi, B.M. Big-data driven approaches in materials science for real-time detection and prevention of fraud. *Mater. Today Proc.* **2023**, *81*, 969–976. [CrossRef]
3. Cherif, A.; Badhib, A.; Ammar, H.; Alshehri, S.; Kalkatawi, M.; Imine, A. Credit card fraud detection in the era of disruptive technologies: A systematic review. *J. King Saud Univ. Comput. Inf. Sci.* **2023**, *35*, 145–174. [CrossRef]
4. Sadgali, I.; Sael, N.; Benabbou, F. Detection of credit card fraud: State of art. *Int. J. Comput. Sci. Netw. Secur.* **2018**, *18*, 76–83.
5. Tian, X.; He, J.S.; Han, M. Data-driven approaches in FinTech: A survey. *Inf. Discov. Deliv.* **2021**, *49*, 123–135. [CrossRef]
6. Lebichot, B.; Paldino, G.M.; Siblini, W.; He-Guelton, L.; Oblé, F.; Bontempi, G. Incrementalearning strategies for credit cards fraud detection. *Int. J. Data Sci. Anal.* **2021**, *12*, 165–174. [CrossRef]
7. Barker, K.J.; D'amato, J.; Sheridon, P. Credit card fraud: Awareness and prevention. *J. Financ. Crime* **2008**, *15*, 398–410. [CrossRef]
8. Fair, L. Facts About Fraud From the FTC—And What It Means for Your Business. 2024. Available online: https://www.ftc.gov/business-guidance/blog/2024/02/facts-about-fraud-ftc-what-it-means-your-business (accessed on 12 August 2024).
9. Lin, T.H.; Jiang, J.R. Credit card fraud detection with autoencoder and probabilistic random forest. *Mathematics* **2021**, *9*, 2683. [CrossRef]
10. Du, H.; Lv, L.; Guo, A.; Wang, H. AutoEncoder and LightGBM for credit card fraud detection problems. *Symmetry* **2023**, *15*, 870. [CrossRef]
11. Ding, Y.; Kang, W.; Feng, J.; Peng, B.; Yang, A. Credit card fraud detection based on improved Variational Autoencoder Generative Adversarial Network. *IEEE Access* **2023**, *11*, 83680–83691. [CrossRef]
12. Pumsirirat, A.; Liu, Y. Credit card fraud detection using deepearning based on auto-encoder and restricted boltzmann machine. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*.
13. Randhawa, K.; Loo, C.K.; Seera, M.; Lim, C.P.; Nandi, A.K. Credit card fraud detection using AdaBoost and majority voting. *IEEE Access* **2018**, *6*, 14277–14284. [CrossRef]
14. Khalid, A.R.; Owoh, N.; Uthmani, O.; Ashawa, M.; Osamor, J.; Adejoh, J. Enhancing credit card fraud detection: An ensemble machineearning approach. *Big Data Cogn. Comput.* **2024**, *8*, 6. [CrossRef]
15. Zhou, Y.; Lei, T.; Liu, H.; Du, N.; Huang, Y.; Zhao, V.; Dai, A.M.; Le, Q.V.; Laudon, J. Mixture-of-experts with expert choice routing. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 7103–7114.
16. Yuksel, S.E.; Wilson, J.N.; Gader, P.D. Twenty years of mixture of experts. *IEEE Trans. Neural Networks Learn. Syst.* **2012**, *23*, 1177–1193. [CrossRef]
17. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
18. Dablain, D.; Krawczyk, B.; Chawla, N.V. DeepSMOTE: Fusing deepearning and SMOTE for imbalanced data. *IEEE Trans. Neural Networks Learn. Syst.* **2022**, *34*, 6390–6404. [CrossRef]
19. West, J.; Bhattacharya, M. Intelligent financial fraud detection: A comprehensive review. *Comput. Secur.* **2016**, *57*, 47–66. [CrossRef]
20. Raj, S.B.E.; Portia, A.A. Analysis on credit card fraud detection methods. In Proceedings of the 2011 International Conference on Computer, Communication and Electrical Technology (ICCCET), Tirunelveli, India, 18–19 March 2011; pp. 152–156.
21. Abdallah, A.; Maarof, M.A.; Zainal, A. Fraud detection system: A survey. *J. Netw. Comput. Appl.* **2016**, *68*, 90–113. [CrossRef]
22. Bolton, R.J.; Hand, D.J. Statistical fraud detection: A review. *Stat. Sci.* **2002**, *17*, 235–255. [CrossRef]
23. Vatsa, V.; Sural, S.; Majumdar, A.K. A rule-based and game-theoretic approach to online credit card fraud detection. *Int. J. Inf. Secur. Priv. (IJISP)* **2007**, *1*, 26–46. [CrossRef]
24. Gopal, R.K.; Meher, S.K. *A Rule-Based Approach for Anomaly Detection in Subscriber Usage Pattern*; World Academy of Science, Engineering and Technology: Chicago, IL, USA, 2007; pp. 396–399.
25. Duffield, N.; Haffner, P.; Krishnamurthy, B.; Ringberg, H. Rule-based anomaly detection on IP flows. In Proceedings of the IEEE INFOCOM 2009, Rio De Janeiro, Brazil, 19–25 April 2009; pp. 424–432.
26. Islam, S.; Haque, M.M.; Karim, A.N.M.R. A rule-based machineearning model for financial fraud detection. *Int. J. Electr. Comput. Eng. (IJECE)* **2024**, *14*, 759–771. [CrossRef]
27. Jawahir, I.; Balaji, A.; Rouch, K.; Baker, J. Towards integration of hybrid models for optimized machining performance in intelligent manufacturing systems. *J. Mater. Process. Technol.* **2003**, *139*, 488–498. [CrossRef]
28. Huang, Z.; Zhu, J.; Lei, J.; Li, X.; Tian, F. Tool wear predicting based on multi-domain feature fusion by deep convolutional neural network in milling operations. *J. Intell. Manuf.* **2020**, *31*, 953–966. [CrossRef]
29. Çakır, M.Y.; Şirin, Y. Enhanced autoencoder-based fraud detection: A novel approach with noise factor encoding and SMOTE. *Knowl. Inf. Syst.* **2024**, *66*, 635–652. [CrossRef]

30. Van Dyk, D.A.; Meng, X.L. The art of data augmentation. *J. Comput. Graph. Stat.* **2001**, *10*, 1–50. [CrossRef]
31. Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; Yang, Y. Random erasing data augmentation. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 13001–13008.
32. Dargan, S.; Kumar, M.; Ayyagari, M.R.; Kumar, G. A survey of deepearning and its applications: A new paradigm to machineearning. *Arch. Comput. Methods Eng.* **2020**, *27*, 1071–1092. [CrossRef]
33. Wong, S.C.; Gatt, A.; Stamatescu, V.; McDonnell, M.D. Understanding data augmentation for classification: When to warp? In Proceedings of the 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Gold Coast, QLD, Australia, 30 November–2 December 2016; pp. 1–6.
34. Jiang, L.; Huang, D.; Liu, M.; Yang, W. Beyond synthetic noise: Deepearning on controlled noisyabels. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 13–18 July 2020; pp. 4804–4815.
35. Mumuni, A.; Mumuni, F. Data augmentation: A comprehensive survey of modern approaches. *Array* **2022**, *16*, 100258. [CrossRef]
36. Garcea, F.; Serra, A.; Lamberti, F.; Morra, L. Data augmentation for medical imaging: A systematiciterature review. *Comput. Biol. Med.* **2023**, *152*, 106391. [CrossRef]
37. Zhou, S.; Zhang, J.; Jiang, H.; Lundh, T.; Ng, A.Y. Data augmentation with Mobius transformations. *Mach. Learn. Sci. Technol.* **2021**, *2*, 025016. [CrossRef]
38. Wang, X.; Wang, K.; Lian, S. A survey on face data augmentation for the training of deep neural networks. *Neural Comput. Appl.* **2020**, *32*, 15503–15531. [CrossRef]
39. Worldline and the Machine Learning Group of ULB (Université Libre de Bruxelles) Credit Card Fraud Detection. 2016. Available online: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud (accessed on 13 August 2024).
40. Awoyemi, J.O.; Adetunmbi, A.O.; Oluwadare, S.A. Credit card fraud detection using machineearning techniques: A comparative analysis. In Proceedings of the 2017 International Conference on Computing Networking and Informatics (ICCNI), Lagos, Nigeria, 29–31 October 2017; pp. 1–9.
41. Chicco, D.; Warrens, M.J.; Jurman, G. The Matthews correlation coefficient (MCC) is more informative than Cohen's Kappa and Brier score in binary classification assessment. *IEEE Access* **2021**, *9*, 78368–78381. [CrossRef]
42. Hanley, J.A. Receiver operating characteristic (ROC) methodology: The state of the art. *Crit Rev Diagn Imaging* **1989**, *29*, 307–335. [PubMed]
43. Hinton, G.E.; Roweis, S. Stochastic neighbor embedding. *Adv. Neural Inf. Process. Syst.* **2002**, *15*, 857–864.

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