International Journal of Advances in Data and Information Systems
Vol. 5, No. 2, October 2024, pp. 173~182
ISSN: 2721-3056, DOI: 10.59395/ijadis.v5i2.13

Improvement The Accuracy of Convolutional Neural Network with Using Undersampling Method on Unbalanced Credit Card Dataset

Kyi Pyar[1]
[1]Faculty of Computer Science, University of Computer Studies, Thaton, Myanmar
kyipyar@ucstt.edu.mm

Article Info

Article history:

Received May 19, 2024
Revised Sep 10, 2024
Accepted Oct 30, 2024

Keywords:

Credit Card Fraud Detection
Convolutional Neural Network
Undersampling
Minority
Class Imbalance

ABSTRACT

In this study, we address the challenge of imbalanced data in credit card fraud detection by proposing a novel approach that leverages Convolutional Neural Networks (CNNs) and undersampling techniques. The imbalance in the dataset, typical of real-world financial transactions, often leads to biased models favoring the majority class. To mitigate this, we employ undersampling to balance the classes, thereby enhancing the CNN's ability to learn from minority instances crucial for fraud detection. Our method is validated on a large unbalanced credit card dataset, demonstrating significant improvements in accuracy compared to traditional CNN models trained on imbalanced data. We evaluate our approach using standard performance metrics, including precision, recall, and F1-score, showcasing its effectiveness in accurately identifying fraudulent transactions while minimizing false positives. Furthermore, we pro-vide insights into the CNN's decision-making process through visualization techniques, shedding light on its ability to discern fraudulent patterns within the data. Our findings highlight the importance of addressing class imbalance in fraud detection tasks and underscore the efficacy of undersampling in enhancing the performance of deep learning models, particularly CNNs, in handling imbalanced datasets.

Corresponding Author:

Kyi Pyar[1]
[1]Faculty of Computer Science, University of Computer Studies, Thaton, Myanmar
kyipyar@ucstt.edu.mm

1. INTRODUCTION

Credit card fraud remains a significant concern for financial institutions and consumers worldwide, with billions of dollars lost annually to fraudulent transactions. Traditional methods of fraud detection often rely on rule-based systems or machine learning algorithms [1]. However, the imbalance between genuine and fraudulent transactions poses a considerable challenge, as the minority class (fraudulent transactions) is often underrepresented in the dataset, leading to biased models that struggle to accurately identify fraudulent activity.

In recent years, deep learning techniques, particularly Convolutional Neural Networks (CNNs), have shown promising results in various domains, including image recognition, natural language processing, and anomaly detection [2]. CNNs are well-suited for capturing complex patterns in data, making them a compelling choice for fraud detection tasks. However, their performance can be hindered by imbalanced datasets, where the model tends to prioritize the majority class, resulting in suboptimal detection rates for minority class instances [3].

In this context, our research focuses on addressing the challenge of imbalanced data in credit card fraud detection by proposing a novel approach that combines CNNs with undersampling techniques. Undersampling involves reducing the number of instances in the majority class to achieve a more balanced distribution between the classes, thereby allowing the model to learn from minority class instances more effectively.

The primary objective of this study is to investigate the effectiveness of undersampling in improving the accuracy of CNNs for fraud detection on unbalanced credit card datasets. By balancing the class distribution, we aim to enhance the CNN's ability to identify fraudulent transactions while minimizing false positives. Furthermore, we seek to provide insights into the decision-making process of the CNN, elucidating the features and patterns it relies on to distinguish between genuine and fraudulent transactions.

Through empirical evaluation on a large unbalanced credit card dataset, we demonstrate the efficacy of our proposed approach in enhancing fraud detection performance compared to traditional CNN models trained on imbalanced data. Our findings not only contribute to the advancement of fraud detection techniques but also provide practical insights for financial institutions seeking to strengthen their security measures against fraudulent activities.

2. RELATED WORKS

Using credit card data from an actual Chinese bank, this study investigated the creation of a churn prediction model by utilizing two data mining methods [4]. The contribution of four variable types is examined: transaction activity information, card information, risk information, and customer information. In place of survey data, the study presents a mechanism for processing variables provided from a database. Rather of adding all 135 variables to the model at once, a subset is chosen by taking into account factors related to both economic relevance and correlation analysis. The study also presents a misclassification cost metric for assessing the credit card churn prediction model. This measurement incorporates two types of errors and economic factors, providing a more comprehensive assessment of model performance. The study utilizes logistic regression and decision tree algorithms, both established and effective classification techniques. While the results indicate slightly superior performance of logistic regression over decision trees, both algorithms demonstrate efficacy in predicting credit card churn.

Big Data technologies are pivotal across various sectors such as Healthcare, Finance, Manufacturing, Transport, and E-Commerce. Among these, the financial sector, particularly banking services, is significantly impacted by digitalization and the rise of e-commerce transactions. Consequently, the proliferation of credit card usage alongside the growing number of fraudsters presents a range of challenges for the banking industry. These challenges hamper the effectiveness of Fraud Control Systems, including both Fraud Detection Systems and Fraud Prevention Systems, undermining the transparency of online payments. In response, financial institutions are actively striving to secure credit card transactions and ensure the safe and efficient use of e-banking services by customers. To achieve this, they are exploring the development of more sophisticated fraud detection techniques capable of accurately identifying fraudulent transactions and reducing instances of fraud. This article aims to delineate the fundamental aspects of fraud detection, existing fraud detection systems, the specific challenges faced by the banking sector in combatting fraud, and the available solutions leveraging machine learning techniques [5].

Using resampling techniques to increase or decrease samples from the dominant class is a common way to address class imbalance in datasets, especially in fraud detection. The main goal of this work is to provide a framework pattern for dealing with unbalanced datasets in fraud detection [6]. In order to balance the evaluated dataset, undersampling strategies (namely, Random and NearMiss) and oversampling techniques (namely, Random, SMOTE, and BorderLine SMOTE) were chosen for experimentation. This was the first time that a large-scale imbalanced dataset from the Kaggle website was used to evaluate how well these techniques worked at identifying fraudulent activity in the energy and gas usage records of a Tunisian company. For evaluation, four machine learning classifiers were used: XGBoost, Random Forest, Naive Bayes, and Logistic Regression (LR). The outcomes were analyzed using common performance measures like

accuracy, precision, recall, and F1-score. The results of the experiment showed that the Random Forest (RF) model performed better than the other classifiers.

This study addresses a significant challenge encountered in medical applications, particularly in the context of autism spectrum disorder (ASD) data, where imbalances exist be-tween cases and controls within the dataset [7]. In autism diagnosis datasets, the number of instances representing individuals without ASD typically outweighs those representing indi-viduals with ASD. This imbalance poses performance issues for machine learning models, as they may prioritize the majority class while neglecting the minority class. The research aims to empirically assess the impact of class imbalance on the performance of various classifiers using real autism datasets. Different data imbalance techniques are employed in the preprocessing phase to mitigate this issue. Oversampling techniques, such as Synthetic Minority Over-sampling Technique (SMOTE), and undersampling techniques are utilized in conjunction with classifiers like Naive Bayes, RIPPER, C4.5, and Random Forest. The study evaluates the performance of these models based on metrics such as the area under the curve (AUC) and other relevant measures. The findings indicate that oversampling techniques generally out-perform undersampling techniques, particularly concerning the toddlers' autism dataset under consideration. Additionally, the study suggests exploring the integration of sampling techniques with feature selection methods to develop models that avoid overfitting the dataset.

This study addresses the challenge of class imbalance in a medical application focused on screening for autism spectrum disorder (ASD), aiming to determine the most effective data resampling method for stabilizing classification performance [8]. Experimental analyses were conducted on an imbalanced ASD dataset pertaining to adults to evaluate the performance of various oversampling and undersampling techniques. Results obtained from multiple classifiers applied to the datasets demonstrated superior performance, particularly in specificity, sensi-tivity, and precision, when oversampling techniques were employed during the preprocessing stage.

Three machine learning approaches are assessed in this study: convolutional neural networks (CNN), multilayer perceptrons (MLP), and support vector machines (SVM) [9]. Several feature representations derived from raw photos, Canny images, and a unique combination of both are experimented with. The experimental results outperform other similar studies carried out using the same datasets. The effectiveness of the suggested methods was proven by thorough testing on three datasets. The findings show that a CNN approach performed better on a challenge dataset that resembled picture spam, whereas an SVM model had the best accuracy on a publicly accessible image spam dataset. Notably, the challenge dataset's findings greatly exceeded any earlier research in the field, while the public image spam dataset's results somewhat outpaced earlier research.

This research uses advanced machine learning and deep learning approaches to tackle the problem of credit card fraud [10]. Given the global severity of credit card fraud, our focus is on developing a model for detecting fraudulent activities, particularly imposter scams, using deep neural networks. The objective is to analyze and understand typical user behavior, specifically targeting the identification of identity fraud. Individuals exhibit distinct transaction patterns, including preferred operating systems, transaction times, and spending habits within specific timeframes. These patterns can be discerned through the application of neural networks, which excel at identifying and rec-ognizing complex patterns within data. Machine learning involves training computers to recognize patterns, akin to how the human brain functions. Deep learning, a subset of machine learning, utilizes algorithms inspired by the brain's structure and function to extract increasingly abstract representations from raw data. The model is trained using real data, processed using one-hot encoding to enable the inclusion of categorical variables for machine learning algorithms.

This study proposes an effective method for automatically detecting fraudulent credit card activities within insurance companies using a deep learning algorithm known as Autoencoders [11]. The efficacy of this approach has been demonstrated through its successful identification of fraud in real-world transaction data from European cardholders in September 2013. Ad-ditionally, the paper addresses the issue of data imbalance, which commonly affects many existing algorithms. The proposed solution involves training an autoencoder to reconstruct normal data, thereby providing a means to identify anomalies. By setting a reconstruction error threshold, cases

exceeding this threshold are classified as anomalies. The algorithm's per-formance in detecting fraudulent transactions varied: achieving a detection rate of 64% at a threshold of 5, 79% at a threshold of 3, and 91% at a threshold of 0.7. Notably, it outperformed logistic regression, which achieved a detection rate of 57% on an unbalanced dataset.

3. METHODOLOGY

This section provides random sampling, undersampling and convolutional neural network.

3.1. Random Sampling

A key technique in statistics and research is random sampling, which selects a subset of people or things from a larger population while guaranteeing that every member of the population has an equal chance of being chosen. Many different sectors, such as the social sciences, marketing research, and quality control in manufacturing, use this sampling technique extensively. Clearly defining the population from which the sample will be taken is the first stage in the random sampling process. The population may consist of individuals, objects, events, or any other entities under study. Once the population is defined, researchers must determine the desired size of the sample. This involves specifying the number of individuals or items that will be included in the sample, based on factors such as the research objectives, available resources, and desired level of precision. Each member of the population is assigned a unique identifier or number to facilitate the random selection process. This numbering ensures that every individual or item in the pop-ulation has an equal chance of being selected.

3.2. Undersampling

Undersampling is a method used in statistics and machine learning to address the issue of class imbalance in datasets, where one class significantly outweighs the other(s). This technique involves reducing the number of instances in the majority class(es) to balance the distribution of classes within the dataset. Undersampling aims to ensure that the dataset used for training machine learning models contains a more equal representation of all classes, thereby mitigating the impact of class imbalance on model performance [12]. The first step in undersampling is to identify the presence of class imbalance within the dataset. This involves examining the dis-tribution of classes and determining whether one or more classes are disproportionately rep-resented compared to others. Once class imbalance is identified, the minority class(es) with fewer instances are selected for undersampling. These classes are often the target classes of interest, such as positive cases in a binary classification problem.

In undersampling, instances from the majority class(es) are randomly selected and re-moved from the dataset until a more balanced distribution of classes is achieved. The goal is to reduce the number of instances in the majority class(es) to match the size of the minority class(es). After undersampling, the remaining instances from both the minority and majority classes are combined to create a balanced dataset. This balanced dataset is then used for training machine learning models. By balancing the distribution of classes within the dataset, under-sampling helps prevent machine learning models from being biased towards the majority class. This can lead to improved model performance, particularly for classification tasks where class imbalance is prevalent. Undersampling can help reduce the risk of overfitting in machine learning models by preventing them from memorizing the majority class and instead en-couraging them to learn more generalizable patterns from the data. Since undersampling reduces the size of the dataset by removing instances from the majority class(es), it can lead to faster training times for machine learning models, especially in cases where the original dataset is large.

3.3. Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a type of deep neural network designed to learn spatial hierarchies of features from input data, particularly effective for grid-like data such as images. Key components of a CNN include convolutional layers, pooling layers, and fully connected layers.

The convolutional layer is fundamental in extracting features by applying filters (kernels) to the input data through convolution. Each filter, a small matrix of weights, is convolved across the input, producing a feature map that highlights local patterns like edges and textures. Parameter sharing and sparse connectivity in convolutional layers enhance efficiency and reduce computational complexity. The stride parameter controls the step size of the filter, and padding helps maintain spatial dimensions. An activation function, usually ReLU, is applied to introduce non-linearity, enabling the model to learn complex feature relationships.

$$f(x) = \max(0, x) \quad (1)$$

The pooling layer in a Convolutional Neural Network (CNN) reduces the spatial dimensions of feature maps from convolutional layers, thereby decreasing computational complexity while retaining essential features. Operating independently on each feature map, pooling layers enhance the network's ability to learn invariant representations. They introduce spatial and translation invariance, with max pooling and average pooling being the primary operations. Max pooling retains the maximum value within each pooling window, highlighting salient features and promoting translation invariance. Average pooling computes the average value within each window, providing smoother summarization but potentially reducing discriminative power. By adjusting the pooling window size and stride, the pooling layer controls the degree of downsampling and information preservation in the network.

The fully connected layer in a Convolutional Neural Network (CNN) is the final stage of feature extraction and decision-making. It integrates the hierarchical representations learned from convolutional and pooling layers into a comprehensive understanding of the input data. Unlike convolutional layers, fully connected layers establish dense connections between every neuron in the current and preceding layers, each connection associated with a learnable weight. During training, these weights are adjusted through backpropagation to minimize prediction errors. In classification tasks, the fully connected layer is typically followed by an output layer with an appropriate activation function. For binary classification, a sigmoid activation function is used to produce class probabilities, while softmax activation is used for multi-class classification to generate probabilities that sum to one. In this situation, the output of the network is mapped to a probability distribution using the sigmoid activation function, which makes it easier to categorize the input data into different classes. The characteristics of this activation function are

$$\boldsymbol{\sigma(\vec{z})i} = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad (2)$$

Where, $\vec{z}$ is the input vector, z_i is the input vector's elements, $e^{z_i}$is the standard exponential function, and K is the number of classes in the multi-class classifier.

The fully connected layer in a CNN is responsible for synthesizing the hierarchical representations learned from earlier layers and making final predictions or classifications based on the integrated features. Its dense connectivity, parameterization, and application of activa-tion functions contribute to the network's ability to model complex relationships and achieve high performance in various tasks, including image recognition, object detection, and classification. Stochastic gradient descent (SGD) and its variations, Adam and RMSprop, are examples of gradient-based optimization techniques used in CNN training. The network learns to minimize a predetermined loss function, which calculates the difference between expected and actual outputs, during training. By iteratively updating the network's parameters and efficiently computing gradients, backpropagation optimizes the network's performance over time.

4. SYSTEM ARCHITECTURE

This system's goal is to build a convolutional neural network (CNN) model that has been trained on both balanced and unbalanced datasets in order to detect credit card fraud. The main objective is to assess how well the CNN model performs when trained on datasets with different class distributions. This technology specifically uses the dataset that is used to detect credit card fraud. Essentially, by using CNNs, the method seeks to overcome the problem of class imbalance that is common in credit card fraud detection datasets. We want to evaluate the efficacy of the CNN architecture in reliably detecting fraudulent transactions across various data distributions by training the model on both unbalanced and balanced datasets. The dataset is made up of credit card

transactions made in September 2013, namely by cardholders in Europe [13]. It includes transactions over a two-day period in which 492 fraudulent transactions out of 284,807 transactions were found to be fraudulent. This dataset offers a glimpse into actual credit card transactions and can be used to gain important insights regarding how common fraud is in the financial system. By documenting the occurrence of fraudulent transactions alongside legitimate ones, it facil-itates the development and evaluation of fraud detection systems and algorithms aimed at safeguarding against unauthorized or deceptive activities. The dataset's composition under-scores the challenge posed by class imbalance in fraud detection tasks, where instances of fraud represent only a small fraction of the overall transaction volume. Figure 1 describes the credit card dataset.

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.06692
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.33984
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.68928
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.17557
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.14126
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473	-2.606837	-4.918215	7.305334	1.914428	...	0.213454	0.111864	1.014480	-0.50934
284803	172787.0	-0.732789	-0.055080	2.035030	-0.738589	0.868229	1.058415	0.024330	0.294869	0.584800	...	0.214205	0.924384	0.012463	-1.01622
284804	172788.0	1.919565	-0.301254	-3.249640	-0.557828	2.630515	3.031260	-0.296827	0.708417	0.432454	...	0.232045	0.578229	-0.037501	0.64013
284805	172788.0	-0.240440	0.530483	0.702510	0.689799	-0.377961	0.623708	-0.686180	0.679145	0.392087	...	0.265245	0.800049	-0.163298	0.12320
284806	172792.0	-0.533413	-0.189733	0.703337	-0.506271	-0.012546	-0.649617	1.577006	-0.414650	0.486180	...	0.261057	0.643078	0.376777	0.00879

284807 rows × 31 columns

Figure 1. Credit Card Dataset

With fraudulent transactions making up just 0.172% of all transactions, the dataset clearly shows a class imbalance. It only includes numerical input variables that are transformed using Principal Component Analysis (PCA). We are unable to provide specific details about the original features and background material due to confidentiality restrictions. The dataset is made up of principle components that were obtained via PCA and are referred to as features V1 through V28. The 'Time' and 'Amount' features are the only ones that don't apply. The 'Time' feature shows the number of seconds that have passed between each transaction and the dataset's first transaction. Conversely, the transaction amount is indicated via the 'Amount' feature. The response variable is the 'Class' feature, which has a value of 0 in the absence of fraud and 1 in the case of fraud. These features provide essential insights into the temporal and monetary aspects of transac-tions, facilitating the development of fraud detection algorithms. While the PCA transfor-mation obscures the original features, the retained features offer valuable information for training machine learning models to accurately classify fraudulent activities in credit card transactions.

Data normalization is a crucial preprocessing step aimed at standardizing the scale of features within a dataset. It involves scaling each sample to have a unit norm while ensuring that all features are represented in the same scale. Reshaping the dataset's distribution to match a normal distribution—also referred to as a probability bell curve—is the main goal of normalization. Every feature is independently rescaled to fit within a defined range, usually between 0 and 1, during the normalization process. Min-max normalization is the technique used to do this. Each feature's minimum value is converted to 0 in min-max normalization, while the maximum value is converted to 1. Each feature's whole value range is therefore proportionally scaled to fit inside the 0 to 1 range. Min-max normalization is used in this system's environment to preprocess the dataset. The approach makes sure that every feature, regardless of its initial scale, contributes equally to the learning process by standardizing the features' scale through the use of min-max normalization. Through faster convergence and a reduction in feature scale-related problems, this normalization strategy helps machine learning models be trained more effectively [14]. Assume that attribute S's lowest and maximum values are represented by mins and maxs. In the range, the value v of S is mapped to v'.

$$v' = \frac{v - min_s}{max_s - min_s}(new\_max_s - new\_min_s) + new\_min_s \quad (3)$$

After normalization, class balancing is performed. Balancing or enhancing the class distribution within a training dataset is a critical task in machine learning, and data sampling provides various strategies to accomplish this. One such technique is undersampling, which involves reducing the number of instances of the majority class to balance the class distribution. Undersampling works by selectively removing instances from the dataset that belong to the majority class. By reducing the number of instances of the majority class, undersampling aims to alleviate the class imbalance issue and improve the performance of machine learning models, particularly in scenarios where the majority class overwhelms the minority class. After applying with the undersampling algorithm, CNN is utilized to train and test the model. In this system, a convolutional neural network (CNN) architecture is employed, consisting of 32 kernels with a dimension of 1*3 and a total of 5 convolutional layers. Each convolutional layer incorporates maximum pooling and Rectified Linear Unit (ReLU) activation functions to facilitate feature extraction and non-linearity. Notably, the pooling size is set to 4 for the first pooling layer and 2 for subsequent pooling layers, ensuring effective downsampling while retaining important features. Following the convolutional layers, a fully-connected layer is employed, utilizing the SoftMax activation function. This layer aggregates the features extracted by the convolutional layers and generates predictions based on the learned representations. The SoftMax function ensures that the output probabilities sum up to one, enabling the model to make probabilistic predictions across multiple classes. Accuracy, precision, recall, and F-measure are among the primary metrics used to evaluate the proposed system's performance. Precision estimates the percentage of accurately predicted positive cases among all positively predicted instances, whereas accuracy assesses the total correctness of the model's predictions. Conversely, recall quantifies the percentage of accurately anticipated positive examples among all of the real positive examples. Finally, the F-measure offers a balanced evaluation metric that takes into account both false positives and false negatives by providing a harmonic mean of precision and recall.

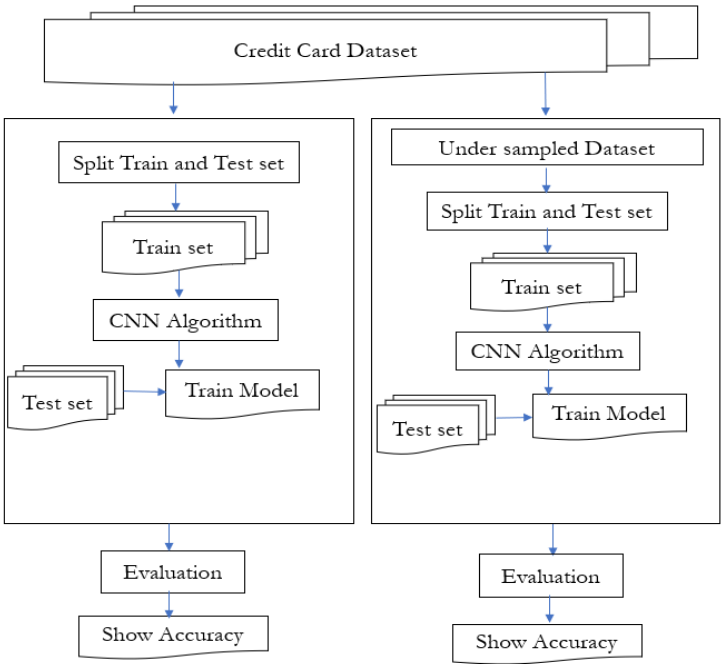

Figure 2. System Architecture

5. PERFORMANCE EVALUATION

The dataset used in this system's testing and training comes from credit card transactions. The model is created and trained with the help of Keras, an open-source machine learning

framework created by Google that is a high-level neural networks API written in Python. For creating and refining deep learning models, Keras offers an intuitive interface, and TensorFlow delivers scalable and effective computation. The initial evaluation of the system makes use of the original dataset's imbalance, which has a highly skewed class distribution between fraudulent and non-fraudulent transactions. Next, the system is tested with a balanced dataset that was obtained by applying undersampling. In order to equalize the distribution of classes and lessen the impact of class imbalance, undersampling entails lowering the number of instances from the majority class (transactions that are not fraudulent). Key performance indicators like accuracy, recall, F-measure, and precision are examined in order to judge the effectiveness of the suggested system. The comparison findings for the suggested model's performance on unbalanced and balanced by undersampling are displayed in Table 1.

Table 1. Performance Results

	Fraud			Otherwise		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Unbalanced	0.9	0.73	0.81	0.72	0.9	0.8
Undersampling	0.91	0.77	0.84	0.75	0.9	0.82

The accuracy results for the suggested model's performance on unbalanced, balanced by undersampling datasets are displayed in Figure 3.

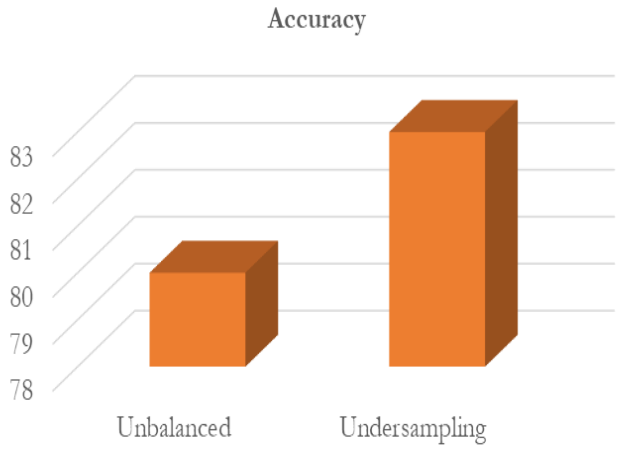

Figure 3. System Accuracy

Based on the evaluation results, it was observed that the application of random sampling techniques, specifically undersampling, led to a notable improvement in the accuracy of the Convolutional Neural Network (CNN). Undersampling involves reducing the number of in-stances in the majority class to balance the class distribution, thereby addressing issues related to class imbalance. Through the process of undersampling, the CNN model was able to achieve enhanced accuracy in classifying instances within the dataset. By selectively reducing the in-stances of the majority class, undersampling ensured that the model was trained on a more balanced representation of the data, thereby improving its ability to generalize and make ac-curate predictions on unseen data. The improvement in accuracy signifies the effectiveness of undersampling as a technique for mitigating the impact of class imbalance on model per-formance. By achieving a more balanced class distribution, undersampling enables the CNN model to allocate sufficient attention to minority class instances, thus improving its overall predictive capabilities.

6. CONCLUSION

The proposed system aims to enhance the effectiveness of credit card fraud detection, addressing limitations observed in many existing systems that primarily focus on detection without adequate support for prevention against emerging attacks using the latest data. To overcome these challenges, a Convolutional Neural Network (CNN) deep learning model has been employed to

develop the credit card fraud detection system. Through the implementation of a deep learning model, specifically a convolutional neural network, the system undergoes training using credit card transaction data. Unlike traditional approaches that may struggle to adapt to evolving fraud patterns, the CNN model leverages its ability to automatically learn and extract intricate patterns and features from the data, thereby enhancing its capacity for detecting fraudulent activities. The system's performance is evaluated using both unbalanced and bal-anced datasets. In the unbalanced dataset scenario, where instances of fraud are significantly outnumbered by legitimate transactions, the system achieves an accuracy of 80%. Upon em-ploying balancing techniques, such as undersampling, to address class imbalance, the system demonstrates improved performance, achieving an accuracy of 83%. Undersampling the majority class, such as non-fraudulent transactions, can significantly reduce the amount of valuable information available for model training, which may hinder the understanding of important patterns within the data. By decreasing the number of instances in the majority class, the model risks losing critical insights that contribute to distinguishing between fraudulent and non-fraudulent transactions. Furthermore, with fewer majority class samples, there is an increased likelihood of overfitting to the minority class, particularly in complex models like convolutional neural networks (CNNs). This overfitting can result in a model that performs well on the training data but struggles to generalize effectively to new, unseen data, leading to suboptimal performance in real-world applications. Therefore, while undersampling can help address class imbalance, it is crucial to approach this technique with caution to preserve essential information and maintain the model's robustness and generalizability. As a future work, we will explore other techniques such as oversampling, Synthetic Minority Over-sampling Technique (SMOTE), or ensemble methods that could potentially offer better performance without sacrificing data from the majority class. Moreover, there will be a comparison with other methods that use the same dataset, or random undersampling can be compared with other methods such as Tomek and others.

REFERENCES

[1] A. M. Babu, and A. Pratap, “Credit Card Fraud Detection Using Deep Learning”, 2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS), December 03-05,2020, Trivandrum.

[2] Z. Zhang, Z. Zhang, X. Zhang, L. Wang, and P. Wang, “A Model Based on Convolutional Neural Network for Online Transaction Fraud Detection” , Hindawi Security and Communication Networks Volume 2018, Article ID 5680264.

[3] K. Fu, D. Cheng, Y. Tu, and L. Zhang, “Credit Card Fraud Detection Using Convolutional Neural Networks”, ICONIP 2016, Part III, LNCS 9949, pp. 483–490, 2016. DOI: 10.1007/978-3-319-46675-0 53.

[4] G. Nie, W. Rowe, L.Zhang, Y. Tian, and Y. Shi, “Credit Card Churn Forecasting by Logistic Regression and Decision Tree”, Elsevier, Volume 38, Issue 12, November–December 2011, pp. 15273-15285, 2011.

[5] N. Bouther, A. Elomri, N. Abghour, K. Moussaid, and M. Rida, “A Review of Credit Card Fraud Detection Using Machine Learning Techniques”, 2020 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications (CloudTech), IEEE, Marrakesh, Morocco, 02 March 2021.

[6] Z. S. Rubaidi, B. B. Ammar, and M. B. Aouicha, “Fraud Detection Using Large-scale Imbalance Dataset”, International Journal on Artificial Intelligence ToolsVol. 31, No. 8 (2022) 2250037.

[7] N. Abdelhamid, A. Padmavathy, D. Peebles, F. Thabtah, and D. Goulder-Horobin, “Data Imbalance in Autism Pre-Diagnosis Classification Systems: An Experimental Study”, Journal of Information & Knowledge Management, Vol. 19, No. 01, 2040014 (2020), 2020.

[8] F. Alahmari, " A Comparison of Resampling Techniques for Medical Data Using Machine Learning", Journal of Information & Knowledge ManagementVol. 19, No. 01, 2040016 (2020), 2020.

[9] T. Sharmin, F. D. Troia, K. Potika, and M. Stamp, “Convolutional Neural Networks for Image Spam Detection”, arXiv:2204.01710v1 [cs.CV] 2 Apr 2022, 2022.

[10] O. Voican, "Credit Card Fraud Detection using Deep Learning Techniques", Informatica Economică vol. 25, no. 1/2021.

[11] M. A. Al-Shabi, " Credit Card Fraud Detection Using Autoencoder Model in Unbalanced Datasets", Journal of Advances in Mathematics and Computer Science, 33(5): 1-16, 2019; Article no.JAMCS.51106, ISSN: 2456-9968.

[12] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning”, vol. 521, Nature, London, 27 May, 2015, pp. 436-444.
[13] J. Han, M. Kamber, and J. Pei, " Data Mining: Concepts and Techniques", Third Edition, 2011.
[14] https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud