Volume (1) No (9), 2025
Journal of Global Research in Multidisciplinary Studies (JGRMS)
Review Paper/Research Paper
Available online at https://saanvipublications.com/journals/index.php/jgrms/index

Design and Analysis of Advanced Machine Learning Methods for Financial Fraud Identification in Credit Card Activities

Sandeep Gupta
SATI, Vidisha
Sandeepguptabashu@gmail.com

***Abstract*—Credit card theft is simple and easy to do. There are more ways to pay for things online, thanks to e-commerce and many other websites. This makes online scams more likely. The frequency of online transaction fraud prompted specialists to employ a toolbox of machine learning methods in their quest to uncover and analyze the problem. The risk of credit card scams has gone up a lot because of the fast growth of digital financial services like online shopping, online banking, and mobile payments. When it comes to complex fraud, traditional security measures like encryption and tokenization often fall short. Credit card theft can be better detected by comparing CNN and K-Nearest Neighbors (KNN), two machine learning models that were recommended in the study. The research employs the SMOTE-ENN resampling method to rectify the data discrepancy in a freely accessible dataset of extremely unequal cardholder transactions throughout Europe. Ensuring data quality is achieved through comprehensive preparation that incorporates Min Max scaling, categorical encoding, and imputation for missing values. Measures such as accuracy, precision, recall, F1-score, and AUPRC mostly concentrate on the model's ability to deal with class imbalance. On the basis of the lab data, the CNN model outperforms the KNN and baseline models with respect to accuracy rate (99.68%) and F1-score (99.60%). My last remarks: Through the application of deep learning techniques, develop scalable fraud detection systems capable of identifying tiny indicators of fraud in real-time financial transactions.**

***Keywords*—Credit Card Fraud Detection, Machine Learning, CNN, KNN, SMOTE-ENN, Financial.**

I. INTRODUCTION

The internet has expanded at an unprecedented rate in the past decade. Due to this, an increasing number of individuals are employing services such as online bill payment systems, swipe and pay systems, and e-commerce. Credit card transactions have also been a focal point of fraudsters' activities as a result [1][2]. Encrypting and tokenizing credit card data are two of several ways to make sure that credit card transactions are secure [3]. While these measures usually work, they still can't guarantee that credit card transactions won't be fraudulent.

One sort of credit card is the standard plastic card, which allows the cardholder to make purchases up to the card's credit limit or get cash advances [4]. With a credit card, you may save time and avoid waiting in line. This means that they can pay back their debt later, by putting it off until the next billing session. It's easy for thieves to steal credit cards [5]. Quickly and safely withdraw a substantial sum without the owner's knowledge [6][7][8]. The fact that con artists are persistent in their attempts to pass off fraudulent transactions as legitimate makes detecting fraud a formidable challenge. According to data given by the FTC in 2017, 1,579 data breaches occurred, compromising over 179 million records. Out of all the reports, 133,015 were related to credit card fraud, 82,051 to tax or employment matters, 55,045 to phone fraud, and 50,517 to bank fraud.

In the United States, customers could only meet with bank workers in person until Citibank and Wells Fargo Bank released the first internet banking app in 1996. Paying with credit cards online became more common after the advent of online banking [8][9]. In the last ten years, this has grown substantially, and with it, new and popular services like social media, online banking, e-commerce, and online payment systems [10]. Because of this, con artists have stepped up their game to steal money from people making purchases online using different payment methods. Modern advancements in digital technology, especially those pertaining to monetary transactions, have altered the way individuals function in relation to their money. Digital platforms have greatly replaced physical pay stations in several payment systems [11][12]. The use of technology in digital transactions has greatly impacted the field of economics, particularly for those seeking to maintain productivity and competitive advantage [13][14]. As a result, consumers have found that using their credit cards for online banking and other transactions is a simple way to handle their finances and other banking needs without leaving the comfort of their homes or offices.

Machine learning-based fraud detection systems have been more useful in spotting suspicious financial transactions and thwarting cybercrime in recent years [15] [16]. Systematic literature reviews (SLRs) reveal that, although using different experimental methodologies, all of these algorithms apply performance measures to evaluate how well they forecast whether a financial transaction is fraudulent. Accuracy, Sensitivity, F1 Score, Recall, and Precision are only a few examples of such measurements.

A. Motivation and Contribution of the Study

Online banking, e-commerce, and tap-and-pay systems have made things a lot easier for customers, but they have also made credit card scams much more likely, thanks to the internet's rapid growth. Despite traditional security measures like encryption and tokenization, fraudsters continue to exploit digital transactions, often making fraudulent activities appear legitimate, which complicates detection efforts.

Intelligent, automated solutions are urgently needed due to the high number of reported fraud instances each year, particularly in the area of credit card usage. ML techniques are becoming increasingly popular as a result of this changing threat scenario. Based on evaluation measures like Recall, Accuracy, Precision and F1 Score, these methods have improved the effectiveness of fraud detection and are better at identifying unusual patterns. A possible solution to the problem of financial insecurity in today's technology-driven world could be to employ machine learning to make fraud detection more effective.

- Improving model learning by utilizing SMOTE-ENN to efficiently balance fraudulent and non-fraudulent transactions.
- Implemented dual-model comparison by evaluating both KNN and CNN to assess performance across traditional and DL approaches.
- Including imputation, categorical encoding, and Min Max scaling, to ensure data quality and model compatibility.
- Used AUPRC and F1-score over accuracy to evaluate models more effectively in the context of highly imbalanced data.
- Identified key influential features via importance analysis, supporting interpretability and potential feature selection for future models.

B. Justification and Novelty

The study compares the efficiency of two DL approaches—CNNs and classical NN—in order to tackle the pressing issue of credit card fraud detection in highly unstable datasets. The novelty lies in the dual-model approach combined with the application of SMOTE-ENN, a hybrid resampling technique that not only balances the dataset but also filters out noisy data, enhancing model learning and generalization. Performance metrics such as AUPRC and F1-score are more applicable to imbalanced data circumstances, and this study differs from others in that it places an emphasis on effective feature scaling, strong preprocessing, and the usage of these techniques rather than typical machine learning or ignoring class imbalance. This study's findings highlight CNN's reliability and accuracy in fraud detection, which is a big step towards developing real-time, scalable systems that prevent fraud with few false positives and maximum precision.

C. Structure of paper

The following is the outline for the paper: In Section II, review the literature on approaches for detecting credit card fraud. Section III lays out the strategy (which makes use of KNN and CNN); The findings and analysis of the experiments are presented in Section IV, while Section V outlines the prospective area of future work.

II. Literature Review

A number of credit card fraud detection techniques based on machine learning have been identified in this literature research. These algorithms mostly include hybrid models, SMOTE, ensemble methods, and assessment measures. Research in this area is summarized in Table I. Credit card fraud is detected by the use of many ML algorithms in testing.

Dharma et al. (2025) proposed a study on credit cards have grown significantly because of quick development of e-commerce and online banking. This leads to many fraud transactions. Normally, credit card frauds occur mainly when the card is lost at an unauthorized purpose or any fraudsters attempt to utilize card improperly by the person using card for his/her online payment transparency. Once the necessary data is gathered and made accessible, ML techniques can detect instances of credit card fraud. In order to identify fraudulent credit card transactions, this research suggests a mixed ML strategy. Totaling 284,807 card purchases conducted across Europe in September 2013, this study's dataset was employed. Hybrid models can be created by combining ML approaches like SVM and LR. An F1-score, accuracy, precision, and recall are the four cornerstones of any Hybrid ML system. When compared to competing models, the one given here outperforms them on all of the chosen performance indicators: High precision (96%) and accuracy (97%) as well as recall (97%) and F1-score (97%) [17].

Sharma et al. (2025) concentrated on vital areas where fraud detection is of the utmost importance, not just for credit cards but for all transactions involving money. The field of ML has developed a suite of impressive tools capable of sifting through mountains of data in search of intricate, previously unseen patterns. Multiple methods for detecting fraud using ML algorithms are detailed in this abstract. In this work, diverse algorithms are compared according to their F1-score, Accuracy, Precision, and Recall. DT, RF, LR, SVM, and KNN are some of the applied algorithms. The dataset was collected from Kaggle, and it contains 284806 rows and a total of 31 columns [18].

Singh et al. (2024) explored the way two well-known machine learning methods—XGBoost and ANNs—could find cases of credit card theft. Use a publicly available dataset of credit card transactions to compare the F1 scores, recall, accuracy, and precision of multiple techniques. The study also looks at how well ANNs and XGBoost scale computationally and whether they are suitable for systems that detect fraud in real-time. Out of the five approaches that were assessed, ANNs attain the highest accuracy at 96.9%, while XGBoost outperforms all other classifiers with 92.7%. Financial institutions thinking about implementing or improving fraud detection systems can benefit from these results since they highlight the pros and limitations of each approach [19].

Singh and Vats et al. (2023) Critical issues have recently surfaced and must be addressed immediately. For the sake of convenience, these days everyone is moving towards online and cashless transactions. The flip side of this ease, though, is a massive fraud enterprise. Many individuals fall for this trap every day. A small step towards resolving this problem has been accomplished by this research. Using Decision Trees, Logistics Regression, and Random Forest, among other Machine Learning approaches, this academic study finds fraudulent transactions in real-world data. The Synthetic Minority Oversampling Technique is another tool for resolving dataset imbalances. Then, they evaluate the performance of machine learning approaches that make use of the "With SMOTE" and "Without SMOTE" methodologies [20].

Verma and Badholia et al. (2022) discussed SMOTE, a Synthetic Minority Oversampling Technique, which makes use of several ML models. After ensuring that all classes are appropriately balanced, use the metaheuristic method of Particle Swarm Optimization (PSO) to fine-tune an ensemble model. On a worldwide scale, PSO finds the best solution here. The following performance matrices are used to evaluate

the proposed model R, F1-Measure, Accuracy (ACC), and Precision (P). Using both S MOTE and baseline ML models, the model is tested and contrasted in an ensemble setting. According to the findings, the suggested approach is reliable and effective enough to identify fraudulent transactions [21].

Shah et al. (2021) processed cases of fraudulent chargebacks using six widely used ML algorithms. For every machine learning technique, one can construct a confusion matrix to measure the algorithm's performance. Recall, specificity, accuracy, precision, F1 score, and many other metrics are utilised to assess their efficacy. The results show that when it comes to identifying fraudulent charges on credit cards, machine learning techniques triumph. Suggestion: To combat fraud, it is recommended to employ a combination of machine learning methods, even though individual algorithms have impressive recall and precision [22].

Ileberi et al. (2021) created an ML approach to detect credit card fraud with the use of an imbalanced real-world dataset including European cardholders. In order to rectify the imbalance in the classes, they resampled the dataset using the SMOTE technique. Various machine learning approaches were used to test this framework Equations including RF, DL, XGBoost, DT, SVM, and LR. The Adaptive Boosting (AdaBoost) method was used in conjunction with these ML algorithms to provide better classification results. Models were assessed using precision, accuracy, recall, MCC, and AUC. According to the experimental results, using AdaBoost improves the performance of the proposed methods. Additionally, compared to previous approaches, the boosted models produced better outcomes [23].

Credit card fraud detection using ML is on the rise, however many important questions are yet unsolved. Without thoroughly investigating the possibilities of deep learning or hybrid deep-ensemble frameworks, the majority of current research depends substantially on conventional machine learning models as SVM, DT, LR, and ensemble approaches. While techniques like SMOTE and AdaBoost have shown improvements in handling class imbalance, limited work has integrated these with advanced models like CNNs, LSTMs, or Transformer-based architectures for real-time detection. Additionally, many studies focus on accuracy-related metrics but overlook computational efficiency, latency, and scalability factors crucial for real-world implementation. Furthermore, a lack of standardized evaluation protocols and the underutilization of real-time streaming data leave a significant gap in deploying truly robust, adaptive, and scalable fraud detection systems

TABLE I. COMPARATIVE ANALYSIS OF MACHINE LEARNING TECHNIQUES FOR FINANCIAL FRAUD DETECTION IN CREDIT CARD ACTIVITIES

Author	Methodology	Data	Key Findings	Limitation	Future Work
Dharma et al. (2025)	Hybrid ML approach using SVM + Logistic Regression	284,807 transactions from European cardholders (Sept 2013)	Occurred with good precision (96%), recall (97%), and F1-score (97%)	Limited to only two algorithms; lacks ensemble or deep learning techniques	Expand model using ensemble or deep neural networks for enhanced accuracy
Sharma et al. (2025)	Comparative study using SVM, KNN, DT, RF, LR	Kaggle dataset with 284,806 rows and 31 features	Provides algorithm-wise comparison using standard metrics (Accuracy, Precision, Recall, F1)	No hybrid or ensemble approach used; lacks deep learning	Include deep learning models and explore real-time deployment strategies
Singh et al. (2024)	Comparison of ANN vs XGBoost	Public credit card dataset	ANN achieved highest Accuracy (96.9%); XGBoost at 92.7% was second-best	Does not combine ANN and XGBoost in an ensemble model	Investigate ensemble of ANN and XGBoost; assess for real-time systems
Singh and Vats et al. (2023)	ML models with SMOTE (DT, LR, RF)	Real-world dataset with class imbalance	SMOTE improved performance of models over imbalanced data	Evaluation lacks advanced algorithms like deep learning or ensembles	Apply CNN, LSTM or ensemble techniques with advanced sampling methods
Verma and Badholia et al. (2022)	SMOTE + Ensemble with PSO optimization	Credit card dataset with class imbalance	The ensemble model tuned via PSO outperformed baseline models	Computational cost of PSO not discussed; lacks comparison with deep learning	Test other metaheuristics (GA, ACO); compare with CNN-based systems
Shah et al. (2021)	Six ML algorithms (unspecified) with Confusion Matrix evaluation	General credit card fraud dataset	Strong results across all models in terms of precision and recall	No mention of data balancing techniques or deep learning	Propose hybrid ML + DL model with real-time fraud alerting
Ileberi et al. (2021)	AdaBoost and SMOTE with ML models (SVM, LR, RF, XGBoost, DT, ET)	Real European cardholders' dataset + synthetic set	Boosted models via AdaBoost performed best, overcoming class imbalance	AdaBoost alone may not suffice in extreme imbalance; lacks DL comparison	Combine boosting with deep models; apply model to large-scale real-time systems

III. METHODOLOGY

The organized pipeline of the credit card fraud detection system is built using data acquired from a Kaggle dataset that contains 284,807 transactions mentioned in Figure 1. Out of these, only 492 are fraudulent, indicating a significant class imbalance. In contrast to class distributions, 1hich highlight imbalance and its correction by SMOTE-ENN resampling, feature correlations reveal few linear relationships between variables. Missing value management, categorical data encoding, and Min Max Scaler scaling of numerical characteristics are all part of the pre-processing procedures. After processing, the data is stratified so that 80% becomes the training set and 20% becomes the testing set. In this case, CNNs and KNN are the models that are utilised. Using the most common class in the set of k-neighbours, the KNN algorithm sorts a test set. The CNN model uses pooling, convolutional, and fully connected layers to retrieve hierarchical information, which enables it to generate class label predictions that are very accurate. Because of the class imbalance, AUPRC and F1-score are used to measure the model's performance instead of AUROC, recall, accuracy, and precision. This helps guarantee that the fraud detection system is robust and easy to understand.

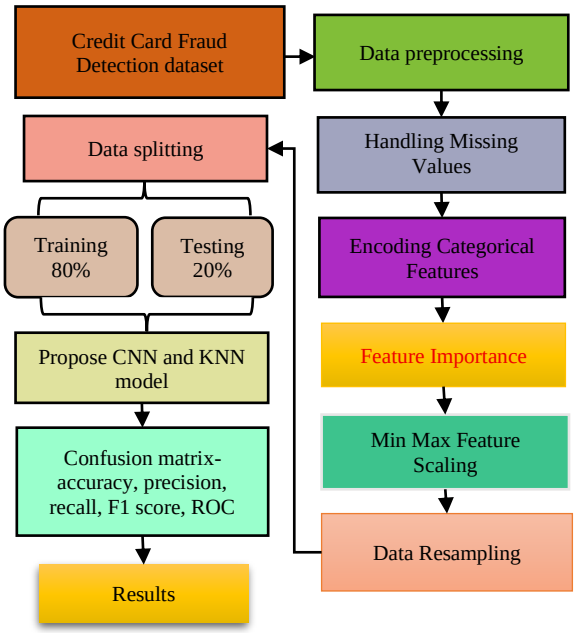

Fig. 1. Flowchart of the Financial Fraud Detection in Credit Card Activities

The following sections provide each step that also shown in methodology and proposed flowchart:

A. *Data Collection*

The Credit Card Fraud Detection dataset, which includes European cardholders' transaction data and was acquired through Kaggle in September 2024, is crucial to this study. In order to construct efficient CCFD models, researchers often use this dataset of European credit card transactions. Among the 2,84,807 total transactions, 492 were fraudulent, making up just 0.172 percent. This represents an incredibly imbalanced number of transactions. The number of features in each transaction increases to 31 when the additional variables 'Class,' 'Time,' and 'Amount' are added to the 28 anonymised major components acquired by principal component analysis (PCA).

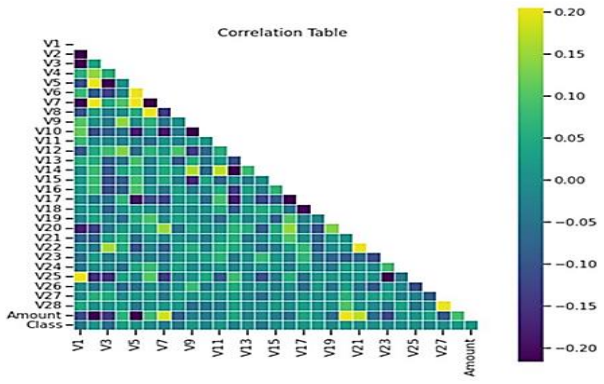

Fig. 2. Features Correlation

Figure 2 presents the pairwise correlation between various features (V1 through V28, Amount, and Class). The color intensity and the accompanying legend indicate the strength and direction of the correlation, ranging from approximately -0.20 (dark purple) to 0.20 (bright yellow). A darker color suggests a strong negative correlation, while a brighter color indicates a strong positive correlation. White lines separate each cell, making individual correlations visible. Most of the correlations appear to be relatively weak, falling within the range of -0.20 to 0.20, suggesting that many features are largely independent of each other. This is particularly noticeable in the "Class" row, where most features show very low correlation with the target variable "Class", implying that no single feature strongly predicts the "Class" on its own.

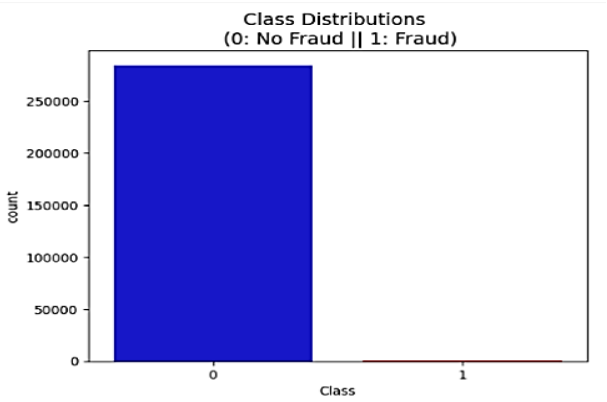

Fig. 3. Class Distribution of Imbalanced

The dataset in Figure 3 is drastically skewed towards the "0: No Fraud" category, with a numerical advantage over the "1: Fraud" category. The "No Fraud" class has a count exceeding 250,000, represented by the large blue bar, while the "Fraud" class has a very small count, barely visible above the x-axis, indicating only a few hundred instances. The issue with ML models is that they can get biased towards the majority class and overlook the fraudulent class because of this severe class imbalance.

B. *Data Preprocessing*

Processing the raw data included dealing with missing values by removing features that were missing more than 60% of the data and filling in the rest using median (numerical) and mode (categorical). Utilising binary and one-hot encoding techniques, categorical features were codified. Every feature was scaled to a range of 0 to 1 by Min Max Scaler. A dataset with 37,894 fraud and non-fraud samples was created by applying SMOTE-ENN to artificially manufacture fraud cases and exclude noisy samples. Resolving the problem of wealth disparity was the primary objective. These steps improved the model's accuracy and the data's overall quality.

C. *Handling Missing Values*

Missing values are prevalent in real-world datasets and, if not handled appropriately, can greatly affect the performance of models. In this study, missing values were analysed, and features with more than 60% missing data were dropped, as they were deemed to carry insufficient information to contribute meaningfully to the predictive model. The remaining features were enhanced by imputed numeric attributes using the median and categorical variables using the mode. Because of the imbalance in the dataset, which could cause biased imputations and a skewed mean, the median was chosen instead of the mean.

D. *Encoding Categorical Features*

The majority of ML algorithms only work with numerical features as input. After removing features with high missing values, 15 categorical attributes remained. Ten of these had binary categories (e.g., True/False), which were encoded as 0 and 1. The one-hot encoding approach was used to transform the remaining multi-level category characteristics. This technique transforms each category into a binary vector,

preserving categorical relationships without implying any ordinal ranking.

E. Features Importance

The main portion of the dataset consists of the values labelled V1 through V28, which are generated by Principal Component Analysis (PCA). Both dimensionality and redundancy can be diminished in this way. No changes have been made to the variables "Time" (the number of seconds that have elapsed) and "Amount". The "Class" feature indicates whether the transaction is fraudulent or not. To account for the massive disparity in class sizes, they don't just look at the model's accuracy but also its AUPRC (Area Under the Precision-Recall Curve). A bar chart (Figure 4) highlights feature importance, offering insights into which variables most influence model predictions.

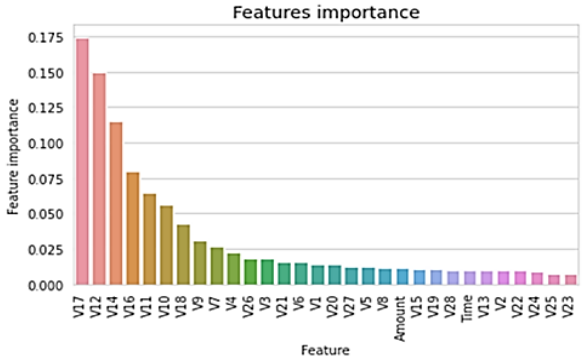

Fig. 4. Features Importance

Figure 4 presents the relative importance of various features, labelled V1 through V27, along with "Amount" and "Time," in a predictive model. The most essential feature is V17, which has a feature importance value of about 0.17. This property has the greatest influence on the model's predictions, followed by V12 (around 0.15), and V14 (about 0.115). The importance generally decreases from left to right, with features like V23 and V25 having the lowest importance scores, barely above zero. In order to make the model more accurate and easier to comprehend, this visualisation is crucial for determining which inputs significantly affect the model's output. This, in turn, can aid with feature selection and engineering.

F. Min Max Feature Scaling

Feature scaling makes sure that all of the model's variables are roughly the same size, so that features with bigger numbers can't overpower features with smaller ones. In this research, Min Max Scaler was used to normalize the features to a range between 0 and 1. This method is particularly effective as it preserves sparsity and does not distort the distribution of data with many zero entries. The formula used for scaling in Equation (1):

$$x' = \frac{x - min\,(x)}{max(x) - min(x)} \quad (1)$$

Where x' is the scaled value, and x is the original feature value.

G. Data Resampling

Problems arise when machine learning models encounter imbalanced datasets, which can cause them to favour the dominant class. This was remedied by employing the SMOTE method. In order to achieve statistical parity, SMOTE generates fictitious instances from the minority group. This resampling method enhanced the clarity between classes and improved model learning. Figure 5 shows the data distribution after applying SMOTE technique.

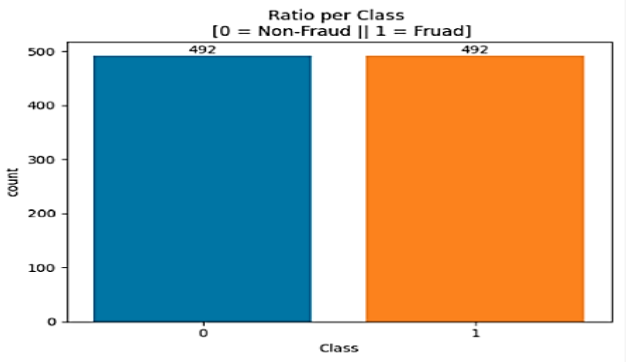

Fig. 5. Class Distribution of Balanced

The bar chart titled "Ratio per Class" displays (Figure 5), the class distribution after balancing, where both "0 = non-fraud" and "1 = Fraud" classes now have an equal count of 492 instances each. The ML model will be trained without bias towards the majority class, thanks to this balanced distribution, which is a huge improvement over the previous imbalanced dataset. More accurate and trustworthy predictions, especially for the important minority class (fraud), should result from correcting the class imbalance so that the model learns equally from fraudulent and non-fraudulent instances.

H. Data Splitting

The effectiveness of the model was evaluated by creating a training set and a testing set from the pre-processed dataset. A common practice is to split the data into a testing set and a training set using the train_test_split function in the Scikit-learn module. By checking if it retains the majority of data when tested on new samples, they can objectively evaluate the model's capacity to generalise. The use of stratified splitting allowed us to maintain the class distribution across both sets. When dealing with issues of imbalanced categorisation, like fraud detection, this is a crucial step to take.

I. Proposed KNN Model

Supervised ML methods like KNN are versatile enough to manage classification and regression tasks [24]. The fact that it doesn't take long makes many people think this is a lazy way to learn. The KNN algorithm allocates a value from the training dataset proportional to the degree of similarity between features. It then utilizes this value to estimate the unknown value of a new data point [25]. A popular technique to quantify the distance between the affected input and surrounding samples is by using the "Euclidean distance" metric. By utilizing the predicted similarity metrics, the input dataset can be reduced to a single distance measure. According to the classification or regression problem, KNN uses the mean or mode of the labels to classify or forecast this input [26]. A two-point distance in terms of the standard geometrical unit can be expressed as in Equation (2):

$$d(v,u) = \sqrt{\sum_{i=1}^{N}(v_i - u_i)^2} \quad (2)$$

where v_i represents the i^{th} feature of the input vector and u_i is the i^{th}Feature of the training range.

J. *Proposed CNN Model*

CNNs enable deep learning. The building blocks of a CNN include a pooling layer, a convolutional layer or layers that follow it, and so on. Along with a classification layer, these two layers complement one another. This research made use of Kim's (2014) proposed CNN model. This model is based on a slightly modified version of Colbert's convolutional neural network architecture. Figure 6 shows the layout of a CNN. To get important features out of the input data, this architecture employs four layers: convolutional, sub-sampling, fully connected, and classification. The supplied data is classified based on these features.

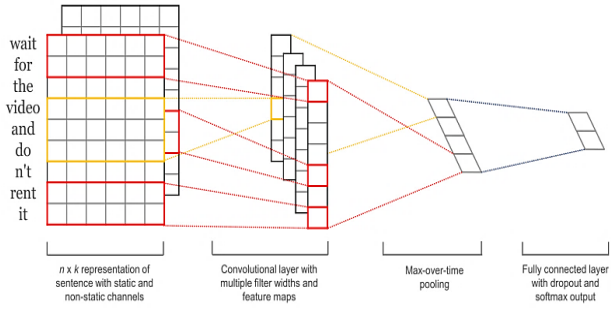

Fig. 6. Convolutional Neural Networks (CNN)

In the input layer, each of the n inputs is shown as a dense vector with k dimensions. This indicates that the input x should be utilised with a feature map that has $d\ b\ k$ dimensions. Word vector $x_i \in \mathbb{R}^k$He k dimensions represent the i-th word in the input sentence. The above Equation (3) represents a sentence of length n as:

$$x_{1:n} = x_1 \oplus x_2 \oplus \ldots \oplus x_n \tag{3}$$

The concatenation operator is represented by the symbol $\oplus$. The process of creating a new feature involves applying a filter from the set $w \in \mathbb{R}^{hk}$On to a window containing a set of h words through a convolution operation. A new variable $c_i$feature is created in this way, for instance, with a word window of $x_{i:i+h-1}$Ds.

$$c_i = f(w.x_{i:i+h-1} + b) \tag{4}$$

The hyperbolic tangent and other non-linear functions are represented by f in Equation. (4), while the bias term is denoted by $b \in \mathbb{R}$. The sentence x_1:h, x_2:h+1,..., $x_{n=h+1}$: n. is applied to each potential word window using this convolution filter to generate a feature map. This feature map is constructed using Equation (5):

$$c = [c_1, c_2, \ldots c_{n-h+1}] \tag{5}$$

In this case, the set $\mathbb{R}^{n-h+1}$includes c. To get the highest values that satisfy the filters, use the feature map and a max-over-time pooling operation. To obtain the most noticeable features for use in feature maps, this method is employed.

The aim of the model is to identify various traits by employing a variety of filters with varied window widths. Layers with these features send their outputs to a fully connected layer at the very bottom. The label probability distribution is determined using a fully-connected SoftMax layer [27].

K. *Performance Matrix*

A confusion matrix, often known as a confusion table, is a common way to show how well a prediction model does when tested with known-true data. Put simply, it is a matrix that spans both the Real Classes and the Predicted Classes.

Definitions of Terms Used in the Confusion Tree:

- **True Positive (TP):** The anticipated frequency of good occurrences.
- **False Positive (FP):** The proportion of predicted positive cases that actually have a negative outcome.
- **True Negative (TN):** This represents the anticipated negative case count.
- **False Negative (FN):** Values that are predicted to have positive results in the event of unfavourable cases.

Accuracy: An accurate model is one that, according to Equation (6), properly classifies a certain percentage of transactions as either fraudulent or genuine out of a total number of transactions:

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{6}$$

Precision: The accuracy with which a system identifies instances of fraud is referred to as its precision. To find the percentage of false predictions compared to the total number of cases that were identified properly, use Equation (7):

$$Precision = \frac{TP}{TP+FP} \tag{7}$$

Recall: Equation (8) states that the recall is the ratio of the number of fraudulent transactions that were successfully detected to the total number of fraudulent transactions (including both false negatives and true positives):

$$Recall = \frac{TP}{TP+FN} \tag{8}$$

F1 Score: The F1-score, which averages recall and precision in the case of non-uniform class distributions (Equation (9), helps to stabilize the two variables:

$$F1 - score = \frac{(precision*recall}{(precision+recall)} \tag{9}$$

The "(AUROC)" is another name for this. It is possible to see how well a model works by using it. Plotting the area under the receiver operating characteristic (AUROC) curve involves using a variety of threshold values. These values encompass both the true positive and false positive rates.

IV. Results and Discussion

This section presents the approach that was used to evaluate the prediction performance of the algorithms that were chosen to identify credit card fraud. Finding other studies that have utilised the same or similar datasets to compare the findings with is the next step. All of the simulations were executed on a robust system with 8 GB of RAM and an Intel® CoreTM i5-1035G1 CPU operating at 1.19 GHz. Use Python 3.10.1 and JupyterLab® 3.2.1. When comparing the KNN and CNN models, the four most important parameters are the F1-score, recall, accuracy, and precision (Table II). No matter what metric is used to compare the two models, the CNN model continuously comes out on top. While KNN only has a 97.83% success rate, CNN manages a whopping 99.68%. In comparison to KNN's 95.69% accuracy, CNN manages a whopping 99.45% precision. Contrasted with KNN's 97.80% recall, CNN's 99.75% is far better. Last but not least, CNN outperforms KNN by a wide margin with an F1-score of 99.60%, while KNN's F1-score remains at 96.84%. Taken together, the results demonstrate that the CNN model outperforms and outperforms the KNN model.

TABLE II. PERFORMANCE PARAMETERS KNN AND CNN MODELS

Measures	KNN	CNN
Accuracy	97.83	99.68
Precision	95.89	99.45
Recall	97.80	99.75
F1-score	96.84	99.60

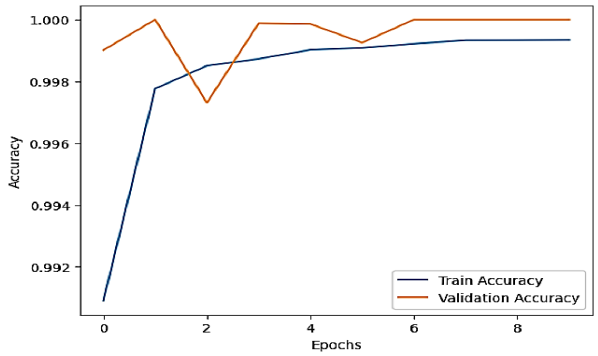

Fig. 7. Accuracy Curve of CNN Model

Figure 7 shows the accuracy of training and validation throughout 9 epochs. The train accuracy (blue line) shows a rapid increase from epoch 0 to around epoch 1, quickly reaching above 0.998, and then steadily climbing to approximately 0.9995 by epoch 7, remaining stable thereafter. The validation accuracy (orange line) exhibits more fluctuation, initially rising to 1.0 at epoch 1, dipping at epoch 2, and then stabilizing around 0.9995 to 1.0 from epoch 4 onwards. While both curves generally show good performance, the slight divergence and the validation accuracy's fluctuations suggest that the model might be experiencing minor overfitting or variability on the validation set, though overall, it appears to generalize well as both accuracies are consistently high and close to each other towards the later epochs.

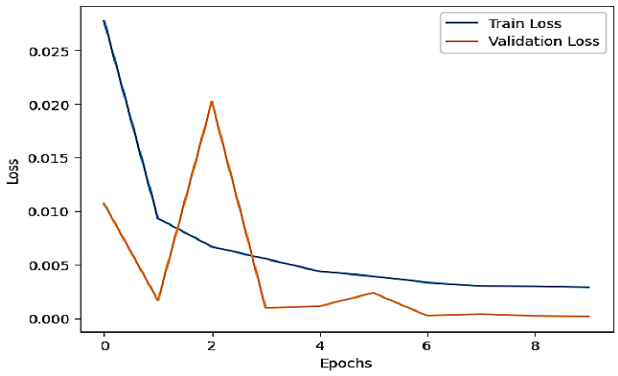

Fig. 8. Loss Curve of CNN Model

The blue line shows the training loss after 9 iterations, while the orange line shows the validation loss (Figure 8). As the model learns from its training data and becomes more accurate, a steep initial train loss followed by a continuous drop in that loss indicates success. By epoch 9, the train loss has stabilized at a very low value, close to 0.003. The validation loss, however, exhibits more volatility. It starts lower than the train loss, drops significantly at epoch 1, spikes sharply at epoch 2, and then drops again to a very low level by epoch 3, where it generally remains stable and close to zero for the subsequent epochs. Validation loss compared to train loss initially shows a large discrepancy and spike, which could indicate instability or overfitting in the early stages of training. However, by the later epochs, both curves converge to very low values, which means the model generalizes well and performs well on unseen data.

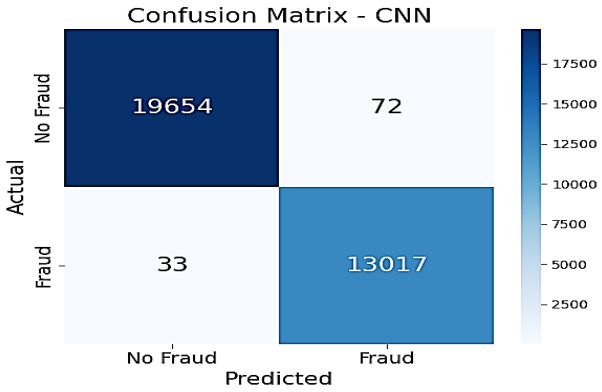

Fig. 9. Confusion Matrix of CNN Model

Experiment results for the CNN model's "No Fraud" and "Fraud" event detection accuracy are displayed in Figure 9. In the matrix, the model had a success rate of 19,654 True Negatives (cases correctly classified as "No Fraud") and 13,017 True Positives (cases correctly classed as "Fraud"). Unfortunately, 72 cases of "No Fraud" were wrongly identified as "Fraud" (False Positives), while 33 cases of "Fraud" were wrongly identified as "No Fraud" (False Negatives). The model is very good at differentiating between the two kinds of cases, but it's far more prone to false positives, in which it incorrectly identifies a valid case as a fraudulent one, than negatives, in which it correctly identifies a fraudulent case (false negatives).

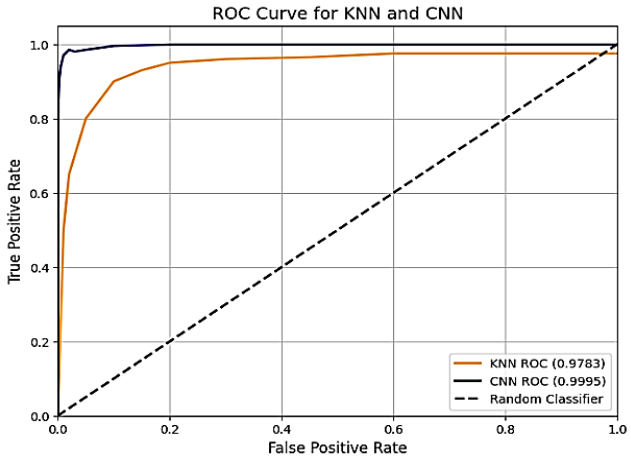

Fig. 10. ROC Curve for CNN and KNN

The ROC curves of the CNN and KNN models are shown in Figure 10, which allows for a visual comparison of their performance in class distinction. The CNN model (blue line) worked very well in differentiating between positive and negative classes, as shown by an AUC of 0.9995. Keeping firmly in the upper left area of the graph, it keeps the TPR high and the FPR low across most thresholds. Alternatively, the KNN model (orange line) is significantly less successful than the CNN, while it is still performing well with an AUC of 0.9783. When compared to the CNN, its FPR for a given TPR is larger since its curve is further from the top-left corner. The dotted black line represents the baseline, a random classifier with an AUC of 0.5; both models significantly outperform it. Accordingly, it appears that, when it comes to this specific task, the CNN model is the superior classifier.

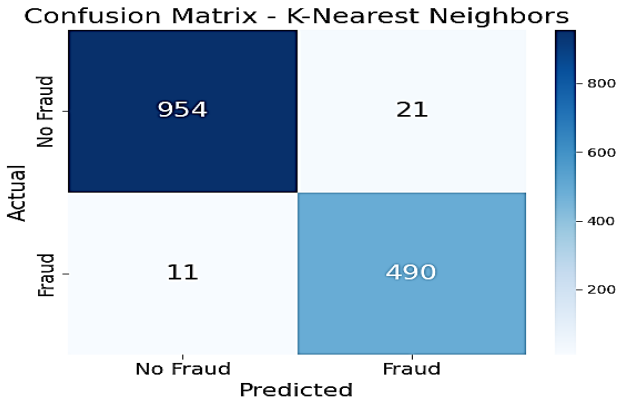

Fig. 11. Confusion Matrix of KNN Model

Figure 11 shows the number of "No Fraud" and "Fraud" occurrences correctly identified by the KNN model. From those instances, 490 were appropriately tagged as "Fraud" and 954 as "No Fraud" (True Negatives) by the model. The system caused 11 instances of "Fraud" to be wrongly labelled as "No Fraud" and 21 instances of "No Fraud" to be incorrectly labelled as "Fraud" (False Negatives). With a smaller dataset and a low number of misclassifications, the KNN model seems to have fewer overall predictions compared to the CNN model's confusion matrix (if available). However, it manages to properly identify both classes.

TABLE III. COMPARISON BETWEEN BASE AND PROPOSED MODELPERFORMANCE MATRIX FOR FINANCIAL FRAUD DETECTION IN CREDIT CARD ACTIVITIES

Matrix	**KNN**	**CNN**	**RF**[1]	**LR**[28]
Accuracy	97.83	99.68	83.78	95.72
Precision	95.89	99.45	79.64	89.11
Recall	97.80	99.75	92.78	60.00
F1-score	96.84	99.60	85.71	71.71

Table III shows a comparison of various machine learning models that try to identify fraudulent credit card purchases. According to reference [28], these models encompass KNN, CNN, RF, and LR. When it comes to precision and dependability, the CNN model is unrivalled. An outstanding performance is shown by a remarkable score of 99.68% in all four metrics: accuracy, precision, recall, and F1-score. Although it does a respectable job, the KNN model is clearly inferior to CNN. The RF and LR models, presumably from external studies, show significantly lower performance compared to both KNN and CNN, especially in Recall for LR (60.00%) and overall F1-score for RF (85.71%).

V. CONCLUSION AND FUTURE SCOPE

In the 21st century, the emergence of internet banking and electronic payment systems has significantly transformed the financial services sector by enhancing accessibility and convenience for consumers. Among these innovations, credit cards play a pivotal role, enabling cashless transactions while offering safeguards against loss, theft, or product damage. Such advancements not only improve the overall customer experience but also necessitate verification processes with merchants to maintain transaction security. The CNN model substantially outperformed the KNN model in identifying financial fraud in credit card transactions, as this study well-demonstrates using sophisticated machine learning techniques. In comparison to KNN's 97.83% accuracy and 96.84% F1-score, the CNN attained an impressive 99.68% accuracy, 99.45% precision, 99.75% recall, and 99.60% F1-score. Rigid data preprocessing, including dataset balance using SMOTE-ENN and normalization using MinMax scaling, yielded these results. Evaluation on the skewed dataset was made meaningful with the usage of AUPRC and F1-score. In the future, to further understand sequential transaction patterns, the research can be enhanced by including hybrid DL models like CNN-LSTM or architectures based on Transformers. To further enhance the security and responsiveness of financial systems, the model can be tested on live-streaming data and implemented into real-time fraud detection systems. This will allow for the validation of scalability, reduction of detection latency, and adaptation to emerging fraud strategies. “

REFERENCES

[1] E. Ileberi, Y. Sun, and Z. Wang, “A machine learning based credit card fraud detection using the GA algorithm for feature selection,” *J. Big Data*, 2022, doi: 10.1186/s40537-022-00573-8.

[2] H. Kali, “Optimizing Credit Card Fraud Transactions identification and classification in banking industry Using Machine Learning Algorithms,” *Int. J. Recent Technol. Sci. Manag.*, vol. 9, no. 11, pp. 85–96, 2024.

[3] S. J. Wawge, “A Survey on the Identification of Credit Card Fraud Using Machine Learning with Precision, Performance, and Challenges,” *Int. J. Innov. Sci. Res. Technol.*, vol. 10, no. 4, May 2025, doi: 10.38124/ijisrt/25apr1813.

[4] V. Verma, “Security Compliance and Risk Management in AI-Driven Financial Transactions,” *Int. J. Eng. Sci. Math.*, vol. 12, no. 7, pp. 107–121, 2023.

[5] B. Chaudhari and S. C. G. Verma, “Synergizing Generative AI and Machine Learning for Financial Credit Risk Forecasting and Code Auditing,” *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 11, no. 2, pp. 2882–2893, Apr. 2025, doi: 10.32628/CSEIT25112761.

[6] V. N. Dornadula and S. Geetha, “Credit Card Fraud Detection using Machine Learning Algorithms,” *Procedia Comput. Sci.*, vol. 165, pp. 631–641, 2019, doi: 10.1016/j.procs.2020.01.057.

[7] S. Pandya, “A Machine and Deep Learning Framework for Robust Health Insurance Fraud Detection and Prevention,” *Int. J. Adv. Res. Sci. Commun. Technol.*, Jul. 2023, doi: 10.48175/IJARSCT-14000U.

[8] R. Q. Majumder, “A Review of Anomaly Identification in Finance Frauds Using Machine Learning Systems,” *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 5, no. 10, pp. 101–110, Apr. 2025, doi: 10.48175/IJARSCT-25619.

[9] B. Chaudhari, S. C. G. Verma, and S. R. Somu, “Transforming Financial Lending: A Scalable Microservices Approach using AI and Spring Boot,” *Int. J. Sci. Res. Mod. Technol.*, vol. 3, pp. 72–81, Aug. 2024, doi: 10.38124/ijsrmt.v3i8.527.

[10] N. Malali, “AI Ethics in Financial Services : A Global Perspective,” *Int. J. Innov. Sci. Res. Technol.*, vol. 10, no. 2, 2025.

[11] J. K. Afriyie *et al.*, “A supervised machine learning algorithm for detecting and predicting fraud in credit card transactions,” *Decis. Anal. J.*, vol. 6, Mar. 2023, doi: 10.1016/j.dajour.2023.100163.

[12] V. Verma, “Deep Learning-Based Fraud Detection in Financial Transactions : A Case Study Using Real-Time Data Streams,” vol. 3, no. 4, pp. 149–157, 2023, doi: 10.56472/25832646/JETA-V3I8P117.

[13] R. P. Mahajan, “Development of Predictive Models for Early Detection of Alzheimer 's Disease Using Machine Learning,” *Int. J. Curr. Eng. Technol.*, vol. 15, no. 2, pp. 115–123, 2025.

[14] N. Prajapati, “The Role of Machine Learning in Big Data Analytics: Tools, Techniques, and Applications,” *ESP J. Eng. Technol. Adv.*, vol. 5, no. 2, 2025, doi: 10.56472/25832646/JETA-V5I2P103.

[15] J. Mishra, B. B. Biswal, and N. Padhy, “Machine Learning for Fraud Detection in Banking Cybersecurity Performance Evaluation of Classifiers and Their Real-Time Scalability,” in *2025 International Conference on Emerging Systems and Intelligent Computing (ESIC)*, IEEE, Feb. 2025, pp. 431–436. doi: 10.1109/ESIC64052.2025.10962752.

[16] L. H. Aros, L. X. B. Molano, F. Gutierrez-Portela, J. J. M. Hernandez, and M. S. R. Barrero, "Financial fraud detection through the application of machine learning techniques: a literature review," *Humanit. Soc. Sci. Commun.*, vol. 11, no. 1, p. 1130, Sep. 2024, doi: 10.1057/s41599-024-03606-0.

[17] B. Dharma and D. Latha, "Fraud Detection in Credit Card Transactional Data Using Hybrid Machine Learning Algorithm," in *2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)*, 2025, pp. 213–218. doi: 10.1109/ICMCSI64620.2025.10883549.

[18] D. Sharma, E. Kaur, N. Sharma, D. Rawat, S. P. Yadav, and M. Manwal, "Comparative Analysis of Machine Learning Algorithms on Credit Card Fraud Detection," in *2025 International Conference on Cognitive Computing in Engineering, Communications, Sciences and Biomedical Health Informatics (IC3ECSBHI)*, IEEE, Jan. 2025, pp. 147–150. doi: 10.1109/IC3ECSBHI63591.2025.10990921.

[19] A. Singh, K. S. Gill, M. Kumar, and R. Rawat, "Beyond Traditional Methods: Evaluating Advanced Machine Learning Models for Superior Fraud Detection," in *2024 4th International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS)*, 2024, pp. 297–300. doi: 10.1109/ICUIS64676.2024.10866102.

[20] U. Jabeen, K. Singh, and S. Vats, "Credit Card Fraud Detection Scheme Using Machine Learning and Synthetic Minority Oversampling Technique (SMOTE)," in *2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA)*, IEEE, Aug. 2023, pp. 122–127. doi: 10.1109/ICIRCA57980.2023.10220646.

[21] B. P. Verma, V. Verma, and A. Badholia, "Hyper-Tuned Ensemble Machine Learning Model for Credit Card Fraud Detection," in *2022 International Conference on Inventive Computation Technologies (ICICT)*, IEEE, Jul. 2022, pp. 320–327. doi: 10.1109/ICICT54344.2022.9850940.

[22] A. Shah and A. Mehta, "Comparative Study of Machine Learning Based Classification Techniques for Credit Card Fraud Detection," in *2021 International Conference on Data Analytics for Business and Industry (ICDABI)*, IEEE, Oct. 2021, pp. 53–59. doi: 10.1109/ICDABI53623.2021.9655848.

[23] E. Ileberi, Y. Sun, and Z. Wang, "Performance Evaluation of Machine Learning Methods for Credit Card Fraud Detection Using SMOTE and AdaBoost," *IEEE Access*, 2021, doi: 10.1109/ACCESS.2021.3134330.

[24] A. Singh, M. N. Halgamuge, and R. Lakshmiganthan, "Impact of Different Data Types on Classifier Performance of Random Forest, Naïve Bayes, and K-Nearest Neighbors Algorithms," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 12, 2017, doi: 10.14569/IJACSA.2017.081201.

[25] M. F. Erturul and M. E. Taluk, "A novel version of k nearest neighbor: Dependent nearest neighbor," *Appl. Soft Comput.*, vol. 55, pp. 480–490, Jun. 2017, doi: 10.1016/j.asoc.2017.02.020.

[26] A. Danades, D. Pratama, D. Anggraini, and D. Anggriani, "Comparison of accuracy level K-Nearest Neighbor algorithm and support vector machine algorithm in classification water quality status," in *Proceedings of the 2016 6th International Conference on System Engineering and Technology, ICSET 2016*, 2016. doi: 10.1109/FIT.2016.7857553.

[27] M. S. Kurt and E. Y. Demirel, "Web Page Classification With Deep Learning Methods," *Uludağ Univ. J. Fac. Eng.*, pp. 191–204, Mar. 2022, doi: 10.17482/uumfd.891038.

[28] M. N. Alatawi, "Detection of fraud in IoT-based credit card collected dataset using machine learning," *Mach. Learn. with Appl.*, vol. 19, p. 100603, Mar. 2025, doi: 10.1016/j.mlwa.2024.100603.

