2024 5th International Conference on Industrial Engineering and Artificial Intelligence (IEAI) | 979-8-3503-8636-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/IEAI62569.2024.00023

Fraud Detection in Online Credit Card Transactions Using Deep Learning

John Patrick J. Aquino
School of Information Technology
Mapúa University
Philippines
jpjaquino@mymail.mapua.edu.ph

Anna Patricia G. De Guia
School of Information Technology
Mapúa University
Philippines
apgdeguia@mymail.mapua.edu.ph

Danilo C. Dela Cruz
School of Information Technology
Mapúa University
Philippines
ddelacruzjr@mymail.mapua.edu.ph

Joel C. De Goma
School of Information Technology
Mapúa University
Philippines
jcdegoma@mapua.edu.ph

***Abstract*—This study focused on developing a more efficient hybrid fraud detection model than the hybrid model of the study it was based on using a combination of supervised and unsupervised algorithms, respectively Light Gradient Boosting Machine (LGBM) and Kernel Principal Component Analysis (KPCA), to tackle the temporal aspect of the data and better detect fraudulent activity in online credit card transactions. With the IEEE-CIS credit card fraud dataset, the researchers trained their model after utilizing preprocessing techniques such as categorical correlation, frequency encoders, and scalers to fully utilize the dataset in batches. The researchers made two versions of some models for thorough comparison and evaluation: Version 1 (V1) as a baseline model with standard parameters, and Version 2 (V2) with more updated data processing techniques and experimental parameters. Computing the Receiver Operating Characteristic-Area Under Curve (ROC-AUC) score in each model as the performance metric, the results show that compared to the ENS-XGB-LGBM model of the basis study with an ROC-AUC score of 0.951, the researchers have achieved their highest score of 0.95376 with their V2 IKPCA-LGBM model showing a noticeably improved fraud detection efficiency. The outcome shows that the V2 IKPCA-LGBM model with its dimensionality reduction and boosting algorithms surpasses the performance of the model of the basis study, however, the researchers recommended utilizing a balanced dataset, that further super parameter and hyperparameter tuning be done, as well as more time in testing the two versions of the models for them to be fully explored and improved upon.**

Keywords*—Fraud detection, hybrid model, credit card transaction, Light Gradient Boosting Machine (LGBM), Kernel Principal Component Analysis (KPCA)

I. Introduction

Bank technology in the 21st century is no exception to fraudulent activities that threaten the integrity and security of the system in safekeeping users' finances, driving the need to continuously develop and improve upon layers of security for online transactions. Credit cards are one of many mediums for online transactions, which is suitable to be examined for fraud detection. [1][12] Different fraud detection studies on credit card transactions acknowledged their struggle to provide a consistent highest-performing fraud detection model due to the temporal nature of transactional data such as the time of transaction, or the time interval of different transactions taking place in the same account. This paper, using [2] as the basis, aimed to create a better fraud detection model using a hybrid approach of supervised and unsupervised algorithms consisting of Light Gradient Boosting Machine (LGBM) and Kernel Principal Component Analysis (KPCA) for addressing the temporal nature of the dataset utilized resulting in better fraud detection in terms of Receiver Operating Characteristic-Area Under Curve (ROC-AUC) score. Along with different unsupervised algorithms [11], several approaches were made for the detection model.

II. Review of Related Literature

[3] The study developed a credit card fraud detection model using LGBM to handle large transaction datasets efficiently. Data cleaning, LGBM model training/testing, and hyperparameter tuning were performed, with performance compared to other models.

In a hybrid model approach [4], both supervised and supervised learning Extreme Gradient Boosting (XGB) and KPC respectively, were employed for personal loan fraud detection. The study, using a dataset from a Chinese bank, compared the hybrid model with XGB to a single XGB model, demonstrating superior performance by the hybrid model.

In [5], researchers presented a financial fraud detection system using XGB. The three-stage framework included feature engineering, model training, and evaluation of a Kaggle credit card transaction dataset. Results showed 99.6% accuracy, 98.6% precision, 96.6% recall, and a 97.6% F1 score. The study emphasized the XGB model's effectiveness but noted the use of an outdated dataset.

In another study [6], Khattri and Singh analyzed various fraud detection papers, compiling and summarizing commonly used techniques and parameters. Their findings indicated that transaction amount is the most utilized parameter, followed by location, for fraud detection.

Based on the different studies about and related to fraud detection in credit card transactions, performance improvement

979-8-3503-8636-3/24/$31.00 ©2024 IEEE
DOI 10.1109/IEAI62569.2024.00023

was shown in studies [2] and [4] wherein a hybrid approach was utilized. It can also be noted that these related studies are either comparing 2 or more algorithms with each other or utilizing 2 or more algorithms for their model to improve their performance.

III. Methodology

A. Dataset

The study utilizes a transaction dataset from Kaggle, sourced from the 2019 IEEE Computational Intelligence Society (IEEE-CIS) competition in partnership with Vesta Corporation. The dataset, aimed at enhancing fraud detection accuracy, consists of 'identity' and 'transaction' files joined by 'TransactionID.' The transaction file, containing (394) columns and (590541) rows, details overall transaction information. Categorical features include ProductCD, card1 - card6, addr1, addr2, P_emaildomain, R_emaildomain, and M1 - M9. The 'identity' file, with (41) columns and (144234) rows, encompasses identity information like network connection details and digital signatures, featuring categorical variables DeviceType, DeviceInfo, and id_12 - id_38.

Variable extraction or dropping is algorithm-dependent, and validated during the model testing phase. Feature engineering is employed for merging/grouping and cleaning unnecessary columns [5].

Due to the strict confidentiality of credit card and virtual wallet transactions, accessing recent transaction data requires authorization. The 2019 IEEE-CIS Fraud Detection Competition dataset, being the most recent open-source option, is utilized. The chosen evaluation metric is ROC-AUC, considered ideal for model assessment [5].

1) Temporal Data

This refers to the information associated with a specific time or sequence, capturing variations, trends, and patterns in chronological order. Our dataset contains D1-D15, these features are numeric time delta which contains ‘days between previous transactions and transaction intervals. Containing complex patterns over time.

As a solution, handling missing values was utilized first by using imputations on several features and carefully assessing the temporal data along the way. The process was then shifted to implementing Time-based feature engineering. It is utilized by extracting raw data into specific key features eg; 2020-03-20, 06:45:40 into Month: 3, Day: 20, Year: 2020, Hour: 06, or Day Name: Friday.

TABLE I. Time Feature Extraction (e.g. 2020-03-20, 06:45:40)

Month(M)	Day(D)	DayName(DN)	Year(Y)	Hour(H)
03	20	‘Friday’ (0-7)	2020	06

These can be then renamed into DT_D, DT_W, and so forth. Holidays are also considered in this feature extraction method. A normalization technique is then applied for feature scaling and adjusting feature weight to prevent overfitting. Afterward, an encoder such as Frequency or Timeblock Frequency was applied to handle categorical variables before feeding the processed data into KPCA or Principal Component Analysis (PCA).

It is worth noting that there are other approaches utilized such as Time series aggregation. This is implemented by using intervals hourly to daily to reduce noise and prioritize complex patterns. The detailed definitions are explained after this section for the techniques mentioned in Table I.

B. Procedures

This study enhances and extends [2]. However, due to limited information and no access to the coding repository in [2], variations exist in our methodology.

Data cleansing, preprocessing, and modeling were performed in Jupyter Notebook using Python libraries: numpy, pandas, scipy, matplotlib, seaborn, graphviz, and sklearn.

For the proposed hybrid model, LGBM served as the baseline algorithm, complemented by the unsupervised algorithm KPCA—an extension of traditional PCA. KPCA addresses the nonlinear nature of fraudulent patterns in datasets, overcoming limitations posed by PCA's linearity [7].

1) Versioning Report

The study utilized two versions for different approaches in the methods used for preprocessing, feature engineering, and modeling, here is its main key difference:

V1: Processed with standard feature engineering and conventional processing techniques. This also presents a baseline model configuration as structured according to [8].

V2: This version is done by using updated and modified data processing techniques with experimental key parameters. This presents a different application of feature engineering to provide enhancement to the ability of the model.

Both versions involve memory reduction, handling missing values, and similar preprocessing techniques. Both also performed feature engineering, including Unique Identifier (UID) aggregation and new feature generation.

2) Exploratory Data Analysis (EDA)

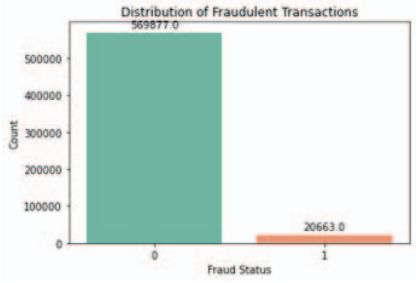

Fig. 1. Fraud Count in the Dataset

Fig. 1 highlights a highly imbalanced dataset, with 3.5% fraud transactions and the rest nonfraud, potentially causing overfitting. The missing values, also known as Not a Number (NaN) values, and Unique Values for train and test datasets were noted as 45.18% and 41.20%, respectively. To address the imbalance, the researchers employed evaluation metrics, parameter tuning, handling temporal features, and an encoder,

followed by training the model on standard 500 and 1000 samples. Additionally, of the 434 total columns, 414 were found to have missing values.

A bar graph was used to identify columns with the most null and unique values, revealing V columns as predominant in both datasets. Notably, V268, V91, V334, V301, V107, V2, and V108 contain abundant null values. The column with the most unique values is also predominantly populated by V columns, observed in both training and testing datasets. Addressing this, various feature engineering techniques and assertive data cleansing processes were implemented. Furthermore, with further analysis, there are NaN structures: V2-11, V12-34, V35-52, V53-74, V75-94, V95-131, V139-166, V167-216, V217-278 V279-317, V322-339.

Furthermore, the research also dropped features that have more than 90% missing values which are mostly from the ID column sets, as it is a common practice for most data cleansing because it improves interpretability and simplification of features. It also avoids bias and due to Its predictive power is close to being insignificant.

For V2, 82 features which are mostly from the ID features were therefore dropped as these features contain 90% missing values. Based on these observations, the problematic data features present in the dataset are coming from the V and ID columns. Most of these are unnecessary and redundant due to their observed data nature containing mostly null, NaN, and unique values. As a result, most of these features were dropped from the final dataset.

To maintain confidentiality, the meanings of V columns were deliberately kept vague [9]. To reveal correlations with the 'isFraud' column, a Seaborn library heatmap was utilized for feature correlation visualization.

The study applied correlation techniques and removed collinear features as stated in the methodology. Columns exhibiting a correlation coefficient of 0.75 or above were either dropped or strategically grouped, enhancing data clarity and model interpretability. Redundant features underwent systematic removal, incorporating type transformation, filling NaN values with constant imputations, detecting outliers, and applying Frequency encoding for categorical data and Timeblock Frequency Encoding for temporal data. These preprocessing steps collectively optimized the dataset for subsequent analyses, ensuring the integrity of data, the efficiency of memory usage, and the relevance of features to the modeling process.

3) Feature Engineering

The following feature engineering techniques were applied to extract relevant information, reduce dimensionality, and improve the model's ability to identify patterns associated with fraudulent activities, by the utilization of frequency encoding and aggregation frequency encoding, categorical correlation or collinearity features, normalization, or scaling, and UID aggregation.

4) Dimensionality Reduction Techniques Using PCA and KPCA

Various approaches were considered for dimensionality reduction, including the removal of collinear features, correlation techniques, and UID aggregation, however, since the study proposed a hybrid model by using LGBM paired with KPCA as previously mentioned, the study conducted dimensionality reduction by taking advantage of KPCA's ability to address challenges posed by the complex nature of fraud patterns in datasets.

IV. Results and Discussion

This chapter details the study's methodologies, including dataset description, features, procedures, and necessary undertakings. It also outlines the machine learning algorithm and tools, elaborating on model creation and dataset management approaches.

A. Model Testing

In the study's goal to comprehensively assess the model's capabilities, the approach was divided into two distinct versions: V1 and V2 as explained in the study's methodology.

The stratified k-fold cross-validation results [8], as shown in Table II included the number of bins to indicate how many subsets the data are divided during cross-validation, the instances of positive and negative classes to identify the number of Instances of credit card fraud and nonfraudulent transactions, and lastly, the number of data points for each fold to understand the underlying validation process.

TABLE II. Stratified K-Fold Cross-Validation Results (V1)

Fold #	Bins	Positive	Negative	Number of Data Points (train)
1	72,275	16,531	455,901	472,432
2	72,276	16,531	455,901	472,432
3	72,422	16,530	455,902	472,432
4	72,198	16,530	455,902	472,432
5	72,334	16,530	455,902	472,432

As an additional note, the results were only shown for V1 since the study employed k-fold validation exclusively for this version [8]. V2 did not undergo stratified k-fold cross-validation, as the standard split data function was utilized for this version to provide a simple and faster training time [10].

1) Super Parameters

All single algorithm models were tested by the prepared training and validation datasets and each model is trained with super parameters tuned for optimal performance. The training involves multiple iterations, each with a different seed value to account for variability. The parameters used by each single model algorithm are shown below:

For testing the different versions of the Hybrid Models, the study utilized GridSearch to determine the optimal parameters for the model. However, the researchers also manually assessed each super parameter for the tuning process. The researchers then analyzed what are the Implications of each super parameter and values to start on. XGB was tuned on default for model comparison analysis, and these parameters are on a V2 basis.

Hence, newer tuned parameters. In addition, as a basis for the values being set for each parameter [10] for proper tuning and manual tuning.

a) Current Study

In this section, the set values for each super parameter based on the Grid Search analysis and manual tuning procedure are shown in Tables III and IV.

TABLE III. LIGHTGBM SUPER PARAMETERS

Parameter	Parameter Description	Value
learning_rate	Step size for adjusting model weights during boosting	0.01
num_leaves	Maximum number of leaf nodes in a decision tree	255
max_bin	Maximum number of bins used to discretize numerical features	63
num_iterations	Number of boosting iterations or rounds	500
tree_learner	Type of tree learning algorithm to use	'Serial'
min_data_in_leaf	Minimum number of data points allowed in a leaf	1
min_sum_hessian_in_leaf	Minimum sum of Hessian	100
sparse_threshold	The threshold for treating features as sparse	1.0
bagging_freq	Frequency of bagging to introduce randomness in training	5
bagging_seed	Seed for reproducibility in random bagging	42
feature_fraction_seed	Seed for reproducibility in random feature selection	42
verbose	Controls the level of training output details	1
n_threads	Number of parallel threads used during training	-1
boosting_type	The type of boosting algorithm employed	'gbdt'
seed	Seed for reproducibility in random number generation	2

b) Basis Study

The basis of the study is the research's main reference as a goal point for the enhancement of the current study's Hybrid Model is shown in Tables V and VI. Song's model has a significantly higher number of boosting rounds (10,000 vs. 500). This might allow the model to capture more intricate patterns in the data, but it requires careful monitoring to avoid overfitting. Furthermore, Song's model has a specified tree depth of 12, while the study's model doesn't explicitly set this parameter. Deeper trees can capture more complex relationships but may also increase the risk of overfitting.

B. Comparison Analysis and Results

The comparison of ROC-AUC scores of each model is shown in Table VII and the hybrid model comparison with the basis study is shown in Table VIII.

1) Model Comparison

The study focused on the application of machine learning models for fraud detection, specifically utilizing LGBM and KPCA. Through an evaluation process such as employing a 5-fold cross-validation method and standard split-based method, the models were assessed based on the ROC-AUC.

TABLE IV. XGBOOST SUPER PARAMETERS (DEFAULT)

Parameter	Parameter Description	Value
n_estimators	Number of boosting rounds or trees to be built	100
n_jobs	Number of parallel threads used for training	-1
learning_rate	Step size for updates in boosting	0.3
max_bin	Maximum number of bins	256
early_stopping_rounds	Number of rounds before stopping	None
gamma	**Minimum loss reduction** needed to create a new split **on a leaf node**	0
max_depth	Maximum depth of a tree	6
max_leaves	Maximum number of leaves in a tree	0
verbose	Controls the level of training output details	1
random_state	Seed for random number generation	0

TABLE V. LIGHTGBM SUPER PARAMETERS (ZIJIAN SONG)

Parameter	Parameter Description	Value
n_estimators	Number of boosting rounds or trees to be built	10000
learning_rate	The step size shrinkage used in the update to prevent overfitting	0.01
bagging_fraction	Sample rate of rows	0.8
max_depth	Depth of the trees	12
feature_fraction	Sample rate of column	0.4
boosting_type	Type of boosting algorithm employed	'gbdt'

TABLE VI. XGBOOST SUPER PARAMETERS (ZIJIAN SONG)

Parameter	Parameter Description	Value
n_estimators	Number of boosting rounds or trees to be built	10000
learning_rate	The step size shrinkage used in the update to prevent overfitting	0.01
subsample	Sample rate of rows	0.8
max_depth	Depth of the trees	12
colsample_bytree	Sample rate of column	0.4
tree_method	Type of boosting algorithm employed	'gpu-hist'

TABLE VII. MODEL COMPARISON TABLE

Version	Model	ROC-AUC
V2	**IKPCA-LGBM**	**0.95376**
V2	IPCA-KPCA-LGBM	0.95368
V2	PCA-LGBM	0.95349
V1	Catboost	0.95212
V1	T-LGBM	0.94838
V1	IPCA-KPCA-LGBM	0.94601
V1	IKPCA-LGBM	0.94412
V2	IPCA-KPCA-XGB	0.94229
V2	IKPCA-XGB	0.94210
V2	PCA-XGB	0.94078

TABLE VIII. HYBRID MODEL COMPARISON TABLE

Model	ROC-AUC
IKPCA-LGBM	**0.95376**
ENS-XGB-LGBM	0.951

Among the different versions explored, the second iteration (V2) of the model stood out as the most effective configuration. By incorporating IKPCA, and LGBM, this version achieved an impressive ROC-AUC of 0.95376. This underscores the efficacy of integrating advanced dimensionality reduction techniques with a boosting algorithm in enhancing fraud detection accuracy.

PCA-LGBM also demonstrated competitive performance, particularly when combining PCA with LGBM, resulting in an ROC-AUC of 0.95349. The inclusion of Catboost in V1 further substantiated the robustness of the approach, yielding an ROC-AUC of 0.95212.

V. CONCLUSION AND RECOMMENDATION

As shown in Table XII, the researcher's Hybrid model gained a noticeable improvement of 0.953 ROC-AUC score, over the ensemble model of the basis study [2] which had a score of 0.951 ROC-AUC score. Thus, a 0.002 improvement. And V2 IKPCA-LGBM with a score of 0.95376 ROC-AUC which proved to be the highest score compared to other model combinations and baseline models.

In summary, the researcher's investigation into fraud detection models utilizing LGBM and KPCA emphasizes the significance of feature engineering, model selection, testing, and handling of temporal data.

Given the imbalanced nature of the acquired dataset and the inherent difficulty in acquiring open-sourced datasets due to the confidential nature of credit card transaction data, the study faced challenges. As a recommendation, it is suggested to test the model using a balanced dataset for a more comprehensive evaluation of its performance. In addition, the researchers suggest that more in-depth and precise model tuning is required to acquire a higher ROC-AUC score. There are a few key limitations such as time constraints present on V1, thus room for improvement.

Additionally, further model tuning is highly suggested to acquire a higher ROC-AUC score.

REFERENCES

[1] Benchaji, I., Douzi, S., El Ouahidi, B., and Jaafari, J., 2021, "Enhanced credit card fraud detection based on attention mechanism and LSTM deep model," 8, 1, December 2021, doi: https://doi.org/10.1186/s40537-021-00541-8.

[2] Song, Z. 2020, "A data mining based fraud detection hybrid algorithm in e-bank," 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), June 2020. DOI:https://doi.org/10.1109/icbaie49996.2020.00016, in press.

[3] Ge, D., Gu, J., Chang, S., and Cai, J. 2020, "Credit card fraud detection using lightgbm model," 2020 International Conference on E-Commerce and Internet Technology (ECIT), April 2020, DOI:https://doi.org/10.1109/ecit50008.2020.00060, in press.

[4] Wen, H. and Huang, F. 2020, "Personal loan fraud detection based on hybrid supervised and unsupervised learning," 2020 5th IEEE International Conference on Big Data Analytics (ICBDA), May 2020, DOI:https://doi.org/10.1109/icbda49040.2020.9101277, in press.

[5] Lei, S., Xu, K., Huang, Y., and Sha, X. 2020, "An xgboost based system for financial fraud detection," ResearchGate, April 10, 2023, in press.

[6] Khattri, V. and Singh, D. K. 2018, "Parameters of automated fraud detection techniques during online transactions, Emerald Insight, Journal of Financial Crime, vol. 25, no. 3, pp. 702–720, 2017, DOI: https://doi.org/10.1108//JFC.

[7] NIRPY Research, "PCA and kernel PCA explained • NIRPY research," NIRPY Research, Jun. 10, 2020, Retrieved from https://nirpyresearch.com/pca-kernel-pca-explained. *(Article)*

[8] H. Taemyung, "IEEE-CIS Fraud Detection | Kaggle," Kaggle, 2020, [Online], Retrieved from https://www.kaggle.com/competitions/ieee-fraud-detection/discussion/111696. *(Kaggle Online)*

[9] Lynn@Vesta, 2019, IEEE-CIS Fraud Detection, Oct. 2022, Retrieved from: https://www.kaggle.com/c/ieee-fraud detection/discussion/101203#latest-607486. *(Kaggle Online)*

[10] Fatih Fidan. 2021, IEEE-CIS Fraud Detection Notebook Guide, Retrieved from https://www.kaggle.com/code/kirshoff/fraud-detection-lightgbm-xgboost/notebook. *(Kaggle Online)*

[11] Maria R. Lepoivre, Chloé O. Avanzini, Guillaume Bignon, Loïc Legendre, and Aristide K. Piwele, "Credit Card Fraud Detection with Unsupervised Algorithms," Vol. 7, No. 1, pp. 34-38, February, 2016. doi: 10.12720/jait.7.1.34-38.

[12] Ibtissam Benchaji, Samira Douzi, and Bouabid El Ouahidi, "Credit Card Fraud Detection Model Based on LSTM Recurrent Neural Networks," Journal of Advances in Information Technology, Vol. 12, No. 2, pp. 113-118, May 2021. doi: 10.12720/jait.12.2.113-118.