Fraud Detection In Banking Data By Machine Learning Techniques

Kokala Sowmya **Dr. A. Lakshmi**
PG Student Assistant Professor
Department of IT (Data Science) Department of IT(Data
Science)
BVRIT Hyderabad College of Engineering For Women BVRIT Hyderabad College
of Engineering For Women
Bachupally, Hyderabad – 500090 Bachupally, Hyderabad -
500090
kokalasowmya@gmail.com
lakshmi.a@bvrithyderabad.edu.in

Abstract: *The study primarily centers on using machine learning methods to identify fraudulent activities in banking data. This is a critical concern in the financial sector, where it's essential to detect and prevent fraudulent transactions. To improve fraud detection, the study introduces class weight-tuning hyperparameters. These parameters help the model differentiate between legitimate and fraudulent transactions more effectively, enhancing the accuracy of the fraud detection system. The study strategically employs three popular machine learning algorithms: CatBoost, LightGBM, and XGBoost. Each algorithm has unique strengths, and their combined use aims to boost the overall performance of the fraud detection method. Deep learning techniques are integrated into the study to fine-tune hyperparameters. This integration enhances the performance and adaptability of the fraud detection system, making it more effective in identifying evolving fraud tactics. The project conducts thorough evaluations using real-world data. These evaluations reveal that the combined use of LightGBM and XGBoost outperforms existing methods when assessing various criteria. This indicates that the proposed approach is more effective at detecting fraudulent activities compared to other methods. It includes, a Stacking Classifier has been implemented, combining predictions from RandomForest and LightGBM classifiers with specific settings. This ensemble algorithm, utilizing a GradientBoostingClassifier as the final estimator, enhances prediction accuracy by leveraging the strengths of diverse models.*

Index terms *- Bayesian optimization, Data Mining, Deep Learning, Ensemble Learning, Hyper parameter, unbalanced data, Machine Learning.*

1. INTRODUCTION

In recent years, there has been a significant increase in the volume of financial transactions due to the expansion of financial institutions and the popularity of web-based e-commerce. Fraudulent transactions have become a growing problem in online banking, and fraud detection has always been challenging [1], [2]. Along with credit card development, the pattern of credit card fraud has always been updated. Fraudsters do their best to make it look legitimate, and credit card fraud has always been updated. Fraudsters do their best to make it look legitimate. They try to learn how fraud detection systems work and continue to stimulate these systems, making fraud detection more complicated. Therefore, researchers are constantly trying to find new ways or improve the performance of the existing methods [3].

People who commit fraud usually use security, control, and monitoring weaknesses in commercial applications to achieve their goals. However, technology can be a tool to combat fraud [4]. To prevent further possible fraud, it is important to detect the fraud right away after its occurrence [5]. Fraud can be defined as wrongful or criminal deception intended to result in financial or personal gain. Credit card fraud is related to the illegal use of credit card information for purchases in a physical or digital manner. In digital transactions, fraud can happen over the line or the web, since the cardholders usually provide the card number, expiration date, and card verification number by telephone or website [6].

There are two mechanisms, fraud prevention and fraud detection, that can be exploited to avoid fraud-related losses. Fraud prevention is a proactive method that stops fraud from happening in the first place. On the other hand, fraud detection is needed when a fraudster attempts a fraudulent transaction [7]. Fraud detection in banking is considered a binary classification problem in which data is classified as legitimate or fraudulent [8]. Because banking data is large in volume and with datasets containing a large amount of transaction data, manually reviewing and finding patterns for fraudulent transactions is either impossible or takes a long time. Therefore, machine learning-based algorithms play a pivotal role in fraud detection and prediction [9].

Machine learning algorithms and high processing power increase the capability of handling large datasets and fraud detection in a more efficient manner. [15] Machine learning algorithms and deep learning also provide fast and efficient solutions to real-time problems [10]. In this paper, we propose an efficient approach for detecting credit card fraud that has been evaluated on publicly available datasets and has used optimised algorithms LightGBM, XGBoost, CatBoost, and logistic regression individually, as well as majority voting combined methods, as well as deep learning and hyperparameter settings. An ideal fraud detection system should detect more fraudulent cases, and the precision of detecting fraudulent cases should be high, i.e., all results should be correctly detected, which will lead to the trust of customers in the bank, and on the other hand, the bank will not suffer losses due to incorrect detection.

2. LITERATURE SURVEY

Main challenge for e-commerce transaction fraud prevention is that fraud patterns are rather dynamic and diverse. [1] This paper introduces two innovative methods, fraud islands (link analysis) and multi-layer machine learning model [10, 15, 20], which can effectively tackle the challenge of detecting diverse fraud patterns. Fraud Islands are formed using link analysis to investigate the relationships between different fraudulent entities and to uncover the hidden complex fraud patterns through the formed network. Multi-layer model is used to deal with the largely diverse nature of fraud patterns. Currently, the fraud labels are determined through different channels which are banks' declination decision, manual review agents' rejection decisions, banks' fraud alert and customers' chargeback requests. It can be reasonably assumed that different fraud patterns could be caught though different fraud risk prevention forces (i.e. bank, manual review team and fraud machine learning model). The experiments showed that by integrating few different machine learning models which were trained using different types of fraud labels, the accuracy of fraud decisions can be significantly improved [10].

With the exponential rise in government and private health-supported schemes, the number of fraudulent billing cases is also increasing. [9] Detection of fraudulent transactions in healthcare systems is an exigent task due to intricate relationships among dynamic elements, including doctors, patients, and services. Hence, to introduce transparency in health support programs, there is a need to develop intelligent fraud detection models for tracing the loopholes in existing procedures, so that the fraudulent medical billing cases can be accurately identified. Moreover, there is also a need to optimize both the cost burden for the service provider and medical benefits for the client. [2] This paper presents a novel process-based fraud detection methodology to detect insurance claim-related frauds in the healthcare system using sequence mining concepts. Recent literature focuses on the amount-based analysis or medication versus disease sequential analysis rather than detecting frauds using sequence generation of services within each specialty. The proposed methodology generates frequent sequences with different pattern lengths. The confidence values and confidence level are computed for each sequence. The sequence rule engine generates frequent sequences along with confidence values for each hospital's specialty and compares them with the actual patient values [2, 7, 9]. This identifies anomalies as both sequences would not be compliant with the rule engine's sequences. The process-based fraud detection methodology is validated using last five years of a local hospital's transactional data that includes many reported cases of fraudulent activities.

With the continuous prosperity of the financial market, credit card volume has always been booming these years. The fraud businesses are also raising rapidly. Under this circumstance, fraud detection has become a more and more valuable problem. But the proportion of the fraud is absolutely much lower than the genius transaction, so the imbalance dataset makes this problem much more challenging. In this paper [3] we mainly tell how to cope with the credit card fraud detection problem by using boosting methods and also gave a contribution of the brief comparison between these boosting methods [29, 30].

Due to the immense growth of e-commerce and increased online based payment possibilities, credit card fraud has become deeply relevant global issue. Recently, there has been major interest for applying machine learning algorithms as data mining technique for credit card fraud detection. However, number of challenges appear, such as lack of publicly available data sets, highly imbalanced class sizes, variant fraudulent behavior etc. [5] In this paper we compare performance of three machine learning algorithms: Random Forest, Support Vector Machine and Logistic Regression in detecting fraud on real-life data containing credit card transactions [20]. To mitigate imbalanced class sizes, we use SMOTE sampling method. The problem of ever-changing fraud patterns is considered with employing incremental learning of selected ML algorithms in experiments. The performance of the techniques is evaluated based on commonly accepted metric: precision and recall.

Credit card fraud is a serious problem in financial services. Billions of dollars are lost due to credit card fraud every year. There is a lack of research studies on analyzing real-world credit card data owing to confidentiality issues. In this paper, machine learning algorithms [10, 15, 20] are used to detect credit card fraud. Standard models are first used. Then, hybrid methods which use AdaBoost and majority voting methods are applied. To evaluate the model efficacy, a publicly available credit card data set is used. [6]Then, a real-world credit card data set from a financial institution is analyzed. In addition, noise is added to the data samples to further assess the robustness of the algorithms. The experimental results positively indicate that the majority voting method achieves good accuracy rates in detecting fraud cases in credit cards.

Healthcare fraud is an expensive, white-collar crime in the United States, and it is not a victimless crime. Costs associated with fraud are passed on to the population in the form of increased premiums or serious harm to beneficiaries [2, 7]. There is an intense need for digital healthcare fraud detection systems to evolve in combating this societal threat. Due to the complex, heterogenic data systems and varied health models across the US, implementing digital advancements in healthcare is difficult. The end goal of healthcare fraud detection is to provide leads to the investigators that can then be inspected more closely with the possibility of recoupments, recoveries, or referrals to the appropriate authorities or agencies. In this article [7], healthcare fraud detection systems and methods found in the literature are described and summarized. A tabulated list of peer-reviewed articles in this research domain listing the main objectives, conclusions, and data characteristics is provided. The potential gaps identified in the implementation of such systems to real-world healthcare data will be discussed. The authors propose several research topics to fill these gaps for future researchers in this domain.

3. METHODOLOGY

i) Proposed Work:

The project introduces an advanced fraud detection system for banking data, utilizing machine learning techniques. It enhances its performance through class weight-tuning and Bayesian optimization, employing algorithms like [29, 30, 31, 32]CatBoost, LightGBM, and XGBoost. Deep learning further fine-tunes the system, and comprehensive evaluations using real-world data and key metrics ensure its effectiveness in identifying and preventing fraudulent activities. It includes, a Stacking Classifier has been implemented, combining predictions from RandomForest and LightGBM [17, 28] classifiers with specific settings. This ensemble algorithm, utilizing a GradientBoostingClassifier as the final estimator, enhances prediction accuracy by leveraging the strengths of diverse models. Additionally, a user-friendly Flask framework integrated with SQLite has been developed, featuring signup and signin functionalities for effective user testing and improving the system's accessibility and practicality in real-world fraud detection applications.

ii) System Architecture:

The system begins with raw data containing details of credit card transactions, including features and labels indicating fraud or legitimacy. The data undergoes preprocessing, involving feature extraction and selection, to prepare it for machine learning. The dataset is divided into two subsets: a training set for model development and a test set for performance evaluation. Bayesian optimization is used to fine-tune the hyperparameters of machine learning algorithms. Machine learning algorithms, such as CatBoost, [17] LightGBM, and XGBoost, are applied to the training data with the use of 5-fold cross-validation to ensure model robustness. We have also explored stacking classifier as an extension to the project. Various evaluation metrics are employed to assess the models' effectiveness in detecting credit card fraud while minimizing false positives.

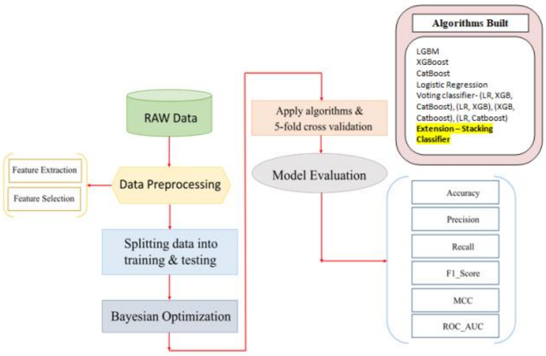
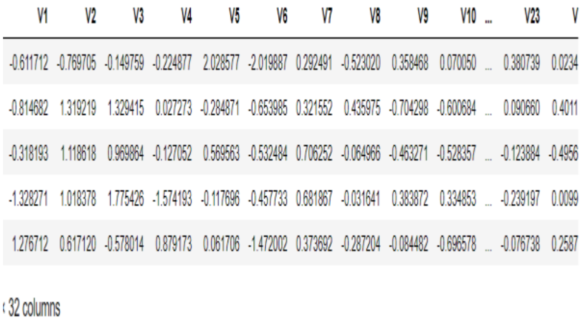

Algorithms Built
LGBM
XGBoost
CatBoost
Logistic Regression
Voting classifier- (LR, XGB, CatBoost), (LR, XGB), (XGB, Catboost), (LR, Catboost)
Extension – Stacking Classifier
RAW Data
Apply algorithms & 5-fold cross validation
Feature Extraction
Feature Selection
Data Preprocessing
Model Evaluation
Accuracy
Precision
Recall
F1_Score
MCC
ROC_AUC
Splitting data into training & testing
Bayesian Optimization

Fig 1 Proposed architecture

iii) Dataset collection:

CREDIT CARD FRAUD DATASET: We employed the Credit Card Fraud Detection dataset obtained from Kaggle to train machine learning algorithms. Initially, the dataset included a range of transaction-related attributes, including "Amount," "Time," and "V1" through "V28." For privacy and security reasons, specific details about these original features were withheld to protect sensitive information while still allowing for effective fraud detection training. So, these are the top 5 rows of the credit card fraud detection dataset. So, it contains 32 columns, we are displaying few of them here [6, 17].



V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	...	V23	V
-0.611712	-0.769705	-0.149759	-0.224877	2.028577	-2.019887	0.292491	-0.523020	0.358468	0.070050	...	0.380739	0.0234
-0.814682	1.319219	1.329415	0.027273	-0.284871	-0.653985	0.321552	0.435975	-0.704298	-0.600684	...	0.090660	0.4011
-0.318193	1.118618	0.969864	-0.127052	0.569563	-0.532484	0.706252	-0.064966	-0.463271	-0.528357	...	-0.123884	-0.4956
-1.328271	1.018378	1.775426	-1.574193	-0.117696	-0.457733	0.681867	-0.031641	0.383872	0.334853	...	-0.239197	0.0099
1.276712	0.617120	-0.578014	0.879173	0.061706	-1.472002	0.373692	-0.287204	-0.084482	-0.696578	...	-0.076738	0.2587

‹ 32 columns

Fig 2 NSL KDD dataset

iv) Data Processing:

Data processing involves transforming raw data into valuable information for businesses. Generally, data scientists process data, which includes collecting, organizing, cleaning, verifying, analyzing, and converting it into readable formats such as graphs or documents. Data processing can be done using three methods i.e., manual, mechanical, and electronic. The aim is to increase the value of information and facilitate decision-making. This enables businesses to improve their operations and make timely strategic decisions. Automated data processing solutions, such as computer software programming, play a significant role in this. It can help turn large amounts of data, including big data, into meaningful insights for quality management and decision-making.

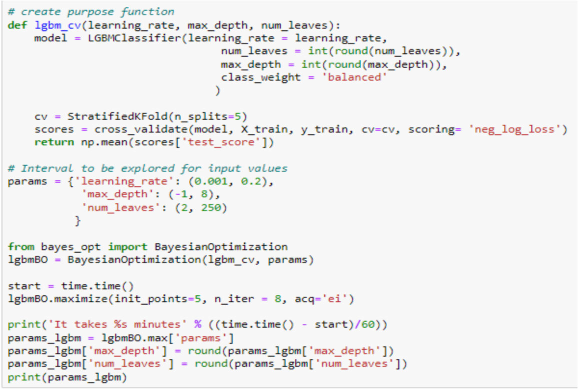
v) Feature selection:

Feature selection is the process of isolating the most consistent, non-redundant, and relevant features to use in model construction. Methodically reducing the size of datasets is important as the size and variety of datasets continue to grow. The main goal of feature selection is to improve the performance of a predictive model and reduce the computational cost of modeling.

Feature selection, one of the main components of feature engineering, is the process of selecting the most important features to input in machine learning algorithms. Feature selection techniques are employed to reduce the number of input variables by eliminating redundant or irrelevant features and narrowing down the set of features to those most relevant to the machine learning model. The main benefits of performing feature selection in advance, rather than letting the machine learning model figure out which features are most important.

vi) Algorithms:

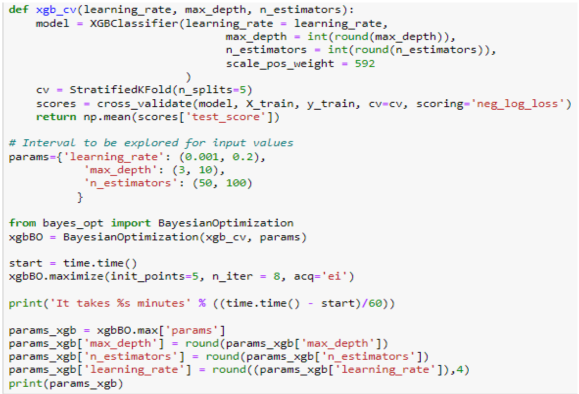
- **LGBM (Light Gradient Boosting Machine):** LGBM is a gradient boosting framework that is particularly efficient and performs well with large datasets. It's known for its speed and accuracy, making it suitable for tasks like fraud detection. LGBM builds an ensemble of decision trees, optimizing the boosting process for faster convergence [28].



```python
# create purpose function
def lgbm_cv(learning_rate, max_depth, num_leaves):
    model = LGBMClassifier(learning_rate = learning_rate,
                            num_leaves = int(round(num_leaves)),
                            max_depth = int(round(max_depth)),
                            class_weight = 'balanced'
                           )

    cv = StratifiedKFold(n_splits=5)
    scores = cross_validate(model, X_train, y_train, cv=cv, scoring= 'neg_log_loss')
    return np.mean(scores['test_score'])

# Interval to be explored for input values
params = {'learning_rate': (0.001, 0.2),
          'max_depth': (-1, 8),
          'num_leaves': (2, 250)
         }

from bayes_opt import BayesianOptimization
lgbmBO = BayesianOptimization(lgbm_cv, params)

start = time.time()
lgbmBO.maximize(init_points=5, n_iter = 8, acq='ei')

print('It takes %s minutes' % ((time.time() - start)/60))
params_lgbm = lgbmBO.max['params']
params_lgbm['max_depth'] = round(params_lgbm['max_depth'])
params_lgbm['num_leaves'] = round(params_lgbm['num_leaves'])
print(params_lgbm)
```

Fig 3 LGBM

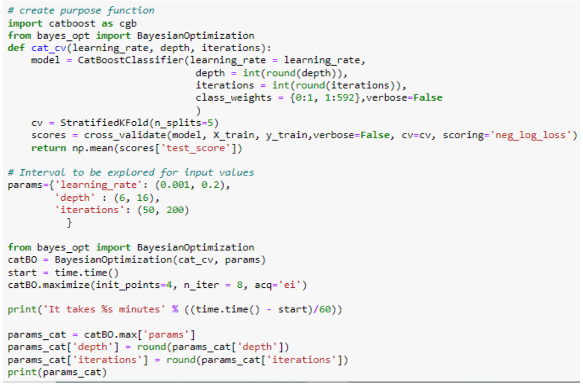
- **XGBoost (Extreme Gradient Boosting):** XGBoost is another gradient boosting algorithm that is widely used for various machine learning tasks. It's known for its robustness and performance. XGBoost uses a regularized gradient boosting framework and is effective in handling imbalanced datasets, which is crucial in fraud detection.



```python
def xgb_cv(learning_rate, max_depth, n_estimators):
    model = XGBClassifier(learning_rate = learning_rate,
                           max_depth = int(round(max_depth)),
                           n_estimators = int(round(n_estimators)),
                           scale_pos_weight = 592
                        )
    cv = StratifiedKFold(n_splits=5)
    scores = cross_validate(model, X_train, y_train, cv=cv, scoring='neg_log_loss')
    return np.mean(scores['test_score'])

# Interval to be explored for input values
params={'learning_rate': (0.001, 0.2),
           'max_depth': (3, 10),
           'n_estimators': (50, 100)
          }

from bayes_opt import BayesianOptimization
xgbBO = BayesianOptimization(xgb_cv, params)

start = time.time()
xgbBO.maximize(init_points=5, n_iter = 8, acq='ei')

print('It takes %s minutes' % ((time.time() - start)/60))

params_xgb = xgbBO.max['params']
params_xgb['max_depth'] = round(params_xgb['max_depth'])
params_xgb['n_estimators'] = round(params_xgb['n_estimators'])
params_xgb['learning_rate'] = round((params_xgb['learning_rate']),4)
print(params_xgb)
```

Fig 4 XGBoost

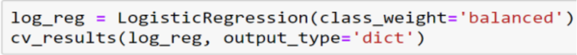
- **CatBoost (Categorical Boosting):** CatBoost is a gradient boosting library specifically designed to handle categorical features effectively. It automates the handling of categorical data, making it easier to work with such datasets. It's robust, handles overfitting well, and can be useful when dealing with real-world banking data [29, 30, 31, 32].



```python
# create purpose function
import catboost as cgb
from bayes_opt import BayesianOptimization
def cat_cv(learning_rate, depth, iterations):
    model = CatBoostClassifier(learning_rate = learning_rate,
                               depth = int(round(depth)),
                               iterations = int(round(iterations)),
                               class_weights = {0:1, 1:592},verbose=False
                               )
    cv = StratifiedKFold(n_splits=5)
    scores = cross_validate(model, X_train, y_train,verbose=False, cv=cv, scoring='neg_log_loss')
    return np.mean(scores['test_score'])

# Interval to be explored for input values
params={'learning_rate': (0.001, 0.2),
       'depth' : (6, 16),
       'iterations': (50, 200)
         }

from bayes_opt import BayesianOptimization
catBO = BayesianOptimization(cat_cv, params)
start = time.time()
catBO.maximize(init_points=4, n_iter = 8, acq='ei')

print('It takes %s minutes' % ((time.time() - start)/60))

params_cat = catBO.max['params']
params_cat['depth'] = round(params_cat['depth'])
params_cat['iterations'] = round(params_cat['iterations'])
print(params_cat)
```

Fig 5 Catboost

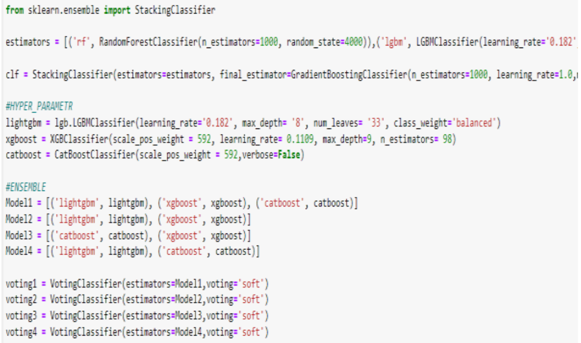
- **Logistic Regression:** Logistic Regression is a fundamental binary classification algorithm. While not as complex as ensemble methods like boosting, it serves as a baseline model for fraud detection. It's simple to understand and can provide insights into feature importance.



```
log_reg = LogisticRegression(class_weight='balanced')
cv_results(log_reg, output_type='dict')
```

Fig 6 Logistic regression

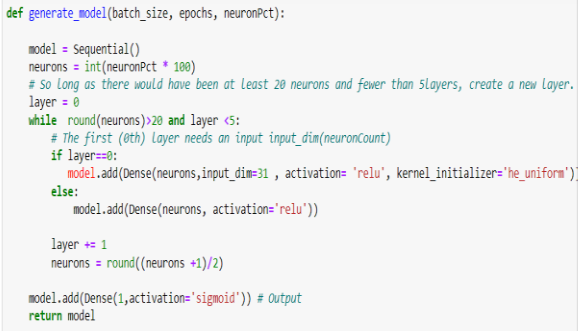
- **Voting Classifier:** The Voting Classifier combines the predictions of multiple machine learning models, such as Logistic Regression, XGBoost, and CatBoost, to make a final prediction. This ensemble technique leverages the collective intelligence of multiple models, often resulting in improved accuracy and robustness. We have built voting classifiers with different combinations of algorithms [19, 24].



```
from sklearn.ensemble import StackingClassifier

estimators = [('rf', RandomForestClassifier(n_estimators=1000, random_state=4000)),('lgbm', LGBMClassifier(learning_rate='0.182'

clf = StackingClassifier(estimators=estimators, final_estimator=GradientBoostingClassifier(n_estimators=1000, learning_rate=1.0,

#HYPER_PARAMETR
lightgbm = lgb.LGBMClassifier(learning_rate='0.182', max_depth= '8', num_leaves= '33', class_weight='balanced')
xgboost = XGBClassifier(scale_pos_weight = 592, learning_rate= 0.1109, max_depth=9, n_estimators= 98)
catboost = CatBoostClassifier(scale_pos_weight = 592,verbose=False)

#ENSEMBLE
Model1 = [('lightgbm', lightgbm), ('xgboost', xgboost), ('catboost', catboost)]
Model2 = [('lightgbm', lightgbm), ('xgboost', xgboost)]
Model3 = [('catboost', catboost), ('xgboost', xgboost)]
Model4 = [('lightgbm', lightgbm), ('catboost', catboost)]

voting1 = VotingClassifier(estimators=Model1,voting='soft')
voting2 = VotingClassifier(estimators=Model2,voting='soft')
voting3 = VotingClassifier(estimators=Model3,voting='soft')
voting4 = VotingClassifier(estimators=Model4,voting='soft')
```

Fig 7 Voting classifier

- **Neural Network:** A Neural Network is a deep learning model inspired by the human brain. In this context, it can capture complex patterns and relationships in the data. Neural Networks are used for their ability to learn intricate fraud patterns, especially in large datasets.

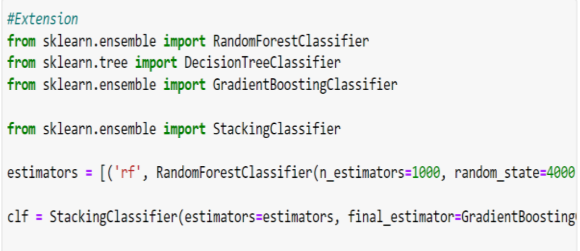


```
def generate_model(batch_size, epochs, neuronPct):

    model = Sequential()
    neurons = int(neuronPct * 100)
    # So long as there would have been at least 20 neurons and fewer than 5layers, create a new layer.
    layer = 0
    while  round(neurons)>20 and layer <5:
        # The first (0th) layer needs an input input_dim(neuronCount)
        if layer==0:
           model.add(Dense(neurons,input_dim=31 , activation= 'relu', kernel_initializer='he_uniform')
        else:
            model.add(Dense(neurons, activation='relu'))

        layer += 1
        neurons = round((neurons +1)/2)

    model.add(Dense(1,activation='sigmoid')) # Output
    return model
```

Fig 8 Neural network

- **Stacking classifier:** as an extension we have built a stacking classifier.

The Stacking Classifier, an ensemble algorithm, merges predictions from two base classifiers (RandomForest and LightGBM) with specific settings. It employs a GradientBoostingClassifier as the final estimator, enhancing prediction accuracy by blending the strengths of diverse models in ensemble learning.



```
#Extension
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import GradientBoostingClassifier

from sklearn.ensemble import StackingClassifier

estimators = [('rf', RandomForestClassifier(n_estimators=1000, random_state=4000

clf = StackingClassifier(estimators=estimators, final_estimator=GradientBoosting
```

Fig 9 Stacking classifier

4. EXPERIMENTAL RESULTS

Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

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

Accuracy: Accuracy is the proportion of correct predictions in a classification task, measuring the overall correctness of a model's predictions.

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

F1 Score: The F1 Score is the harmonic mean of precision and recall, offering a balanced measure that considers both false positives and false negatives, making it suitable for imbalanced datasets.

$$\text{F1 Score} = 2 * \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} * 100$$

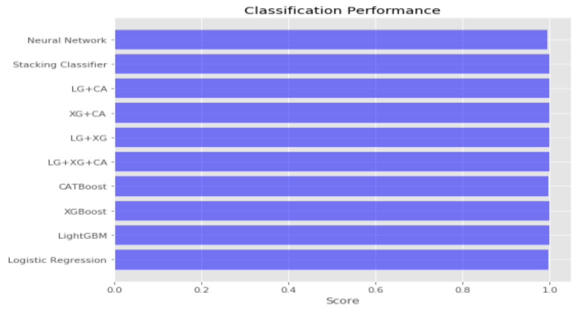
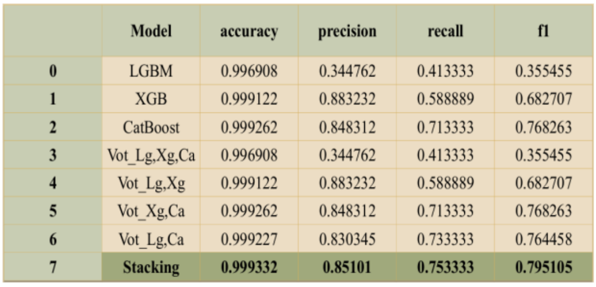
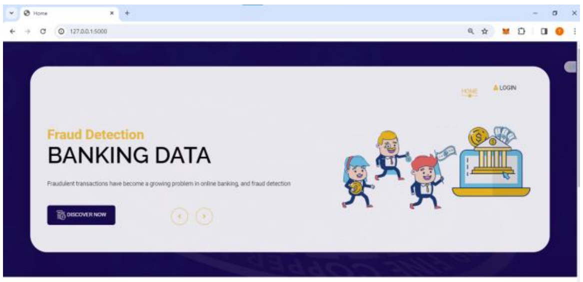


Fig 10 Graph comparing accuracy of NSL-KDD dataset



	Model	accuracy	precision	recall	f1
0	LGBM	0.996908	0.344762	0.413333	0.355455
1	XGB	0.999122	0.883232	0.588889	0.682707
2	CatBoost	0.999262	0.848312	0.713333	0.768263
3	Vot_Lg,Xg,Ca	0.996908	0.344762	0.413333	0.355455
4	Vot_Lg,Xg	0.999122	0.883232	0.588889	0.682707
5	Vot_Xg,Ca	0.999262	0.848312	0.713333	0.768263
6	Vot_Lg,Ca	0.999227	0.830345	0.733333	0.764458
7	**Stacking**	**0.999332**	**0.85101**	**0.753333**	**0.795105**

Fig 11 Performance Assessment Table

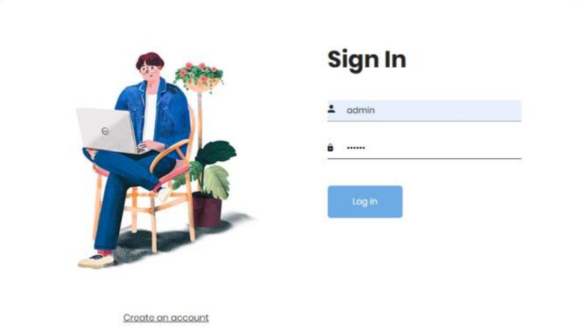


Fraud Detection

BANKING DATA

Fraudulent transactions have become a growing problem in online banking, and fraud detection

Fig 12 Home page

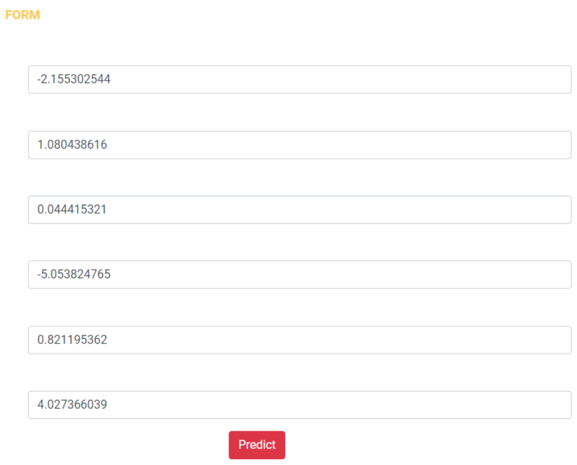


Sign In

admin

••••••

Fig 13 Signin page



FORM

-2.155302544

1.080438616

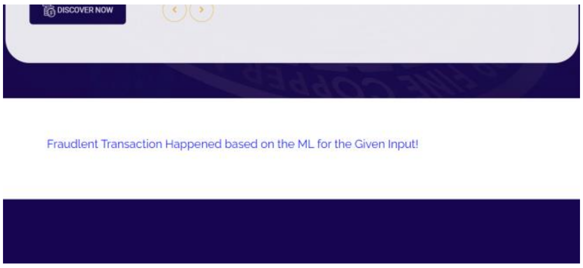
0.044415321

-5.053824765

0.821195362

4.027366039

Fig 14 User input



Fraudlent Transaction Happened based on the ML for the Given Input!

Fig 15 Predict result for given input

5. CONCLUSION

The Stacking Classifier stood out by achieving the highest accuracy among all models, demonstrating its remarkable performance in fraud detection. The project showcased robust performance across a variety of machine learning models, including LightGBM, XGBoost, CatBoost [29, 30, 31, 32], voting classifiers and neural networks, highlighting its adaptability. The utilization of diverse sampling and scaling techniques significantly contributed to improved fraud detection accuracy, emphasizing their importance. Applying the ensemble method, Stacking Classifier, significantly boosted fraud detection accuracy, emphasizing its effectiveness. The creation of a user-friendly Flask front-end streamlines user testing and authentication, ensuring accessibility and practicality. The system's testing in Flask, where input was provided, validates its functionality and user experience. [1, 2, 3] The project's results demonstrate the potential of advanced machine learning techniques in addressing fraud detection challenges within the banking sector, paving the way for future applications. The project's outcomes create opportunities for continuous improvement by exploring additional ensemble techniques and optimization strategies. Ultimately, the project's results benefit the banking industry by bolstering fraud detection capabilities, reducing financial losses, and ensuring secure transactions, enhancing overall security and trust.

6. FUTURE SCOPE

Future research will explore combining additional hybrid models with CatBoost [29] to enhance fraud detection accuracy and robustness. Future work will fine-tune CatBoost's hyperparameters, with a specific focus on optimizing the number of trees to boost the model's efficiency [33]. Research will focus on strategies to adapt to ever-changing fraud patterns, ensuring the model remains effective in identifying emerging fraudulent activities. Ongoing research aims to incorporate real-time data for improved system responsiveness and adaptability, enabling quicker responses to emerging threats. Future efforts will work on making the model's decision-making process more understandable, providing deeper insights into its reasoning for building trust and improving fraud detection strategies.

REFERENCES

[1] J. Nanduri, Y.-W. Liu, K. Yang, and Y. Jia, ‘‘Ecommerce fraud detection through fraud islands and multi-layer machine learning model,’’ in Proc. Future Inf. Commun. Conf., in Advances in Information and Communication. San Francisco, CA, USA: Springer, 2020, pp. 556–570.

[2] I. Matloob, S. A. Khan, R. Rukaiya, M. A. K. Khattak, and A. Munir, ‘‘A sequence mining-based novel architecture for detecting fraudulent transactions in healthcare systems,’’ IEEE Access, vol. 10, pp. 48447–48463, 2022.

[3] H. Feng, ‘‘Ensemble learning in credit card fraud detection using boosting methods,’’ in Proc. 2nd Int. Conf. Comput. Data Sci. (CDS), Jan. 2021, pp. 7–11.

[4] M. S. Delgosha, N. Hajiheydari, and S. M. Fahimi, ‘‘Elucidation of big data analytics in banking: A four-stage delphi study,’’ J. Enterprise Inf. Manage., vol. 34, no. 6, pp. 1577–1596, Nov. 2021.

[5] M. Puh and L. Brki¢, ‘‘Detecting credit card fraud using selected machine learning algorithms,’’ in Proc. 42nd Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO), May 2019, pp. 1250–1255.

[6] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi, ‘‘Credit card fraud detection using AdaBoost and majority voting,’’ IEEE Access, vol. 6, pp. 14277–14284, 2018.

[7] N. Kumaraswamy, M. K. Markey, T. Ekin, J. C. Barner, and K. Rascati, ‘‘Healthcare fraud data mining methods: A look back and look ahead,’’ Perspectives Health Inf. Manag., vol. 19, no. 1, p. 1, 2022.

[8] E. F. Malik, K. W. Khaw, B. Belaton, W. P. Wong, and X. Chew, ‘‘Credit card fraud detection using a new hybrid machine learning architecture,’’ Mathematics, vol. 10, no. 9, p. 1480, Apr. 2022.

[9] K. Gupta, K. Singh, G. V. Singh, M. Hassan, G. Himani, and U. Sharma, ‘‘Machine learning based credit card fraud detection—A review,’’ in Proc. Int. Conf. Appl. Artif. Intell. Comput. (ICAAIC), 2022, pp. 362–368.

[10] R. Almutairi, A. Godavarthi, A. R. Kotha, and E. Ceesay, ‘‘Analyzing credit card fraud detection based on machine learning models,’’ in Proc. IEEE Int. IoT, Electron. Mechatronics Conf. (IEMTRONICS), Jun. 2022, pp. 1–8.

[11] N. S. Halvaiee and M. K. Akbari, ‘‘A novel model for credit card fraud detection using artificial immune systems,’’ Appl. Soft Comput., vol. 24, pp. 40–49, Nov. 2014.

[12] A. C. Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, ‘‘Feature engineering strategies for credit card fraud detection,’’ Expert Syst. Appl., vol. 51, pp. 134–142, Jun. 2016.

[13] U. Porwal and S. Mukund, ‘‘Credit card fraud detection in e-commerce: An outlier detection approach,’’ 2018, arXiv:1811.02196.

[14] H. Wang, P. Zhu, X. Zou, and S. Qin, ‘‘An ensemble learning framework for credit card fraud detection based on training set partitioning and clustering,’’ in Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov. (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Oct. 2018, pp. 94–98.

[15] F. Itoo, M. Meenakshi, and S. Singh, ‘‘Comparison and analysis of logistic regression, Naïve Bayes and knn machine learning algorithms for credit card fraud detection,’’ Int. J. Inf. Technol., vol. 13, no. 4, pp. 1503–1511, 2021.

[16] T. A. Olowookere and O. S. Adewale, ‘‘A framework for detecting credit card fraud with cost-sensitive meta-learning ensemble approach,’’ Sci. Afr., vol. 8, Jul. 2020, Art. no. e00464.

[17] A. A. Taha and S. J. Malebary, ‘‘An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine,’’ IEEE Access, vol. 8, pp. 25579–25587, 2020.

[18] X. Kewei, B. Peng, Y. Jiang, and T. Lu, ‘‘A hybrid deep learning model for online fraud detection,’’ in Proc. IEEE Int. Conf. Consum. Electron. Comput. Eng. (ICCECE), Jan. 2021, pp. 431–434.

[19] T. Vairam, S. Sarathambekai, S. Bhavadharani, A. K. Dharshini, N. N. Sri, and T. Sen, ‘‘Evaluation of Naïve Bayes and voting classifier algorithm for credit card fraud detection,’’ in Proc. 8th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS), Mar. 2022, pp. 602–608.

[20] P. Verma and P. Tyagi, ‘‘Analysis of supervised machine learning algorithms in the context of fraud detection,’’ ECS Trans., vol. 107, no. 1, p. 7189, 2022.

[21] J. Zou, J. Zhang, and P. Jiang, ‘‘Credit card fraud detection using autoencoder neural network,’’ 2019, arXiv:1908.11553.

[22] D. Almhaithawi, A. Jafar, and M. Aljnidi, ‘‘Example-dependent costsensitive credit cards fraud detection using SMOTE and Bayes minimum risk,’’ Social Netw. Appl. Sci., vol. 2, no. 9, pp. 1–12, Sep. 2020.

[23] J. Cui, C. Yan, and C. Wang, ‘‘Learning transaction cohesiveness for online payment fraud detection,’’ in Proc. 2nd Int. Conf. Comput. Data Sci., Jan. 2021, pp. 1–5.

[24] M. Rakhshaninejad, M. Fathian, B. Amiri, and N. Yazdanjue, ‘‘An ensemble-based credit card fraud detection algorithm using an efficient voting strategy,’’ Comput. J., vol. 65, no. 8, pp. 1998–2015, Aug. 2022.

[25] A. H. Victoria and G. Maragatham, ‘‘Automatic tuning of hyperparameters using Bayesian optimization,’’ Evolving Syst., vol. 12, no. 1, pp. 217–223, Mar. 2021.

[26] H. Cho, Y. Kim, E. Lee, D. Choi, Y. Lee, and W. Rhee, ‘‘Basic enhancement strategies when using Bayesian optimization for hyperparameter tuning of deep neural networks,’’ IEEE Access, vol. 8, pp. 52588–52608, 2020.

[27] F. N. Khan, A. H. Khan, and L. Israt, ‘‘Credit card fraud prediction and classification using deep neural network and ensemble learning,’’ in Proc. IEEE Region 10 Symp. (TENSYMP), Jun. 2020, pp. 114–119.

[28] W. Liang, S. Luo, G. Zhao, and H. Wu, ‘‘Predicting hard rock pillar stability using GBDT, XGBoost, and LightGBM algorithms,’’ Mathematics, vol. 8, no. 5, p. 765, May 2020.

[29] S. B. Jabeur, C. Gharib, S. Mefteh-Wali, and W. B. Arfi, ‘‘CatBoost model and artificial intelligence techniques for corporate failure prediction,’’ Technol. Forecasting Social Change, vol. 166, May 2021, Art. no. 120658.

[30] J. Hancock and T. M. Khoshgoftaar, ‘‘Medicare fraud detection using CatBoost,’’ in Proc. IEEE 21st Int. Conf. Inf. Reuse Integr. Data Sci. (IRI), Aug. 2020, pp. 97–103.

[31] B. Dhananjay and J. Sivaraman, ‘‘Analysis and classification of heart rate using CatBoost feature ranking model,’’ Biomed. Signal Process. Control, vol. 68, Jul. 2021, Art. no. 102610.

[32] Y. Chen and X. Han, ‘‘CatBoost for fraud detection in financial transactions,’’ in Proc. IEEE Int. Conf. Consum. Electron. Comput. Eng. (ICCECE), Jan. 2021, pp. 176–179.

[33] A. Goyal and J. Khiari, ‘‘Diversity-aware weighted majority vote classifier for imbalanced data,’’ in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Jul. 2020, pp. 1–8.

[34] A. Roy, J. Sun, R. Mahoney, L. Alonzi, S. Adams, and P. Beling, ‘‘Deep learning detecting fraud in credit card transactions,’’ in Proc. Syst. Inf. Eng. Design Symp. (SIEDS), Apr. 2018, pp. 129–134.