

**ARTICLE****Credit Card Fraud Detection Using Variational Autoencoders****Edward Danso Ansong¹, David Adlai Nettey^{1,*}, Sarika S² and Simon Bonsu Osei¹**¹Department of Computer Science, University of Ghana, Legon, Accra, Ghana²Department of Artificial Intelligence and Data Science, Adi Shankara Institute of Engineering and Technology, Kalady, Kerala, India

*Corresponding Author: David Adlai Nettey. Email: danettey@st.ug.edu.gh

Received: 04 March 2025; Accepted: 27 October 2025; Published: 12 June 2026

ABSTRACT: Credit card fraud has emerged as a pervasive threat, impacting financial institutions and individuals as online banking and payment methods become increasingly integral to daily life. Despite efforts to mitigate this problem through measures like passwords and two-factor authentication, financial institutions continue to suffer substantial losses, often amounting to millions of dollars. Traditional machine learning solutions, developed and trained as supervised learning models, have failed to address this issue effectively. In anomaly detection, such as credit card fraud detection, the available training datasets are vast but inherently imbalanced, posing a formidable obstacle for supervised learning models in developing accurate hypothesis functions during model training. In response to this challenge, we propose an alternative approach using unsupervised machine learning models, specifically the Variational Autoencoder. In our study, we constructed the encoder and decoder components of our VAE using deep neural networks, and we implemented and trained them with pre-processed data using the TensorFlow and Keras libraries. Our model's effectiveness was assessed using the Area Under the Precision-Recall Curve (AUC-PR), yielding a robust score of 88%. Additionally, we subjected our model to an accuracy test on a test dataset, achieving a commendable accuracy score of 78%. Our model also attained 98% and 49% scores for precision and recall, respectively, both commendable for an anomaly detection model. The results of our study underscore the viability of unsupervised learning models, particularly VAE, for credit card fraud detection, demonstrating their potential as a machine learning solution to mitigate credit card fraud. This research offers a promising avenue for bolstering security measures in financial transactions and reducing the impact of fraudulent activities.

KEYWORDS: Credit card fraud; neural networks; variational autoencoders; autoencoders; imbalanced datasets; area under the precision-recall curve (AUC-PR); unsupervised learning

1 Introduction

Technology shifts attributed to research and improved hardware and software components used in the financial sector have led to online banking, commonly known as E-banking. Although it has significantly improved how banking services are rendered, some security issues raise significant concerns for both banks and customers of E-banking services; one such issue is Credit Card Fraud, which this paper seeks to address. "Credit Card Fraud refers to using forged credit cards to purchase goods without paying, and it is also associated with identity theft, such as using others' credit cards illegally to defraud money or property." [1]. In 2010 it was concluded that damages obtained from credit card fraud alone, amongst other types of financial frauds, reached \$7.6 billion and further increased to \$21.81 billion in 2015, representing a staggering 300% increase within five years [2]. These two alone highlight how necessary it is to constantly improve the security measures put in place to mitigate fraudulent activities by fraudsters on E-banking platforms. With many

advancements in machine learning tools and techniques, researchers in the field have explored many ways to tackle credit card fraud, from single pattern recognition methods such as neural networks and decision trees to combined methods; these solutions have been implemented and deployed to detect credit card fraud [1]. Although these systems efficiently detect credit card fraud, they still have a few deficiencies, especially with system models developed based on supervised learning, which need a balanced dataset of normal and fraudulent transactions [1]. For every fraudulent transaction that is correctly identified, about forty non-fraud transactions are detected as fraud [2]. This report shows that more work is needed to improve the prediction accuracy of models created to detect fraudulent activities relating to credit card users.

With the recent advancement of machine learning algorithms due to computer hardware innovation and improvement, some ML algorithms, such as Deep Neural Networks, are being used to detect malicious credit card activities. However, datasets used in modelling are highly imbalanced (i.e., the number of positive labels are higher than that of negative labels and *vice versa*); this makes these models ineffective and unable to flag fraudulent credit card activities with reasonable accuracy [2]. Therefore, this research aims to use unsupervised learning techniques, specifically Variational Autoencoders, to develop a predictive model to detect fraudulent activities based on credit card usage patterns studied by the model when trained with imbalanced datasets.

This research aims to improve implemented and deployed mechanisms using unsupervised learning algorithms to improve the accuracy of predictions of models trained on unbalanced datasets used in detecting credit card fraud.

The objectives of this research are to develop a predictive model that can study and understand the usage pattern of a credit card user and flag any malicious credit card activity using an unsupervised neural network algorithm called Variational Autoencoders, and attain a prediction accuracy of 70% or more.

2 Literature Review

The research on credit card fraud detection has evolved with the implementation of various machine learning algorithms such as Logistic Regression, Support Vector Machines, and Decision Trees. However, the challenge lies in handling imbalanced datasets, which affects the models' accuracy in real-world scenarios. Researchers address this by resampling datasets through under-sampling or over-sampling techniques like SMOTE. Autoencoders, specifically Variational Autoencoder predictive models, have shown promise in detecting fraudulent credit card activities by identifying underlying patterns in imbalanced data. This section reviews related studies, discusses methodologies, and explores drawbacks to inform our proposed solution.

2.1 Anomaly Detection

Anomalies are deviations from expected behaviour and can be classified as atomic univariate, atomic multivariate, and aggregate anomalies. Anomaly detection involves identifying patterns not conforming to expected behaviour. Industries such as manufacturing are leading the charge to advance machine learning for anomaly detection, which is crucial for predictive maintenance. However, detecting anomalies is challenging due to their infrequent occurrence, resulting in imbalanced datasets. Anomaly detection falls into three categories based on dataset types: Supervised, Unsupervised, and Semi-supervised anomaly detection [3].

2.1.1 Supervised Anomaly Detection

Supervised Anomaly Detection utilises supervised learning techniques to distinguish between normal and anomalous data points [3]. It relies on balanced training datasets containing substantial examples of normal and anomalous data for optimal predictive model accuracy [4]. Algorithms like Random Forest,

Logistic Regression, and SVMs have been employed in researching the performance of supervised learning for anomaly detection [5]. However, challenges arise when dealing with imbalanced datasets, hindering these algorithms from achieving high prediction accuracy levels even with proper optimisation [6].

2.1.2 Semi-Supervised Anomaly Detection

Semi-supervised anomaly detection is frequently favoured over supervised methods and operates with single-labelled datasets, unlike the balanced datasets required by supervised anomaly detection [3]. It is often unsupervised, allowing models to learn patterns during training to detect unknown anomalies. The flexibility of semi-supervised anomaly detection to handle imbalanced datasets makes it preferable, as many anomaly datasets are skewed toward normal data points [5].

2.1.3 Unsupervised Anomaly Detection

Unsupervised Anomaly Detection relies on unsupervised learning algorithms to identify anomalous data points in large datasets [3]. Unlike supervised methods, it does not require labelled datasets for training, aiming to discover patterns within the dataset to distinguish between normal and anomalous data points [1]. Unsupervised learning algorithms are often employed as feature extraction tools due to their pattern recognition capabilities, supporting subsequent use in supervised anomaly detection models [6]. For instance, Lin & Jiang utilised autoencoders (unsupervised learning) for feature extraction and Random Forest (supervised learning) for classification in a specific implementation.

2.2 Autoencoders

Initially designed for unsupervised learning, Autoencoders aim to reconstruct input with minimal error [7]. Defined as neural networks, autoencoders learn to copy input to output [8] and discover underlying data correlations in a smaller dimension. Comprising an encoder, code (i.e., latent feature representation), and decoder, autoencoders mathematically define encoder and decoder as $h = (x)$ and $r = (h)$, respectively [7]. The bottleneck (i.e., latent features representation or code) prevents perfect input replication, aiming to understand relevant input features. In autoencoder structures, smaller nodes in the bottleneck force the model to extract only pertinent features [7]. Autoencoder performance is gauged using the reconstruction error function, mathematically expressed as the difference between input and reconstructed output through mean square error or binary cross-entropy. Variations of autoencoders, like sparse, denoising, and contractive autoencoders, are tailored for specific tasks like feature extraction. However, this research focuses on variational autoencoders, which will be explained in the subsequent section. Fig. 1 shows the architecture of autoencoder.

2.3 Variational Autoencoders

Variational Autoencoders (VAEs), a prevalent variation of autoencoders in unsupervised learning, fall under generative models [8]. Unlike Generative Adversarial Networks (GANs), VAEs differ structurally and methodologically in loss function and complexity. In their study, Kingma and Welling [8] define VAEs as “a directed probabilistic graphical model” approximated by a neural network, forming an autoencoder-like architecture. They structurally define VAEs as two independently parameterised models, the encoder and decoder, collaborating to generate varying input instances. In VAEs, the highest layer is the latent variable (z), initiating the generative process. The optimisation objective is the evidence lower bound, expressed using Jensen’s inequality, encompassing the Kullback-Leibler divergence (KL divergence) and the variational lower bound on the marginal likelihood of the data point [8]. Mathematically, this is represented as $\log p_{\theta}(x(i)) = DKL(q_{\theta}(z|x) || p_{\theta}(z)) + L(\theta, \phi; x(i))$, where $p_{\theta}(z)$ is the prior distribution and $q_{\theta}(z|x)$ is the

approximate posterior of z . The encoder ($q_{\phi}(z|x)$) and decoder ($p_{\theta}(x|z)$) correspond to the variational autoencoder's key components, distinguishing it from traditional autoencoders. VAEs represent latent features using parameter distributions, allowing them to generate diverse outputs. This distribution depends on the nature of the training data, classifying VAEs as generative models. Requiring backpropagation for training, VAEs utilise a technique called reparameterization. The reparameterization trick involves using a random variable from a standard normal distribution, mitigating the high variance associated with the Monte Carlo method traditionally used for calculating the variational lower bound [8]. Fig. 2 represents an approximate posterior of a variational autoencoder.

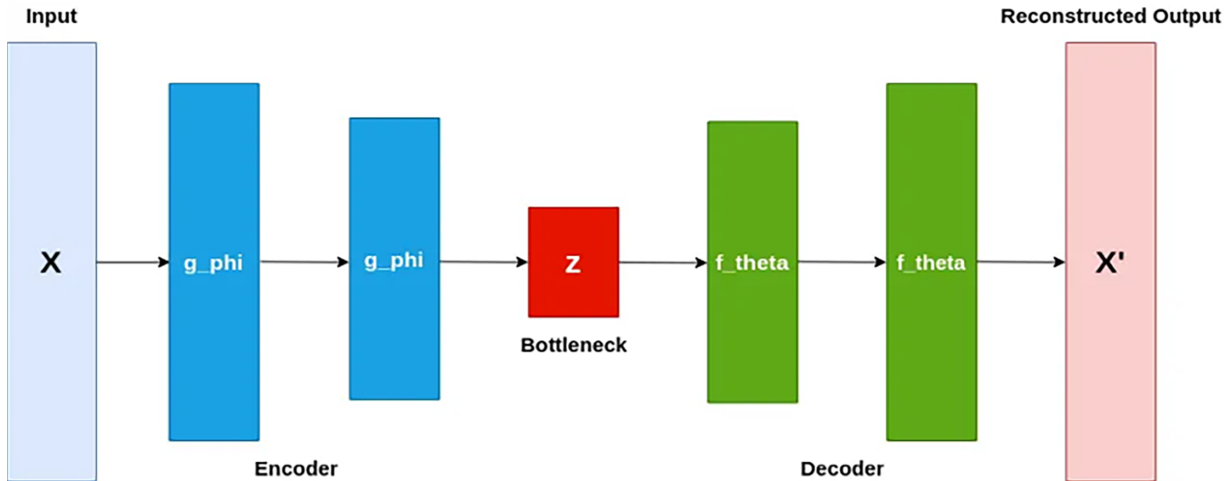


Figure 1: Architecture of an autoencoder. Reprinted from Bergmann and Stryker [9].

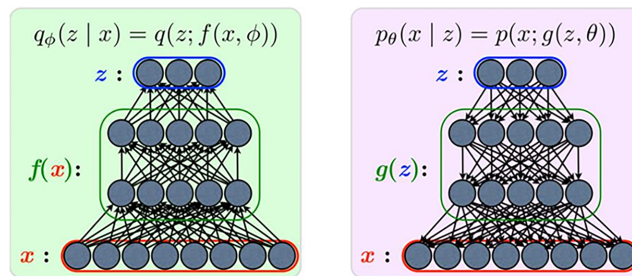


Figure 2: Representation of an approximate posterior of a variational autoencoder. Reprinted from An and Cho [10].

3 Methodology

3.1 Experimental Setup

For the experiment used in this research, we used an HP Pavilion personal computer with the following specifications: 11th Generation Intel Core i5-1135G7 processor (CPU), an installed RAM of 8 GB and an Intel Iris X graphics card. The software environment comprised Python, TensorFlow, TensorFlow Probability, NumPy, Scikit-learn and Pandas. These libraries were selected due to their robustness and compatibility with deep learning and anomaly detection tasks. These specific hardware and software configurations were chosen to simulate a realistic resource-constrained environment while ensuring the model can perform well. This was necessary because of the limited available resources when this research was conducted. No additional GPU acceleration was used during training. CPU-based optimisation techniques such as batch

processing and datatype precision adjustments (i.e., float32 instead of float64) were employed to ensure efficient memory usage.

3.2 Data Acquisition

The dataset used in this study was sourced from the machine learning Group-ULB repository, initially created in 2018 and updated in 2021. Specifically, the credit card detection dataset was obtained from Kaggle and is still publicly accessible. According to the dataset owners, it consists of credit card transactions made by European cardholders for two days. The dataset, available in CSV format, contains 284,807 transactions, with only 492 being fraudulent, constituting 0.17% of the overall dataset. This significant class imbalance characterises the dataset as highly imbalanced. This dataset was chosen over publicly available alternatives, like PaySim, due to its real-world relevance, authenticity, and established benchmarking history in fraud detection literature.

3.3 Data Cleaning and Normalisation

In this phase, we aimed to enhance the dataset quality for improved model performance. Initially, we converted dataset values from 'float64' to 'float32' to ensure compatibility with the model's training requirements and optimise the values for effective utilisation. Subsequently, the Standard Scaler from Scikit-learn was applied to normalise the range of values in the 'Amount' column, mitigating complexity stemming from large value ranges. Another reason why it was used was due to its compatibility with the assumptions of Gaussian latent space distributions. The 'time' column was excluded from the dataset to streamline and refine its structure. It also did not significantly contribute to the model's predictive capability and was found to be weakly correlated with fraudulent behaviour. Additionally, duplicate records were checked and removed, although the dataset was relatively clean with no missing values.

3.4 Data Splitting

This step focused on dividing our dataset into the training and testing datasets. Given that the original dataset from Kaggle lacked predefined partitions, we employed a splitting ratio of 70:30. This ratio designated 70% of the dataset as the training dataset, utilised for training the model. The remaining 30% constituted the testing dataset employed for model evaluation. This ratio is commonly used to ensure adequate data is provided for both training and evaluation. This split did not use stratified sampling because it would have introduced a slight distribution shift. Hence, the under-sampling technique was used to reduce the majority class in the training dataset. This enhanced the model's ability to recognise rare fraudulent patterns during training without artificially inflating the positive class.

3.5 Model Development

3.5.1 Encoder

As outlined by Welling and Kingma, a Variational Autoencoder consists of three integral components: the encoder, the decoder, and the code. They emphasise that the decoder functions as the inverse of the posterior (i.e., the encoder). Aligning with this structural principle, our study adheres to the concept that the deep neural network in our decoder is the exact opposite of our encoder. The encoder in our model is a sequential model consisting of five layers, forming a deep neural network. The first layer acts as the input layer, receiving input data from the neural network. Comprising five Dense layers and a single Multivariate-Normal-Tril layer, which acts as the probabilistic layer in this model's architecture from TensorFlow Probability, the hidden layers generate a probability distribution of learned input features, per Welling and Kingma's framework. Consequently, a probability layer serves as the final layer in our encoder.

This probabilistic layer allows the encoder to output a multivariate Gaussian distribution from which latent representations are sampled. The selection of five layers and the dimensionalities used were based on iterative hyperparameter tuning, involving grid search and empirical performance comparisons.

3.5.2 Decoder

As previously mentioned, the decoder's structure in a variational autoencoder is the exact inverse of its encoder. However, the primary function of a variational autoencoder, akin to an autoencoder as per Welling and Kingma, is to replicate the same or similar structures of its input, generating variations of the input. In line with this, the decoder's role is to reproduce the input using the output of the encoder, which is the probability distribution. Contrary to the encoder, the inverse structure of the encoder excludes a probability layer. Our decoder's structure comprises an input layer, serving a purpose analogous to the encoder, but with its input being the output of the encoder. Additionally, the decoder consists of five Dense layers forming the hidden layer of the model, with the final layer acting as the ultimate output of the variational autoencoder.

3.5.3 Activation Function

Our study employs the Leaky Rectified Linear Unit (ReLU) as the activation function for the Dense layers in our Variational Autoencoder's Encoder and Decoder. The Leaky ReLU serves as a variation of the standard ReLU function, addressing the "dying problem" often associated with traditional ReLU activation. We opt for the Leaky ReLU because it can introduce sparsity, enhance the model's generalisation performance during training, and mitigate the risk of overfitting.

3.5.4 Loss Function

In Variational Autoencoders, the loss function consists of the reconstruction loss and the Kullback-Leibler divergence, commonly known as the reparameterisation trick [11]. The loss function is the sum of these two. Notably, the Kullback-Leibler divergence optimises the probability distribution during training in a Variational Autoencoder [9]. This distinction is the sole difference between the loss functions of a Variational Autoencoder and an Autoencoder. For the reconstruction loss, during the training phase, the model adjusts weights to ensure that the resulting outputs closely resemble the input, aiming for high-quality outputs [11].

3.6 Model Training

After developing the Variational Autoencoder model, we trained it with our segmented training dataset to familiarise it with potential production data, aiming to learn essential features of valid credit card transactions. Following Welling and Kingma's approach, the encoder grasped relevant input features, and the decoder proficiently reconstructed inputs. During training, we focused on the loss metric, which is crucial for evaluating the model's performance on the test dataset. In an Autoencoder, the key metric is reconstruction loss, while in a Variational Autoencoder, the critical metric is the loss value from the entire loss function. We opted for 50 epochs, considering it optimal based on achieving the best loss values compared to epochs 20, 30, and 60. While previous works have explored using VAEs for feature extraction in ensemble architectures, our study tests their standalone efficacy in anomaly detection.

3.7 Threshold Selection

In standard and probabilistic (i.e., variational) autoencoders, the reconstruction error is highly considered and monitored for anomaly detection [12]. The higher the reconstruction error score, the more likely an

anomaly has occurred [12]. Hence, there is a need for a threshold score. The purpose of the threshold score is to better assist in identifying an anomaly by comparing the reconstruction error score to the threshold score. Our research generates our threshold score by calculating the mean value obtained from our test dataset's negative log-likelihood (i.e., reconstruction error).

4 Evaluation

4.1 Area Under Precision-Recall (AUC-PR)

Our study used the Precision-Recall (PR) curve over the Receiver Operating Characteristic (ROC) curve to evaluate our model's performance. According to Khan and Rana, the PR curve is preferable in classifier tasks with highly imbalanced datasets, as it employs Precision as a performance indicator, which is more reliable in such scenarios [13]. This choice is particularly advantageous for our study, where the class distribution is skewed heavily towards non-fraudulent credit card activities, making the PR curve more suitable for assessing our model's effectiveness. Furthermore, although data imbalance was addressed to some extent through sampling, advanced techniques such as hybrid sampling methods (e.g., SMOTE with Tomek Links) and cost-sensitive learning were not applied. These techniques can potentially improve model robustness and are suggested for future work [14].

4.2 Model's AUC-PR Score

In our evaluation, we employed the AUC-PR curve to assign an AUC-PR score to our model, aiding in determining its performance as a threshold anomaly detector based on Precision and Recall. Out of 1,142 datasets (comprising both fraudulent and non-fraudulent instances) sampled from the original test dataset, our model achieved an AUC-PR score of 88%. This score indicates the model's effectiveness as a threshold anomaly detector, showcasing its balanced Precision and Recall performance. Refer to the plotted AUC-PR curve below in Fig. 3 for a visual representation of our model's evaluation.

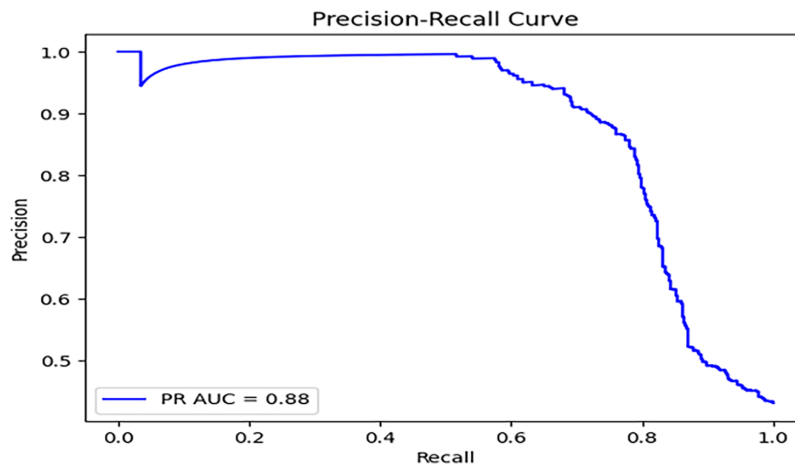


Figure 3: AUC-PR Curve for our VAE model. Note. The AUC-PR curve illustrates the model's performance on the imbalanced dataset, where higher values indicate improved precision-recall trade-offs.

While we explored the effectiveness of VAEs as an unsupervised anomaly detection technique, we did not compare their performance directly with other state-of-the-art unsupervised methods such as Isolation Forests and Support Vector Machines (SVMs). Future work can include comparative analysis to confirm or challenge the VAE's superiority in similar tasks.

4.3 Model's Accuracy Score

We determined a threshold score to assess our anomaly detection model's accuracy (i.e., the AUC-PR analysis mean 327.11, rounded to two decimal places). Evaluating the model with 1,142 sampled data points (fraudulent and non-fraudulent credit card activities), we achieved a 78% accuracy score. Despite being slightly lower than the estimated AUC-PR score of 88%, the obtained accuracy score is still commendable. Refer to the plotted confusion matrix in Fig. 4 for a visual representation of our threshold-based model evaluation.

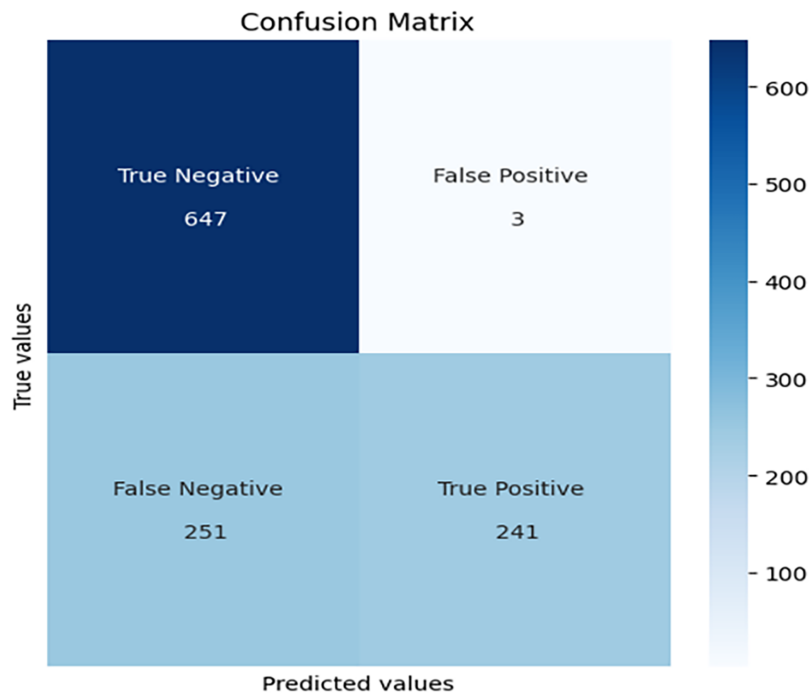


Figure 4: Confusion matrix of the VAE. Note: The confusion matrix summarises our model's classification outcomes based on the threshold applied.

4.4 Model's F1-Score and Confusion Matrix

We calculate other evaluation metrics from the numbers obtained from our confusion matrix: Precision score = 0.98, Recall = 0.49, F1-Score = 0.65, False Positive Rate = 0.005. These results show a model with high precision, suggesting minimal false alarms, which is crucial in production-grade fraud detection systems. However, the recall is relatively low, indicating that a significant proportion of fraud cases are missed. This reinforces the trade-off between avoiding false positives and capturing all fraudulent instances. Improving recall while maintaining precision will be critical in future iterations.

5 Conclusion

With the rise of fraudulent credit card activities posing a growing challenge, machine learning models have been employed to detect and mitigate this issue. However, traditional supervised models struggle to identify new fraudulent patterns effectively due to imbalanced datasets. To address this limitation, we adopted an unsupervised approach using Variational Autoencoders. Leveraging deep neural networks for the encoder and decoder, our model learned underlying features for identifying fraudulent credit card activities. Evaluation results from employing a threshold score indicate our model's ability to detect fraudulent activities

with an accuracy score of 78% based on reconstruction probability. This approach provides a promising alternative for addressing challenges posed by imbalanced datasets in fraud detection.

Future Works

During our study, we faced challenges that merit consideration for future research. While plotting the visual distribution of the reconstruction probability for fraudulent and non-fraudulent credit card activities, we observed some overlap, impacting True-Negatives and True-Positives scores in the evaluation. The selected threshold score also significantly influenced our model's high accuracy. Therefore, we propose further exploration into threshold scores for anomaly detection models, including methodologies for their determination. Some of these methodologies include implementing dynamic thresholding techniques, reinforcement techniques and methods for threshold selection. Addressing these aspects can enhance the robustness and reliability of future fraud detection studies.

Another area to explore would be more advanced sampling techniques like SMOTE with Tomek Links to provide better class separation, which can improve recall scores. In addition to using other sampling techniques, we suggest using traditional fraud detection methods, such as Isolation Forests, which could yield an ensemble model that leverages the strengths of both generative learning and established anomaly detection strategies. For instance, VAE-based reconstruction scores could be used as features in downstream supervised models to enhance robustness and interpretability.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Study Conception and Design: Edward Danso Ansong and Simon Bonsu Osei; Data Collection: David Adlai Nettey and Simon Bonsu Osei; Analysis and Interpretation of Results: David Adlai Nettey; Draft Manuscript Preparation: Edward Danso Ansong and David Adlai Nettey; Manuscript Revising and Editing: Sarika S. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are openly available in Machine Learning Group—ULB at <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>.

Ethics Approval: Not applicable.

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

References

1. Chen J, Shen Y, Ali R. Credit card fraud detection using sparse autoencoder and generative adversarial network. In: 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON); 2018 Nov 1–3; Vancouver, BC, Canada. p. 1054–9. doi:10.1109/IEMCON.2018.8614815.
2. Zamini M, Montazer G. Credit card fraud detection using autoencoder based clustering. In: 2018 9th International Symposium on Telecommunications (IST); 2018 Dec 17–19; Tehran, Iran. p. 486–91. doi:10.1109/ISTEL.2018.8661129.
3. Nassif AB, Abu Talib M, Nasir Q, Dakalbab FM. Machine learning for anomaly detection: a systematic review. IEEE Access. 2021;9:78658–700. doi:10.1109/ACCESS.2021.3083060.
4. Goy G, Gezer C, Gungor VC. Credit card fraud detection with machine learning methods. In: 2019 4th International Conference on Computer Science and Engineering (UBMK); 2019 Sep 11–15; Samsun, Turkey. p. 350–4. doi:10.1109/UBMK.2019.8906995.
5. Itoo F, Meenakshi, Singh S. Comparison and analysis of logistic regression, naïve bayes and KNN machine learning algorithms for credit card fraud detection. Int J Inf Technol. 2021;13(4):1503–11. doi:10.1007/s41870-020-00430-y.

6. Xie Y, Liu G, Cao R, Li Z, Yan C, Jiang C. A feature extraction method for credit card fraud detection. In: 2019 2nd International Conference on Intelligent Autonomous Systems (ICoIAS); 2019 Feb 28–Mar 2; Singapore. p. 70–5. doi:10.1109/ICoIAS.2019.00019.
7. Michelucci U. Introduction to autoencoders. Towards data science; 2022 [cited 2025 Jan 1]. Available from: <https://towardsdatascience.com/introduction-to-autoencoders-b6fc3141f072/>.
8. Kingma DP, Welling M. An introduction to variational autoencoders. FNT Mach Learn. 2019;12(4):307–92. doi:10.1561/22000000056.
9. Bergmann D, Stryker C. What is a variational autoencoder? [cited 2024 Jun 12]. Available from: <https://www.ibm.com/think/topics/variational-autoencoder>.
10. An J, Cho S. Variational autoencoder based anomaly detection using reconstruction probability. Seoul, Republic of Korea: SNU Data Mining Center; 2015.
11. Asperti A, Trentin M. Balancing reconstruction error and Kullback-Leibler divergence in variational autoencoders. IEEE Access. 2020;8:199440–8. doi:10.1109/ACCESS.2020.3034828.
12. Fernández-Saúco IA, Acosta-Mendoza N, Gago-Alonso A, García-Reyes EB. Computing anomaly score threshold with autoencoders pipeline. In: Progress in pattern recognition, image analysis, computer vision, and applications. Cham, Switzerland: Springer International Publishing; 2019. p. 237–44. doi:10.1007/978-3-030-13469-3_28.
13. Fu GH, Yi LZ, Pan J. Tuning model parameters in class-imbalanced learning with precision-recall curve. Biom J. 2019;61(3):652–64. doi:10.1002/bimj.201800148.
14. Pes B, Lai G. Cost-sensitive learning strategies for high-dimensional and imbalanced data: a comparative study. PeerJ Comput Sci. 2021;7(2):e832. doi:10.7717/peerj-cs.832.