Balancing Efficacy and Efficiency: Identifying the Optimal Model for Credit Card Fraud Detection

Abdel Rahman Nasir
University of Western Ontario
Bachelor's of Management and Organizational Studies
Specialization in Accounting

Abstract

This study addresses the escalating challenge of credit card fraud in the digital finance era. With the objective of identifying a model that adeptly balances precision in fraud detection with computational efficiency, this research explores and evaluates various machine learning and deep learning techniques. Utilizing a dataset from the Machine Learning Group at Université Libre de Bruxelles, encompassing 284,807 transactions with a severe class imbalance, the study employs a hybrid approach of Synthetic Minority Over-sampling Technique (SMOTE) and majority random under-sampling to enhance model training. A range of models including Naive Bayes, Logistic Regression, K-Nearest Neighbors, SGDClassifier, LightGBM, and a TensorFlow-based Autoencoder are meticulously analyzed using metrics such as accuracy, recall, precision, F2 score, Matthews Correlation Coefficient (MCC), and Receiver Operating Characteristic Area Under the Curve (ROC AUC). The study reveals LightGBM and SGDClassifier as standout models in terms of their ability to effectively and efficiently detect fraudulent transactions. This research contributes to the domain of fraud detection by offering insights into optimal modeling techniques that balance detection capabilities with the practicalities of real-world application, providing valuable guidance for financial institutions in enhancing their fraud prevention frameworks.

Keywords: Credit Card Fraud Detection, Machine Learning, Deep Learning, SMOTE, Class Imbalance, Computational Efficiency

I. Introduction

The ubiquity of credit card transactions in the financial landscape is both a marker of progress and vulnerability. In 2022, the volume of these transactions worldwide highlighted the scale at which the system operates, yet it also brought to light the stark reality of fraud prevalence, with losses amounting to $32.34 billion (USD) and an anticipated increase in the coming years.[1] These statistics not only underline the economic impact, but also the necessity for advanced detection mechanisms.

The modern era of e-commerce and contactless payments has seen an exponential rise in credit card usage, with an estimated 1.1 trillion cards in the United States alone, making it a prime target for fraudsters.[2] This increase in usage has been paralleled by a significant rise in credit card fraud cases, with Americans reporting 271,823 cases in 2019, a 72.4% increase from the previous year.[3] Such trends highlight the challenges in monitoring transactions due to their sheer volume, often resulting in overlooked fraudulent activities

[1] The Nilson Report, 'Payment Card Fraud Losses Reach $32.34 Billion,' GlobeNewswire, December 22, 2022

[2] Tzu-Hsuan Lin and Jehn-Ruey Jiang, "Credit Card Fraud Detection with Autoencoder and Probabilistic Random Forest," *Mathematics* 9, no. 21 (2021): 2683, https://doi.org/10.3390/math9212683.

[3] Lin and Jiang, "Credit Card Fraud Detection."

and substantial losses for both cardholders and issuers.[4]

As the financial sector increasingly adopts internet-based transaction methods, the challenge of securing online credit card transactions against malicious users becomes paramount. With the forecasted rise in fraudulent transactions expected to exceed 30 billion pounds annually by 2027, the importance of effective fraud detection strategies cannot be overstated.[5] The surge in Card-Not-Present (CNP) frauds, regarded as the most serious threat to the credit card industry, underscores the need for sophisticated machine learning techniques in fraud analysis, detection, and prevention.[6]

This study delves into the application of machine learning algorithms to tackle the challenges of credit card fraud detection, with an emphasis on balancing detection performance with computational efficiency. A critical obstacle in developing such models is the inherent class imbalance present in transactional data, where fraudulent instances are significantly outnumbered by legitimate transactions. To address this imbalance, a hybrid approach is employed, combining Synthetic Minority Over-sampling Technique (SMOTE), with majority random under-sampling strategies, aiming to refine the training dataset for better model generalization.

We explore machine learning models, including Naive Bayes, Logistic Regression, K-Nearest Neighbors, Stochastic Gradient Descent Classifier, and LightGBM, and extend our investigation to include Deep Learning techniques using Autoencoders. Each model's performance is meticulously evaluated on a set of metrics that account for both the precision of fraud detection, as well as the practicality of deployment in real life systems. The hybrid sampling strategy's impact on model efficacy is also scrutinized, offering insights into the trade-offs between improving detection rates and maintaining operational efficiency.

Through this comprehensive analysis, the research aims to contribute to the domain of fraud detection by pinpointing optimal modeling techniques that can be pragmatically adopted by financial institutions to enhance their fraud prevention frameworks.

II. Background and Related Work

Credit card fraud detection has undergone significant evolution, transitioning from traditional rule-based systems to advanced machine learning models. A study by Van Vlasselaer highlights the limitations of expert rules in adapting to the dynamic strategies employed by fraudsters.[7] Dal Pozzolo further emphasizes the need for machine learning methods to continuously evolve, noting that static models failing to adapt to new fraud strategies quickly become obsolete.[8] This shift has been catalyzed by the growing sophistication of fraudulent activities, necessitating more advanced detection methodologies.[9]

[4] Lin and Jiang, "Credit Card Fraud Detection."

[5] F. K. Alarfaj et al., "Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms," IEEE Access 10 (2022): 39700-39715, https://doi.org/10.1109/ACCESS.2022.3166891.

[6] Alarfaj et al., "Credit Card Fraud Detection," 39700.

[7] V. Van Vlasselaer et al., "APATE: A Novel Approach for Automated Credit Card Transaction Fraud Detection using Network-Based Extensions," Decision Support Systems 75 (2015): 38–48, https://doi.org/10.1016/j.dss.2015.04.013.

[8] A. Dal Pozzolo et al., "Learned lessons in credit card fraud detection from a practitioner perspective," Expert Systems with Applications 41, no. 10 (2014): 4915–4928, https://doi.org/10.1016/j.eswa.2014.02.026.

[9] Alejandro Correa Bahnsen, "Feature Engineering Strategies for Credit Card Fraud Detection," accessed December 4, 2023, https://albahnsen.github.io/files/Feature%20Engineering.

The incorporation of machine learning techniques in fraud detection systems has garnered increasing interest in recent years. Subsequent studies have noted the successful application of these techniques in detecting fraudulent transactions.[10] The focus has been on various machine learning models, including Gradient Boosting, Support Vector Machines, Decision Trees, Logistic Regression, and Random Forests. Research has shown these models to achieve high performance, particularly when datasets are balanced through techniques like undersampling.[11]

One of the critical challenges in credit card fraud detection is the class imbalance inherent in datasets, where fraudulent transactions are far fewer than legitimate ones. Additionally, the cost-sensitive nature of the problem emphasizes that the financial impact of false positives and false negatives varies greatly, influenced by transaction amounts.[12] Studies have proposed a cost-based measure for evaluating fraud detection models that account for these varying financial costs.[13] This approach underscores the need for models not only to be accurate but also to be sensitive to the financial implications of their predictions.

Recent studies have expanded the scope of machine learning applications in fraud detection. One study investigated the use of Adaptive Boosting and Majority Voting methods, demonstrating the high accuracy of AdaBoost-SVM models.[14] Other papers compared different neural network architectures, finding that Multilayer Perceptrons (MLPs) outperformed Extreme Learning Machines (ELMs) in terms of accuracy, albeit with higher complexity.[15] These comparative analyses provide a foundation for understanding the relative strengths and weaknesses of various algorithms in fraud detection.

Machine learning has become integral in addressing credit card fraud, evolving from simple decision tree models to more complex neural networks and Bayesian approaches. However, complexities like feature redundancy and model overfitting pose significant challenges. Convolutional neural networks have been introduced to mitigate these issues by effectively reducing feature redundancy.

Credit card fraud detection, primarily a binary classification problem, has seen various categorizations and methodologies. These include traditional and merchant-related frauds, as well as internet frauds, with different sampling approaches applied to skewed transaction data. Research indicates that a 10:90 fraud-to-legitimate

[10] S. Bhattacharyya et al., "Data mining for credit card fraud: A comparative study," Decision Support Systems 50, no. 3 (2011): 602–613, https://doi.org/10.1016/j.dss.2010.08.008.

[11] S. Khatri, A. Arora, and A. P. Agrawal, "Supervised machine learning algorithms for credit card fraud detection: A comparison," in *Proc. 10th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence)*, Jan. 2020, pp. 680–683.

[12] R. J. Bolton, D. J. Hand, F. Provost, and L. Breiman, "Statistical fraud detection: A review," Statistical Science 17, no. 3 (2002): 235–255, https://doi.org/10.1214/ss/1042727940.

[13] A. Correa Bahnsen, D. Aouada, and B. Ottersten, "Example-dependent cost-sensitive logistic regression for credit scoring," in Proceedings of the 2014 13th International Conference on Machine Learning and Applications (Detroit, USA: IEEE, 2014), 263–269, https://doi.org/10.1109/ICMLA.2014.48.

[14] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi, "Credit Card Fraud Detection Using AdaBoost and Majority Voting," IEEE Access 6 (2018): 14277-14284, https://doi.org/10.1109/ACCESS.2018.2806420.

[15] F. Z. El Hlouli et al., "Credit card fraud detection based on multilayer perceptron and extreme learning machine architectures," in Proc. Int. Conf. Intell. Syst. Comput. Vis. (ISCV), Jun. 2020, pp. 1–5.

case distribution often yields the best performance, balancing realism with model efficacy.[16]

The field of credit card fraud detection, while critical, suffers from limited public research, primarily due to data privacy concerns. Supervised learning methods dominate the field, employing various balancing techniques and dynamic models to adapt to evolving fraud patterns. Unsupervised methods focus on behavior changes and unusual transaction patterns, often used in conjunction with supervised methods for effective fraud detection.[17]

In tackling the high-dimensional feature space of credit card data, feature selection algorithms like Genetic Algorithm have proven effective, especially when used with methods like Random Forest, which can handle large variable numbers and noisy data.[18]

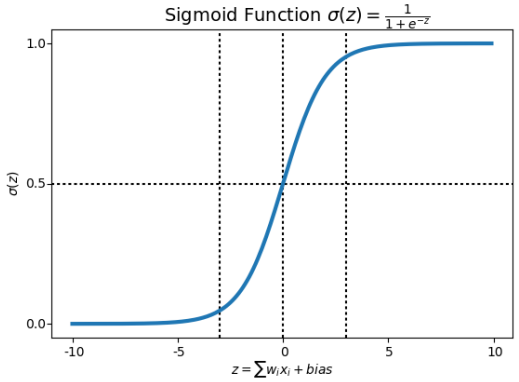
Finally, the autoencoder with probabilistic random forest (AE-PRF) method stands out for its efficient handling of imbalanced data, a pervasive issue in credit card fraud detection. This method combines feature extraction via autoencoders with the probabilistic classification capabilities of random forests, showing promise in improving detection performance.[19]

III. Principles of Fraud Detection Models

A. Logistic Regression

Logistic Regression, a cornerstone in binary classification, is particularly apt for fraud detection. Central to this model is the logistic function $\sigma(z) = \frac{1}{1+e^{-z}}$, where **z** represents the linear combination of the input features

$z = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \ldots + \omega_n x_n$.



This formulation allows the model to weigh each feature x_i, such as transactional attributes, according to its learned coefficient ω_i. The sigmoid function then maps these weighted sums to a probability value, indicating the likelihood of a transaction being fraudulent. The model classifies a transaction as fraudulent if this probability equals or exceeds the threshold of 0.5.

This decision boundary is derived through an iterative process of stochastic gradient ascent, an optimization technique that adapts the weights ω to maximize the likelihood of correctly identifying fraud cases, while maintaining computational efficiency essential for handling large datasets. This methodological approach highlights Logistic Regression's capability to discern intricate patterns in transaction data, crucial for effective fraud detection.

[16] John Awoyemi, Adebayo Olusola Adetunmbi, and Samuel Adebayo Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis," in *2017 International Conference on Computing Networking and Informatics (ICCNI)* (2017): 1-9.

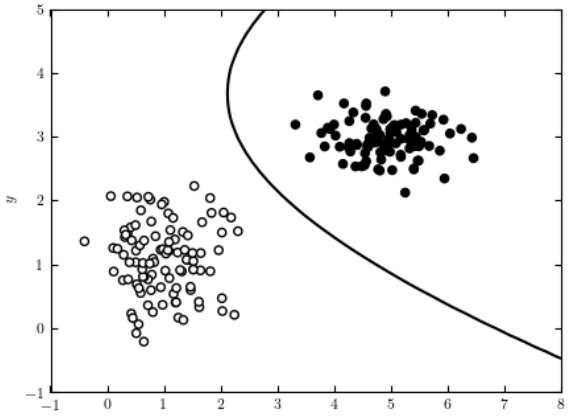
[17] Wensi Yang et al., "FFD: A Federated Learning Based Method for Credit Card Fraud Detection," in *Proceedings of the 2019 BigData Congress* [Services Society], 2019.

[18] E. Ileberi, Y. Sun, and Z. Wang, "A machine learning based credit card fraud detection using the GA algorithm for feature selection," Journal of Big Data 9, no. 24 (2022): https://doi.org/10.1186/s40537-022-00573-8.

[19] Lin and Jiang, "Credit Card Fraud Detection."

B. Gaussian Naive Bayes

The Naive Bayes classifier operates under the assumption of feature independence given the class label, a principle encapsulated in Bayes' theorem. The Gaussian Naive Bayes variant specifically deals with continuous data and assumes that the features follow a normal (Gaussian) distribution.



In the mathematical formulation of Gaussian Naive Bayes, the likelihood of observing a feature x_i given a classic c is modeled by the Gaussian (normal) distribution. For a feature x_i with a class c, the probability density is given by:

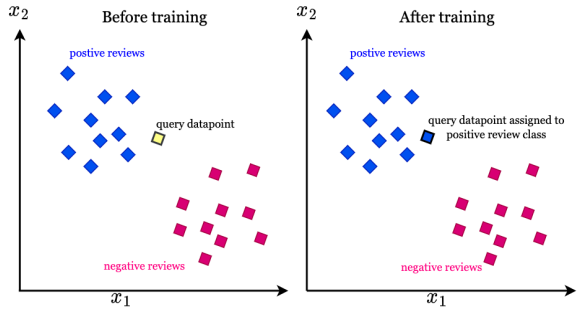
$$P(x_i|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} exp\left(-\frac{(x_i-\mu_c)^2}{2\sigma_c^2}\right)$$

Where μ_c is the mean of the feature for class c, and σ_c^2 is the variance of the feature for class c. The classifier then computes the product of these probabilities across all features and multiplies by the prior probability of the class P(c), yielding a posterior probability that the data belongs to class c. The class with the highest posterior probability is the predicted class for the given input.

The analytical strength of Gaussian Naive Bayes in fraud detection lies in its simplicity and effectiveness, especially when the assumption of feature independence holds reasonably true. This simplicity allows for fast computation and straightforward interpretation of the likelihoods that inform the model's predictions.

C. K-Nearest Neighbors

The K-Nearest Neighbors (KNN) algorithm is a quintessential example of instance-based learning, where classification is achieved by a simple yet effective majority vote among the nearest neighbors of a given data point. It is defined by two key components: the number of neighbors 'k' and the metric used to measure distance between instances.



Mathematically, the distance between two points in the feature space, x_i and x_j, is often computed using the Euclidean distance, represented as:

$$D_{ij} = \sqrt{\Sigma_{k=1}^{n}(X_{ik} - X_{jk})^2}$$

Where n is the number of dimensions in the feature space. The algorithm selects the 'k' closest points to the query instance

based on this distance metric, and a vote is conducted to determine the most frequent label (class) among these points. This makes KNN particularly suited for datasets where proximity in feature space implies similarity in class membership.

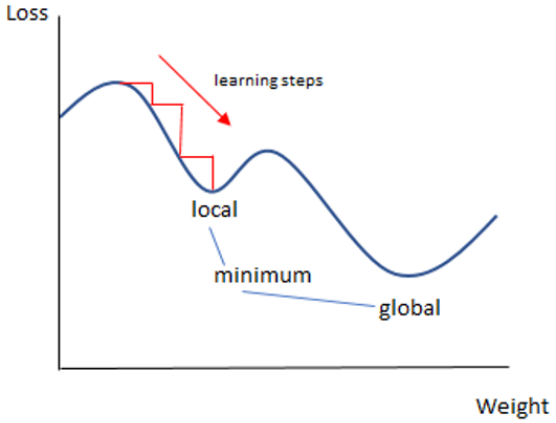
In the context of fraud detection, KNN's reliance on local structures and patterns allows it to detect anomalies that deviate from the established norm of transactional behavior. The selection of 'k' is critical, as it determines the granularity of the neighborhood considered for the voting process, affecting the model's sensitivity and specificity to fraudulent activities.

D. Stochastic Gradient Descent Classifier

The SGDClassifier embodies a pragmatic approach to linear classification, particularly in scenarios like credit card fraud detection, where it's crucial to process large volumes of data efficiently. At its core, SGD employs a linear function to establish a decision boundary:

$$\widehat{y} = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \ldots + \omega_n x_n,$$

where x_i represents the feature values, and ω_i signifies the model parameters.



SGD stands apart by updating model parameters, ω, incrementally using each training sample one at a time, which significantly reduces the computational costs. This updating process is guided by the gradient of the loss function, reflecting the discrepancy between predicted and actual values, adjusted by a learning rate. The iterative nature of this optimization allows SGD to fine-tune the decision boundary, enhancing the model's ability to distinguish between fraudulent and non-fraudulent transactions. The convergence to optimal parameters is achieved by traversing the gradient in a stepwise fashion, making it a powerful tool in fraud detection with the advantage of scalability and speed.

E. Light Gradient-Boosting Machine

LightGBM operates on the principles of gradient boosting mechanisms, characterized by its distinctive leaf-wise tree growth algorithm, which optimizes traditional gradient boosting methods for both efficiency and efficacy. LightGBM can be defined as:

$$L(\phi) = \Sigma_{i=1}^{n} l(y_i, \widehat{y}_i(\phi)) + \Sigma_{k=1}^{k} \Omega(f_k),$$

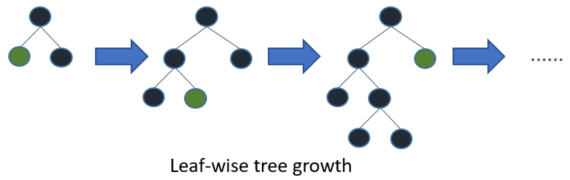
where $L(\phi)$ is the loss function that LightGBM aims to minimize,$l(y_i, \widehat{y}_i(\phi))$ represents the loss due to the difference between the actual target y_i and the prediction $\widehat{y}_i$ for each instance i, and $\Omega(f_k)$ is the complexity of the tree f_k. This complexity term helps prevent overfitting by penalizing the model for the number of leaves and the magnitude of leaf values.

Its objective function can be delineated as follows:

$$Obj = -\Sigma_{i-1}^{n}[y_i \cdot log(\widehat{y_i}) + (1 - y_i) \cdot log(1 - \widehat{y_i})] + \lambda \cdot \Sigma_{j=1}^{T} \omega_j^2$$

Here, the objective, *Obj*, is a composite function comprising a binary cross-entropy loss term for classification, and a regularization term where λ denotes the regularization coefficient, T represents the total number of leaves, and ω_j stands for

the value of each leaf. In the context of fraud detection, the algorithm converges to an optimal solution that accentuates the minority class's predictive signals through iterative refinement of the decision trees, thereby enhancing the detection rate of fraudulent transactions.

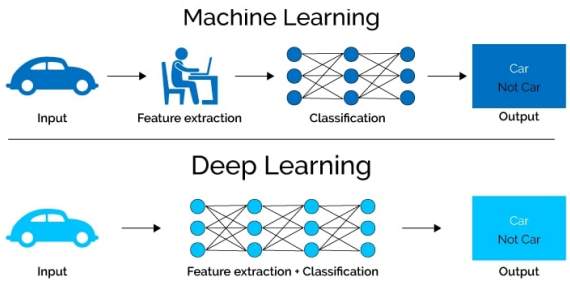


The core algorithmic enhancement in LightGBM is its Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB), which address the issues of data sparsity and dimensionality, pivotal in fraud analytics. GOSS retains the data instances with large gradients, thereby honing in on the harder-to-learn, often more critical, examples such as fraudulent transactions. Concurrently, EFB reduces feature dimensions without significant loss of information, leveraging the sparsity of the data.

This nuanced and computational resource-conscious approach facilitates the handling of large-scale data inherent in fraud detection mechanisms, providing a robust, scalable, and highly accurate predictive modeling framework.

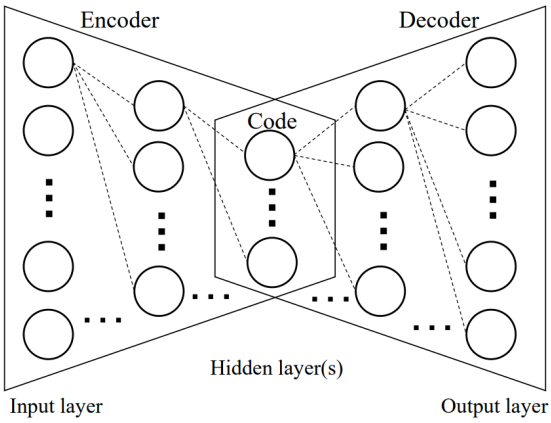
F. Deep Learning: Autoencoders

Deep learning is at the cutting edge of modern computational intelligence, distinguished by its ability to directly assimilate raw data. Instead of relying on expert feature extraction, this sophisticated subclass of machine learning cultivates the ability to independently discover relevant features through deep, layered architectures. Deep learning's success stems from its versatility, which automates the learning process to discover nuanced patterns hidden within vast datasets. Deep learning in fraud detection examines transactional activities for abnormalities that indicate fraud. This approach is critical in automating the detection process as it allows the system to learn and adapt to the dynamic nature of fraudulent activities without the need for manual feature engineering. Its use in our study demonstrates a commitment to utilizing cutting-edge technologies to improve the security of financial transactions.[20]



Autoencoders, a subset of deep learning models, are designed for unsupervised learning tasks, primarily for dimensionality reduction or feature learning. They consist of an encoder that compresses the input and a decoder that reconstructs the input from the compressed form, attempting to minimize the difference between the original input and the reconstruction. This process forces the autoencoder to capture the most salient features of the data.

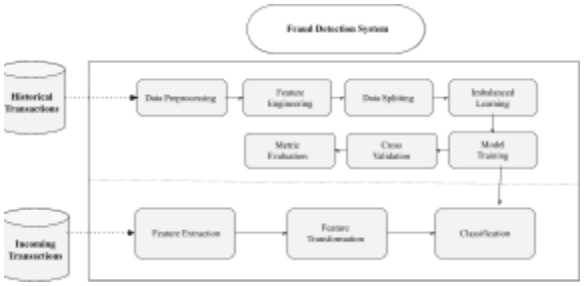
[20] Mohammadhossein Reshadi, "Anomaly Detection Using Deep Learning" (Master's thesis, University of Alberta, Spring 2021), https://doi.org/10.7939/r3-4trf-cy91.



In the context of credit card fraud detection, autoencoders can be particularly useful for anomaly detection. By training on normal transaction data, the model learns a representation of typical transaction profiles. Transactions that significantly deviate from this representation can be flagged as potential fraud.

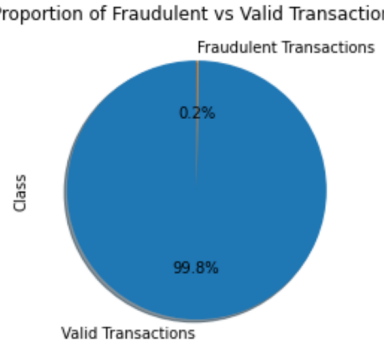
The mathematical function for autoencoders can be expressed as: $y = \sigma(Wx + b)$. Here, σ is a nonlinear activation function such as a sigmoid or ReLU, W represents the weights of the neural network, x is the input vector, b is the bias, and y is the output which aims to approximate x. During training, the weights W and biases b are adjusted to minimize the reconstruction error, often measured by the mean squared error between the input and output. For fraud detection, the learned compressed representation can highlight unusual patterns in the data, which can then be analyzed to identify transactions that are likely to be fraudulent.

IV. Methodology



A. Data Preprocessing

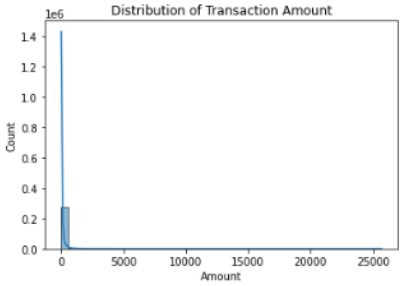
The dataset utilized in this study was provided by the Machine Learning Group at Université Libre de Bruxelles, encompassing transactions from European cardholders during September 2013. Comprising 284,807 transactions over two days, it presents a severe class imbalance with only 492 fraudulent cases, a mere 0.172% of the data. The dataset has been anonymized for confidentiality, containing 30 numerical input variables derived from a Principal Component Analysis transformation, thus omitting original feature details and certain background information to maintain data privacy.[21]

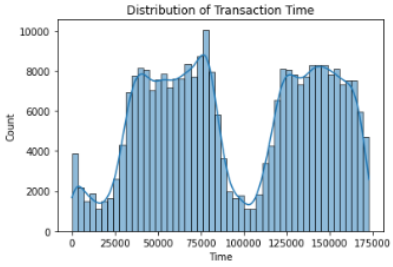


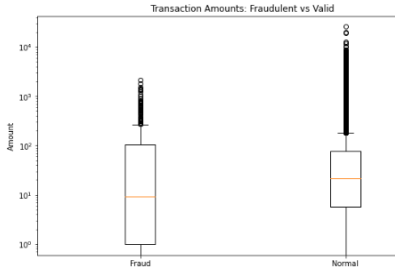
The histogram of transaction amounts showcases a heavily right-skewed distribution, with a preponderance of smaller transactions and a tail indicative of infrequent but substantial outliers. The distribution over time exhibits a bimodal pattern, potentially correlating with diurnal rhythms in spending behavior. The

[21] Yang et al., "FFD."

comparative boxplot analysis delineates stark contrasts between fraudulent and legitimate transactions, where the former displays a wider interquartile range and more pronounced outliers, suggesting atypical transaction behaviors associated with fraud. Utilizing a logarithmic scale enhances the visibility of these distinctions, particularly for fraudulent transactions, underscoring the necessity of advanced analytical techniques to identify such anomalies. This initial analytical phase lays the groundwork for more nuanced feature engineering and model development.

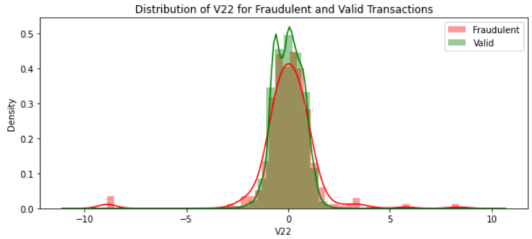


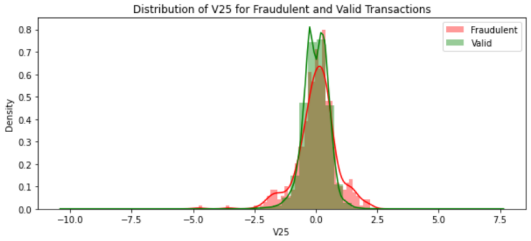




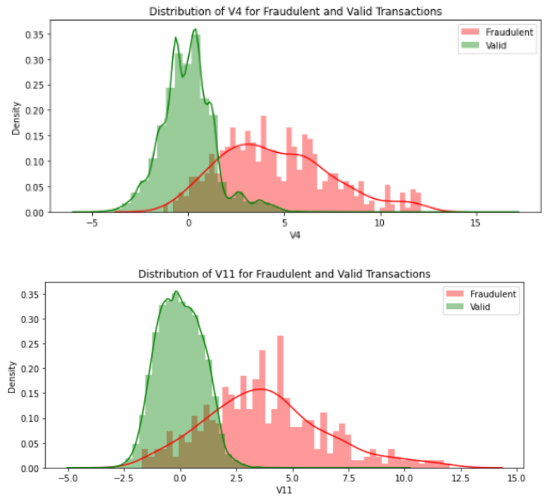
Feature distribution analysis reveals critical insights. Features such as V14 and V17 emerge as potential key indicators of fraud due to their distinctive distributions, signaling their value in predictive modeling. Conversely, features like V22 and V25 exhibit overlapping distributions for both transaction classes, suggesting a reduced discriminative power when considered in isolation.

The presence of outliers across the spectrum of features underscores the complexity inherent in discerning fraudulent from legitimate activity. While some outliers may signify fraudulent transactions, others might represent uncommon yet genuine behaviors, necessitating nuanced preprocessing techniques to retain valuable information while mitigating noise.





Skewed distributions prevalent in the dataset, especially within fraudulent transactions, can pose challenges to modeling. These skewed features may require strategic transformations to ensure model robustness and accuracy.



Notably, features like V4 and V11 show promise due to their association with higher values in fraudulent transactions, while features such as V12, V14, V17, and V18, with their lower values linked to fraud, could be pivotal in enhancing detection capabilities. Similarly, V10 and V16 are highlighted for their distinctive patterns in fraudulent cases, underscoring their potential utility in fraud detection algorithms.

B. Feature Engineering

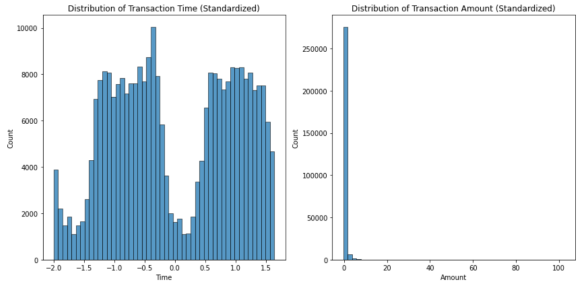
In the nuanced domain of fraud detection, the standardization of features is not merely a procedural step but a cornerstone of effective model training. Our dataset, a confluence of time-based, transactional, and PCA-transformed variables, presents a diverse range of scales and distributions. Standardization, therefore, becomes imperative to harmonize these disparate scales, ensuring that each feature contributes equitably to the model's learning process.

Standardization transforms the features of our dataset to have a mean (μ) of zero and a standard deviation (σ) of one. This process is crucial for models sensitive to feature scales, like logistic regression and K-nearest neighbors.

The standardization of a feature X is computed as: $X_{standardized} = \frac{X-\mu}{\sigma}$, where $X_{standardized}$ is the standardized feature, μ is the mean feature, σ is the standard deviation of the feature.

In our fraud detection model, the 'Time' and 'Amount' features undergo standardization. Unlike the PCA-transformed variables, these features retain their original scale, necessitating their normalization to prevent skewed model training and biased predictions.

Standardization mitigates the risk of one feature dominating the model due to its larger scale, a scenario that could obscure significant patterns in other features. By ensuring that each feature contributes on a comparable scale, standardization enhances the model's ability to identify subtle yet critical indicators of fraudulent transactions.



This methodological approach underscores the importance of feature engineering in fraud detection. It highlights how standardization, as a transformative step, equips the model to discern complex patterns in transaction data, a fundamental aspect of effective fraud detection.

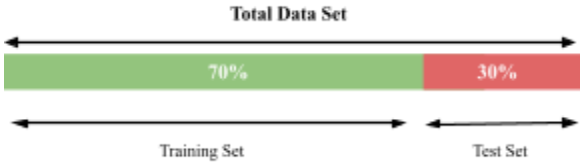
C. Data Splitting

Data splitting is a fundamental step in the development of predictive models, particularly in fraud detection. It involves dividing the dataset into distinct sets: training and testing. This division is crucial for evaluating the model's performance

objectively and ensuring its generalizability to unseen data.

- Model Training: The training set is used to train the machine learning models. It's where the models learn to identify patterns and make decisions.
- Model Evaluation: The testing set, unseen by the model during training, is used for evaluation. It provides an unbiased assessment of the model's effectiveness in real-world scenarios, particularly in detecting fraudulent transactions

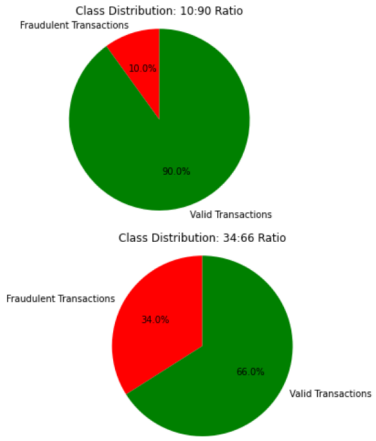
Our approach involves a randomized split of the dataset, ensuring that both the training and testing sets are representative of the overall data distribution. In our examination, we opt for a 70:30 split ratio, meaning 70% of the data is used for training and the remaining 30% for testing.



We employ stratified sampling to split the data. This method ensures that the proportion of the classes (fraudulent and non-fraudulent transactions) is consistent across both the training and testing sets. Stratified sampling is crucial in our context due to the class imbalance inherent in fraud detection datasets. It prevents scenarios where the minority class is underrepresented, particularly in the test set.

D. Imbalanced Learning

In the domain of fraud detection, the challenge of class imbalance is paramount, and our approach to addressing this issue is informed by significant research findings. Drawing from recent studies which advocates for stratified sampling to under-sample legitimate records to meaningful numbers, experimenting with 50:50, 10:90, and 1:99 distributions of fraud to legitimate cases. The study reports that a 10:90 distribution yields the best performance as it closely mirrors the real distribution of frauds and legitimate transactions.[22] Additionally, the study provides insights that guide our decision to explore a 34:66 class distribution. Our methodology adopts a hybrid approach that combines oversampling and under-sampling techniques, allowing for a comparative analysis of model performance across these varied class distributions.



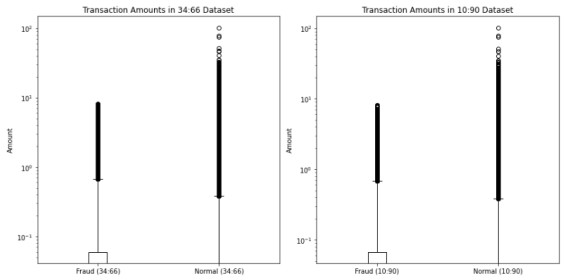
In our initial dataset comprising 284,807 transactions, a severe class imbalance is evident, with only 492 cases being fraudulent, constituting a mere 0.172% of the total transactions. This disproportion poses a challenge for effective model training and fraud detection. To address this, we implemented resampling techniques, resulting in two distinct datasets with altered class distributions. The first, adhering to a 10:90 fraud-to-legitimate transaction ratio, expanded the fraudulent cases to 22,745 out of 250,195 total transactions. The second, following a 34:66 ratio, further increased the prevalence of fraud cases to 77,333 out of 304,782 total

[22] Awoyemi, Adetunmbi, and Oluwadare, "Credit card fraud detection," 1.

transactions. These resampled datasets stand in stark contrast to the original data in terms of the volume and proportion of fraudulent transactions. By artificially elevating the incidence of fraud in these training sets, we aim to provide our models with a more balanced and representative sample of both fraudulent and non-fraudulent transactions. This approach enhances the model's ability to learn the subtleties of fraudulent behavior, a crucial step in developing an effective fraud detection system.

At the core of our strategy is the Synthetic Minority Over-sampling Technique (SMOTE), which is instrumental in addressing the imbalance by generating synthetic samples from the minority class. SMOTE operates by randomly selecting points along the line segments joining any or all of the k minority class nearest neighbors. For a minority class sample x_i, SMOTE calculates the difference δ between x_i and its nearest neighbor x_{nn}. A synthetic sample x_{new} is then created as follows: $x_{new} = x_i + \lambda \cdot \delta$, where λ is a random number between 0 and 1, and $\delta = x_{nn} - x_i$. This technique is particularly effective in our fraud detection dataset, where fraudulent transactions are significantly underrepresented.

In conjunction with SMOTE, our approach also involves under-sampling the majority class. This is crucial to prevent the model from being overwhelmed by the volume of legitimate transactions. By first applying SMOTE to augment the minority class and then randomly under-sampling the majority class, we achieve the desired class distributions of 34:66 and 10:90.



The efficacy of this hybrid approach is evaluated through a comparison of model performance on both the 34:66 and 10:90 distributions. Key metrics provide a comprehensive view of the model's ability to accurately and robustly detect fraudulent activities. While the 10:90 distribution is anticipated to yield optimal results based on its alignment with real-world data distribution, the potential advantages of the 34:66 distribution are also rigorously assessed.

E. Model Training

In our fraud detection project, the model training phase was particularly influenced by the need to balance computational efficiency with the effectiveness of our predictive models. Given the substantial size and complexity of our dataset, especially post-resampling, the choice and configuration of models were paramount. Our focus was on ensuring that the training process was feasible within our computational limits without significantly compromising the ability to detect fraud effectively.

We selected a suite of 5 machine learning models, each with its unique strengths and suitability for fraud detection. These models included Logistic Regression, K-Nearest Neighbors (KNN), Gaussian Naive Bayes, SGDClassifier, and LightGBM. The choice of each model was driven by its proven effectiveness in classification tasks and its ability to handle the peculiarities of imbalanced data, a characteristic challenge in fraud detection.

For KNN, we set the number of neighbors to five, optimizing for both accuracy and computational efficiency. The KNN model's reliance on the proximity of data points made it particularly suited for identifying subtle patterns associated with fraudulent transactions.

A key adaptation to our computational constraints was the use of the SGDClassifier as a stand-in for a traditional Support Vector Machine (SVM). SVMs are renowned for their classification capabilities but are computationally demanding with large datasets. The SGDClassifier, configured to approximate an SVM using a stochastic gradient descent, offered a more computationally viable alternative. This choice allowed us to leverage the benefits of an SVM-like model while managing computational resources effectively. The SGDClassifier was initialized with a hinge loss, effectively creating a linear SVM within a pipeline incorporating StandardScaler. This configuration not only standardized the features but also ensured computational efficiency, a critical factor given the extensive nature of our data.

Gaussian Naive Bayes was implemented using the GaussianNB class, capitalizing on its proficiency in handling high-dimensional datasets and its effectiveness in probabilistic classification.

LightGBM, employed for its efficiency in large, imbalanced datasets, was initialized using the LightGBM Classifier. We trained the LightGBM model to predict class labels and probabilities on the test set, leveraging its ability to manage complex data structures.

The implementation of a deep learning model, specifically an Autoencoder, marks a specialized approach within our array of predictive techniques. The Autoencoder, designed as a neural network for unsupervised learning using TensorFlow’s Keras API, stands out from traditional machine learning models due to its unique architecture and training process.

It includes an input layer with 30 nodes, corresponding to the number of features in our dataset, followed by two hidden layers with 128 and 64 nodes, respectively. Each layer incorporates *relu* activation and is regularized using L2 regularization to prevent overfitting. The model also integrates dropout layers, further enhancing its robustness against overfitting.

The network culminates in an output layer with a single node using a sigmoid activation function, tailored for binary classification. This design marks a departure from the typical Autoencoder architecture, focusing instead on directly classifying transactions as fraudulent or legitimate.

We compile this neural network model with the Adam optimizer, using a learning rate of 0.0001, and set the loss function to 'binary_crossentropy', a standard choice for binary classification tasks. The model's training spans 100 epochs with a batch size of 256. To address the class imbalance inherent in our data, we apply class weights, assigning a higher weight to the minority class. This is complemented by an early stopping mechanism, monitoring the validation loss and terminating the training process if no improvement is observed, thereby optimizing the training duration and preventing overfitting.

The training involves both the 34:66 and 10:90 class distribution datasets, ensuring the model is robustly tested against different scenarios of class imbalance. Validation during training is conducted on the test set to monitor and evaluate the model's performance continuously.

Upon training completion, the model's effectiveness is assessed using the test data, with the predictions being binary classifications determined by a threshold on the output probabilities. This approach enables us to directly classify transactions as

fraudulent or non-fraudulent, leveraging the neural network's capability to learn complex, non-linear relationships in high-dimensional data.

The deployment of this deep learning model in our fraud detection system underscores our commitment to leveraging advanced techniques for enhanced performance. While computationally intensive, this model's nuanced feature representation and classification capabilities significantly augment our system's ability to detect fraudulent transactions accurately and efficiently.

In all instances, the training process was duplicated for each class distribution - 34:66 and 10:90. This dual training approach allowed us to comprehensively assess each model's performance across different scenarios, ensuring robustness and adaptability in our fraud detection system. By applying this consistent methodology, we could evaluate the effectiveness of each model in detecting fraudulent activities, ensuring a thorough and rigorous analysis.

Managing the computational load was a constant consideration throughout the training process. We optimized hyperparameters to balance model complexity with training duration and monitored resource utilization meticulously. The training of each model was an iterative process, with hyperparameter tuning and configuration adjustments made to ensure optimal performance within our computational environment.

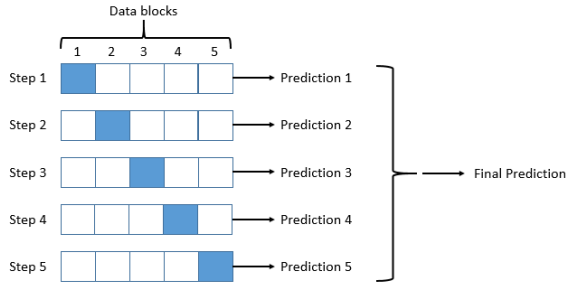
In summary, our approach to model training was a careful exercise in balancing the computational feasibility with the necessity for effective and robust fraud detection models. This pragmatic and strategic selection and configuration of models underlined our commitment to developing practical and efficient solutions in fraud detection, capable of accurately identifying fraudulent transactions within the constraints of our computational resources.

F. Cross-Validation

Cross-validation played a critical role in ensuring the robustness and generalizability of our machine learning models, which included Logistic Regression, Gaussian Naive Bayes, SGDClassifier, LightGBM, and K-Nearest Neighbors (KNN). To this end, we implemented a Stratified K-Fold cross-validation strategy, typically with 5 folds. This method was meticulously applied to both the 34:66 and 10:90 class distributions across different models to comprehensively evaluate performance across varied data scenarios.

The Stratified K-Fold approach was chosen for its effectiveness in maintaining the proportion of each class within each fold, a crucial factor given the imbalanced nature of our dataset. For each model, the data was partitioned into five subsets, with each subset serving as a representative of the whole. Four of these subsets were used for training, and the fifth for validation. This process was repeated five times, with each subset used once as the validation set, thereby providing a thorough evaluation of the models.

However, we adapted this strategy for the KNN model due to computational constraints. The KNN algorithm, known for its computational intensity especially with larger datasets and a higher number of neighbors, necessitated a reduction in the number of folds to three. This adaptation allowed us to balance the benefits of cross-validation, such as reducing variance in model evaluation, with the practical limitations of our computational resources.



In contrast to our traditional machine learning models, the deep learning model in our project, a neural network-based Autoencoder, did not undergo the typical cross-validation process. Due to the computational demands and complexity of training deep learning models, we focused on a training-validation split supplemented by early stopping and class weights. This approach was designed to ensure the model's robustness and ability to generalize effectively on unseen data.

Through the strategic implementation of cross-validation for our traditional models, and the tailored training approach for the deep learning model, we sought to rigorously test and validate our models across different subsets of data.

G. Performance Metrics & Results

To assess the predictive models' effectiveness, a set of performance metrics was carefully chosen. These indicators are critical in determining how successfully each model detects fraudulent transactions, balancing the need for precision in detecting genuine frauds with the broader scope of all transactions.

Accuracy represents the ratio of correctly predicted observations to the total observations. It is the most straightforward measure of a model's overall performance.

$$Accuracy = \frac{TP + TN}{TP+FP+FN+TN}$$

In the realm of fraud detection, accuracy provides insight into the proportion of transactions that were correctly classified. However, given the class imbalance typical in fraud datasets, accuracy might not fully capture a model's effectiveness, as it can be disproportionately influenced by the majority class.

Sensitivity, or recall, is the metric that quantifies the model's ability to correctly identify actual positives, which in our case are the fraudulent transactions.

$$Recall = \frac{TP}{TP+FN}$$

For fraud detection systems, recall is particularly crucial. It reflects the model's capability to catch fraudulent activity, which is essential — a fraudulent transaction that goes undetected could result in significant financial loss.

Precision measures the quality of the positive predictions made by the model. It is the proportion of positive identifications that were actually correct, and it is formulated as:

$$Precision = \frac{TP}{TP+FP}$$

In the context of fraud detection, precision is significant because it indicates the reliability of the model's fraud alerts. High precision means that when a model predicts a transaction as fraudulent, it is very likely to be so, minimizing the cost and inconvenience of false alarms.

The F2 score is an essential metric in the context of fraud detection due to the asymmetric nature of the costs associated with false positives and false negatives. The disproportionate impact of these errors necessitates the use of an evaluation metric that can adequately reflect the severity of false negatives — that is, the failure to detect fraud.

False positives, while carrying administrative costs and potentially affecting customer satisfaction, are less impactful than

false negatives, which can result in substantial financial losses equivalent to the value of the undetected fraudulent transaction. As such, our evaluation strategy must prioritize the detection of fraud even if it means accepting a higher number of false positives.

The F2 score is specifically designed to weigh recall more heavily than precision. This is expressed in its formula:

$$F2 = \frac{(1+2^2)\cdot(Precision\cdot Recall)}{(2^2\cdot Precision)+Recall}$$

This formulation assigns greater importance to recall, the ability of the model to identify all relevant instances of fraud. A higher F2 score indicates a model that effectively captures most fraudulent transactions, which aligns with the primary goal of minimizing the costliest error: failing to flag a fraudulent transaction (a false negative).

By opting for the F2 score over other metrics like accuracy or the F1 score, we ensure that our models are fine-tuned to prioritize the detection of fraud. This aligns with the insights from Hand, Whitrow, Adams, Juszczak, and Weston, reflecting a deliberate choice to address the cost-sensitive nature of fraud detection and highlighting our commitment to minimizing the financial risks associated with undetected fraud. The F2 score thus becomes a key indicator of our model's performance, emphasizing our focus on a metric that is closely aligned with the business objectives and the real-world implications of credit card fraud detection.[23].

In addition to the standard metrics of accuracy, precision, and recall, our fraud detection models are evaluated using more nuanced metrics that provide a deeper insight into their performance, particularly in the context of an imbalanced dataset. These include the Matthews Correlation Coefficient (MCC), the Balanced Classification Rate (BCR), and the Receiver Operating Characteristic Area Under the Curve (ROC AUC).

The MCC is a reliable statistical rate that produces a high score only if the prediction obtained good results in all of the four confusion matrix categories (true positives, false negatives, true negatives, and false positives), proportionally both to the size of positive elements and the size of negative elements in the dataset.

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

The MCC is essentially a correlation coefficient between the observed and predicted binary classifications; it returns a value between -1 and +1. A coefficient of +1 represents a perfect prediction, 0 is no better than random prediction, and -1 indicates total disagreement between prediction and observation. In fraud detection, a high MCC indicates that both classes are being predicted well, despite any class imbalance.

The BCR is the average of the recall and the true negative rate (specificity). It is particularly useful when dealing with imbalanced datasets as it takes into account the rate of both classes:

$$BCR = \frac{Recall+Specificity}{2}$$

$$Specificity = \frac{TN}{TN+FP}$$

This metric ensures that both the majority and minority classes are being considered equally, and it is particularly meaningful for fraud detection since it balances the focus between identifying frauds (recall) and avoiding false alarms (specificity).

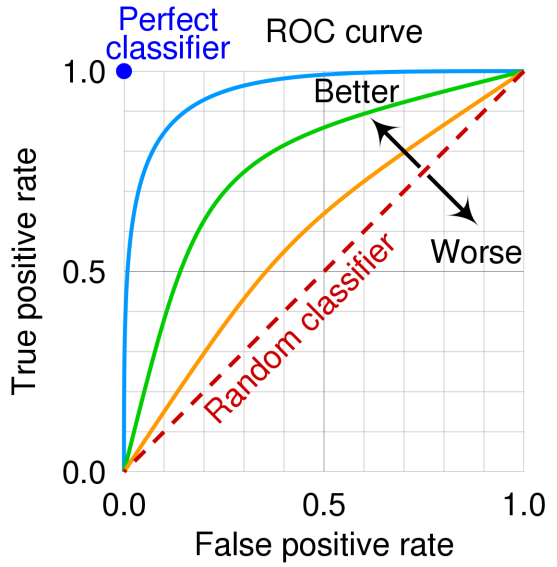
[23] Correa Bahnsen, "Feature Engineering Strategies."

The ROC AUC is a performance measurement for classification problems at various threshold settings. The ROC is a probability curve, and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes.

The higher the AUC, the better the model is at predicting 0s as 0s and 1s as 1s. An AUC score of 0.5 suggests no discrimination ability, equivalent to random chance, while a score of 1 indicates perfect discrimination:

$$\text{ROC AUC} \in [0.5,1]$$

In fraud detection, the ROC AUC metric is particularly useful because it provides a single measure of performance irrespective of the classification threshold, which can be crucial when the cost of misclassification varies between classes.



Each of these metrics – MCC, BCR, and ROC AUC – provides a different perspective on the model's performance and, when used together, offer a comprehensive evaluation of the model's capabilities in the unbalanced landscape of fraud detection. These metrics were carefully selected for their ability to provide a balanced view of performance across both classes, ensuring that our models are rigorously evaluated and refined for optimal performance in detecting fraudulent transactions.

In the methodical pursuit of constructing an effective fraud detection system, we employed an array of metrics to rigorously evaluate our suite of machine learning and deep learning models. The evaluation framework was bifurcated into two distinct phases: pre-cross-validation and post-cross-validation assessments, augmented by specialized metrics for the deep learning model.

Prior to cross-validation, we scrutinized the models against a comprehensive set of metrics gleaned from the confusion matrix. This initial evaluation involved computing accuracy, sensitivity (recall), precision, and the F2 score. We also calculated the Matthews Correlation Coefficient (MCC) and Balanced Classification Rate (BCR), which are particularly insightful in the realm of an imbalanced dataset like ours. Furthermore, we assessed the models' discriminatory capabilities via the Receiver Operating Characteristic Area Under the Curve (ROC AUC), providing us with a nuanced view of each model's ability to differentiate between the fraudulent and non-fraudulent transactions.

For the deep learning model, which was an Autoencoder configured for binary classification, we included additional evaluation metrics. We tracked the ROC AUC, recall, precision, accuracy, and the F2 score, which are standard for binary classifiers. Additionally, we monitored the model loss over epochs, a metric that illustrates the model's learning trajectory and convergence behavior during training.

Following the implementation of cross-validation, the evaluation metrics were streamlined to focus on accuracy, recall,

precision, the F2 score, and the MCC. This post-cross-validation phase was crucial for affirming the models' stability and their performance across different data subsets, ensuring that the final model selection was robust against overfitting and well-suited for practical deployment.

The holistic view of the evaluation strategy encompassed both pre- and post-validation metrics, allowing for a dynamic and iterative process of model refinement. This dual-phase assessment, coupled with the inclusion of a diverse set of performance indicators, fortified our approach to unveiling a fraud detection system that is not only statistically rigorous but also pragmatic and attuned to the operational nuances of credit card fraud detection.

In search of the optimal model for credit card fraud detection, our study presents a rigorous performance evaluation of several classifiers, weighed against the backdrop of efficacy and operational efficiency. The contenders—Naive Bayes, Logistic Regression, K-Nearest Neighbors, SGDClassifier, LightGBM, and an Autoencoder model—underwent a comprehensive assessment using a variety of metrics to gauge their ability to identify fraudulent transactions accurately and efficiently.

Metrics	Classifiers		
	Naive Bayes	*Logistic Regression*	*K-Nearest Neighbour*
Accuracy	0.977	0.992	0.998
Sensitivity	0.857	0.898	0.867
Specificity	0.977	0.992	0.998
Precision	0.060	0.162	0.483
F2 Score	0.237	0.470	0.748
Matthews Correlation Coefficient	+0.225	+0.379	+0.646
Balanced Classification Rate	0.917	0.945	0.933
ROC AUC	0.962	0.979	0.938

Table 1. 34:66 Metrics

Metrics	Classifiers		
	Naive Bayes	*Logistic Regression*	*K-Nearest Neighbour*
Accuracy	0.945	0.965	0.998
Sensitivity	0.848	0.884	1.000
Precision	0.928	0.974	0.993
F2 Score	0.863	0.900	0.998
Matthews Correlation Coefficient	+0.851	+0.906	+0.996

Table 2. 34:66 Metrics Cross-Validated

Metrics	Classifiers		
	Naive Bayes	*Logistic Regression*	*K-Nearest Neighbour*
Accuracy	0.977	0.998	0.998
Sensitivity	0.857	0.867	0.867
Specificity	0.978	0.998	0.998
Precision	0.063	0.438	0.545
F2 Score	0.243	0.725	0.776
Matthews Correlation Coefficient	+0.228	+0.616	+0.687
Balanced Classification Rate	0.918	0.933	0.933
ROC AUC	0.963	0.979	0.938

Table 3. 10:90 Metrics

Metrics	Classifiers		
	Naive Bayes	*Logistic Regression*	*K-Nearest Neighbour*
Accuracy	0.967	0.985	0.998
Sensitivity	0.849	0.859	0.999
Precision	0.798	0.976	0.983
F2 Score	0.838	0.880	0.996
Matthews Correlation Coefficient	+0.805	+0.908	+0.990

Table 4. 10:90 Metrics Cross-Validated

The Naive Bayes classifier, traditionally valued for its simplicity and speed, was observed to have a propensity for higher false positives. This model, while computationally light, could induce substantial operational costs due to a potential increase in customer service interventions and administrative handling of misclassified legitimate transactions. Its application in a high-volume, real-world scenario may be untenable, given these considerations.

Logistic Regression exhibited a commendable balance in its performance metrics, presenting itself as a viable model. Its scalability and relatively modest computational demands render it a practical choice for processing large volumes of transactions. However, the model's performance was outshone by more

sophisticated algorithms that demonstrated superior precision and recall.

K-Nearest Neighbors, with its impressive sensitivity and F2 scores, indicates a strong potential for catching fraudulent activity. Nonetheless, the computational intensity required for this model to process vast datasets poses significant challenges, potentially limiting its feasibility in a large-scale, real-time fraud detection system.

Metrics	Classifiers		
	SGDClassifier	*LightGBM*	*Autoencoder*
Accuracy	0.994	0.999	0.999
Sensitivity	0.867	0.847	0.867
Specificity	0.994	0.999	N/A
Precision	0.198	0.679	0.599
F2 Score	0.517	0.812	0.796
Matthews Correlation Coefficient	+0.412	+0.768	N/A
Balanced Classification Rate	0.931	0.923	N/A
ROC AUC	0.971	0.988	0.930

Table 5. 34:66 Metrics

Metrics	Classifiers	
	SGDClassifier	*LightGBM*
Accuracy	0.963	0.999
Sensitivity	0.876	0.999
Precision	0.978	0.997
F2 Score	0.895	0.999
Matthews Correlation Coefficient	+0.903	+0.998

Table 6. 34:66 Metrics Cross-Validated

The SGDClassifier emerged as a robust option, showing a notable balance between detecting fraud and minimizing false alerts. Coupled with its computational efficiency, the SGDClassifier stands out as a strong candidate for real-world application, where processing speed is crucial.

Metrics	Classifiers		
	SGDClassifier	*LightGBM*	*Autoencoder*
Accuracy	0.999	0.999	0.999
Sensitivity	0.806	0.847	0.867
Specificity	0.999	0.999	N/A
Precision	0.745	0.798	0.654
F2 Score	0.793	0.837	0.814
Matthews Correlation Coefficient	+0.775	+0.822	N/A
Balanced Classification Rate	0.903	0.923	N/A
ROC AUC	0.977	0.985	0.931

Table 7. 10:90 Metrics

Metrics	Classifiers	
	SGDClassifier	*LightVBM*
Accuracy	0.984	0.999
Sensitivity	0.826	0.994
Precision	0.993	0.996
F2 Score	0.854	0.995
Matthews Correlation Coefficient	+0.897	+0.995

Table 8. 10:90 Metrics Cross-Validated

The LightGBM model was particularly notable for its exceptional performance post-cross-validation, showcasing high scores in all key metrics, including the highest F2 score and MCC among the machine learning models. LightGBM's efficiency in handling large datasets, along with its impressive ability to differentiate between fraudulent and legitimate transactions, positions it as an exemplary model in both efficacy and efficiency. Its implementation in a business environment is bolstered by its low latency and high throughput capabilities, making it well-suited for high-stakes, real-time decision-making.

Metrics	Classifiers		
	SGDClassifier	*LightGBM*	*Autoencoder*
True Positive	56,837	56,843	56,819
False Positive	27	21	45
True Negative	19	15	13
False Negative	79	83	85

Table 10. 10:90 Confusion Matrix

Metrics	Classifiers		
	SGDClassifier	*LightGBM*	*Autoencoder*
True Positive	56,837	56,843	56,819
False Positive	27	21	45
True Negative	19	15	13
False Negative	79	83	85

Table 10. 10:90 Confusion Matrix

Metrics	Classifiers		
	Naive Bayes	*Logistic Regression*	*K-Nearest Neighbour*
True Positive	55,567	56,408	56,773
False Positive	1,297	456	91
True Negative	14	10	13
False Negative	84	88	85

Table 11. 34:66 Confusion Matrix

Metrics	Classifiers		
	Naive Bayes	*Logistic Regression*	*K-Nearest Neighbour*
True Positive	55,608	56,755	56,793
False Positive	1,256	109	71
True Negative	14	13	13
False Negative	84	85	85

Table 12. 10:90 Confusion Matrix

The deep learning Autoencoder demonstrated promising results, rivaling the traditional models with high accuracy and sensitivity. Despite the lack of cross-validation, which leaves some uncertainty around its generalizability, the Autoencoder's performance suggests a high capability for pattern recognition in complex data structures. The computational demands of this model are not insubstantial, but given the right infrastructure, it has the potential to excel in a production environment, especially when fine-tuned to detect subtle indicators of fraudulent behavior.

In evaluating these models, the consideration of real-life application is paramount. A model's deployment cost, both computational and operational, must align with the scale of transaction processing typical of credit card companies, which handle immense volumes of data daily. Additionally, the model must demonstrate resilience and adaptability over time, maintaining high performance despite evolving fraud tactics.

Taking into account both the empirical results and the practical considerations for deployment, the LightGBM and SGDClassifier stand out as frontrunners in the quest for the optimal fraud detection model. Their balance of high precision, recall, and computational efficiency aligns with the title of our study, "Balancing Efficacy and Efficiency: Identifying the Optimal Model for Credit Card Fraud Detection." These models offer a compelling blend of detection capability and operational practicality, essential for the dynamic and demanding environment of credit card fraud prevention.

V. Conclusion

In our exploratory journey to fortify the bulwarks against credit card fraud, we meticulously navigated through the intricate nuances of predictive modeling, grounded in the principle of striking an optimal balance between efficacy and operational efficiency. Our article not only casts light on the financial and technical intricacies of fraud detection but also crystallizes the essence of practicality in the deployment of such systems.

The culmination of our investigation revealed that while no model emerged as the undisputed champion across all fronts, the LightGBM and SGDClassifier models demonstrated a compelling synergy of precision, recall, and computational resourcefulness, making them particularly well-suited for integration into the high-velocity realm of credit card transactions. The Autoencoder model, with its deft pattern recognition, albeit not cross-validated, showed promise and beckons future inquiry to validate its prowess.

Our study accentuates the quintessential role of data preprocessing and feature engineering as the bedrock upon which effective predictive models are built. The meticulous standardization of features such as 'Time' and 'Amount' and the strategic handling of imbalanced datasets via hybrid

sampling underscore the sophistication required in preparing data that reflects the complex realities of transactional fraud.

VI. Future Work

Looking forward, the terrain of credit card fraud detection is fertile ground for continued research and innovation. Future studies could venture into the realm of cross-validation for deep learning models such as Autoencoders, to rigorously vet their generalizability and fortify their credibility for practical deployment. Furthermore, investigating the scalability of the LightGBM and SGDClassifier models in a live environment, processing millions of transactions, would be instrumental in substantiating their applicability in real-world settings.

The integration of real-time learning, where models evolve with each transaction, adapting to the ever-shifting strategies of fraudsters, stands as a horizon to aim for. In addition, exploring the cost implications of model deployment, from computational expenditure to customer relationship management, will be invaluable in painting a holistic picture of model viability.

Our research is a testament to the dynamism and potential of machine learning and deep learning in the vanguard against credit card fraud. It is a call to arms for continued scholarly pursuit, where the fusion of academic rigor and industry acumen could pave the way for a new epoch in fraud detection—a domain where innovation is not just celebrated but necessitated by the exigencies of financial security.

VII. References

Alarfaj, F. K., I. Malik, H. U. Khan, N. Almusallam, M. Ramzan, and M. Ahmed. "Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms." IEEE Access 10 (2022): 39700-39715. https://doi.org/10.1109/ACCESS.2022.3166891.

Awoyemi, John, Adebayo Olusola Adetunmbi, and Samuel Adebayo Oluwadare. “Credit card fraud detection using machine learning techniques: A comparative analysis.” In 2017 International Conference on Computing Networking and Informatics (ICCNI), 1-9, 2017.

Bhattacharyya, S., S. Jha, K. Tharakunnel, and J. C. Westland. "Data mining for credit card fraud: A comparative study." Decision Support Systems 50, no. 3 (2011): 602–613. https://doi.org/10.1016/j.dss.2010.08.008.

Bolton, R. J., D. J. Hand, F. Provost, and L. Breiman. "Statistical fraud detection: A review." Statistical Science 17, no. 3 (2002): 235–255. https://doi.org/10.1214/ss/1042727940.

Correa Bahnsen, A., D. Aouada, and B. Ottersten. "Example-dependent cost-sensitive logistic regression for credit scoring." In Proceedings of the 2014 13th International Conference on Machine Learning and Applications, 263–269. Detroit, USA: IEEE, 2014. https://doi.org/10.1109/ICMLA.2014.48.

Correa Bahnsen, Alejandro. "Feature Engineering Strategies for Credit Card Fraud Detection." Accessed December 4, 2023. https://albahnsen.github.io/files/Feature%20Engineering%20Strategies%20for%20Credit%20Card%20Fraud%20Detection_published.pdf.

Dal Pozzolo, A., O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi. "Learned lessons in credit card fraud detection from a practitioner perspective." Expert Systems with Applications 41, no. 10 (2014): 4915–4928. https://doi.org/10.1016/j.eswa.2014.02.026.

El Hlouli, F. Z., J. Riffi, M. A. Mahraz, A. El Yahyaouy, and H. Tairi. "Credit card fraud detection based on multilayer perceptron and extreme learning machine architectures." In Proceedings of the International Conference on Intelligent Systems and Computer Vision (ISCV), June 2020, 1–5.

Ileberi, E., Y. Sun, and Z. Wang. "A machine learning based credit card fraud detection using the GA algorithm for feature selection." Journal of Big Data 9, no. 24 (2022). https://doi.org/10.1186/s40537-022-00573-8.

Khatri, S., A. Arora, and A. P. Agrawal. "Supervised machine learning algorithms for credit card fraud detection: A comparison." In Proceedings of the 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), January 2020, 680–683.

Lin, Tzu-Hsuan, and Jehn-Ruey Jiang. 2021. "Credit Card Fraud Detection with Autoencoder and Probabilistic Random Forest." Mathematics 9, no. 21: 2683. https://doi.org/10.3390/math9212683.

Randhawa, K., C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi. "Credit Card Fraud Detection Using AdaBoost and Majority Voting." IEEE Access 6 (2018): 14277-14284. https://doi.org/10.1109/ACCESS.2018.2806420.

Reshadi, Mohammadhossein. "Anomaly Detection Using Deep Learning." Master's thesis, University of Alberta, Spring 2021. https://doi.org/10.7939/r3-4trf-cy91.

The Nilson Report. "Payment Card Fraud Losses Reach $32.34 Billion." GlobeNewswire, December 22, 2022. https://www.globenewswire.com/news-release/2022/12/22/2578877/0/en/Payment-Card-Fraud-Losses-Reach-32-34-Billion.html

Van Vlasselaer, V., C. Bravo, O. Caelen, T. Eliassi-Rad, L. Akoglu, M. Snoeck, and B. Baesens. "APATE: A Novel Approach for Automated Credit Card Transaction Fraud Detection

using Network-Based Extensions." Decision Support Systems 75 (2015): 38–48. https://doi.org/10.1016/j.dss.2015.04.013

Yang, Wensi, Yuhang Zhang, Kejiang Ye, Li Li, and Chengzhong Xu. “FFD: A Federated Learning Based Method for Credit Card Fraud Detection.” In Proceedings of the 2019 BigData Congress, [Services Society], 2019.