IJITCE

ISSN 2347- 3657

International Journal of Information Technology & Computer Engineering

Email : ijitce.editor@gmail.com or editor@ijitce.com

International Journal of
Information Technology & Computer Engineering

ISSN 2347–3657

Volume 12, Issue 2, 2024

PREDICTION OF FRAUD IN BANKING DATA BY USING MACHINE LEARNING TECHNIQUES: STACKING CLASSIFIER

Authors:

1) Dr. K. Kranthi Kumar
Associate professor
Department of IT
SNIST,Telangana,India.
kranthikumark@sreenidhi.edu.in

2) Mrs. B. Hema Kumari
Assistant professor
Department of IT
SNIST,Telangana,India.
Hema0550@gmail.com

3) M. Gopi Chand
Department of IT
SNIST,Telangana,India.
malisetti.gopi12@gmail.com

4) A. Deekshith
Department of IT
SNIST,Telangana,India.
ailadeekshith133@gmail.com

5) B. Sai Charan
Department of IT
SNIST,Telangana,India.
barucharan85@gmail.com

Abstract: The main focus of the research is on detecting fraudulent actions in financial data using machine learning techniques. The detection and prevention of fraudulent transactions is of the utmost importance in the financial industry, making this a significant problem. The research presents hyperparameters for class weight tweaking with the goal of improving fraud detection. By adjusting these parameters, the model is able to better distinguish between real and fraudulent transactions, which improves the system's ability to identify fraud. Three well-known machine learning algorithms—CatBoost, LightGBM, and XGBoost—are strategically used in the research. The goal of combining these algorithms is to improve the fraud detection approach as a whole by capitalizing on their individual capabilities.In order to optimize hyperparameters, the research incorporates deep learning approaches. The fraud detection system becomes more efficient and flexible as a result of this integration, allowing it to better detect developing fraud strategies. Using real-world data, the project does comprehensive assessments. According to these tests, when compared to other approaches, the one that combines LightGBM with XGBoost performs better across the board. This proves that the suggested strategy outperforms the alternatives when it comes to identifying fraudulent actions. One of its features is a Stacking Classifier that takes into account both RandomForest and LightGBM classifier predictions while taking certain parameters into account. This ensemble method improves prediction accuracy by combining the best features of many models; the final estimator is a GradientBoostingClassifier.

Hyperparameter, data imbalance, machine learning, Bayesian optimization, deep learning, ensemble learning, and data mining are some of the index phrases.

1. INTRODUCTION

2. The proliferation of banks and the rise of online shopping have both contributed to a dramatic increase in the number of monetary transactions in recent years. Online banking has seen an increase in fraudulent transactions, and detecting such activity has proven difficult in the past [1, 2]. There has always been a new pattern to credit card theft that follows the evolution of credit cards. Credit card theft has always evolved, and con artists strive to make their work seem authentic. Con artists make every

effort to make it seem authentic. They keep stimulating these systems in an effort to understand how they identify fraud, which makes fraud detection more difficult. So, scientists are always looking for new approaches or ways to make the current ones work better [3]. Commercial programs often have security, management, and monitoring flaws that fraudsters exploit. But technology may also help fight fraud [4]. An early detection of fraud is critical in preventing its recurrence [5]. Intentional and unlawful use of deceit for monetary or personal benefit is known as fraud. Whether done online or in-store, credit card fraud occurs when someone uses your card details without your permission to make a transaction. Since cards often provide the number, expiry date, and verification number over the phone or online, fraud may occur during digital transactions [6]. Losses due to fraud may be mitigated via the use of two mechanisms: fraud detection and fraud prevention. Preventing fraud from occurring is the primary goal of fraud prevention strategies. However, in the event that a fraudster attempts to conduct a fraudulent transaction, fraud detection becomes necessary. [7]. Data must be categorized as either valid or fraudulent in order for fraud detection in banking to be seen as a binary classification issue [8]. Finding patterns for fraudulent transactions manually is either difficult or takes a long time due to the vast number of financial data and datasets that include a big quantity of transaction data. Consequently, algorithms that rely on machine learning are crucial for detecting and predicting fraud [9]. The capacity to efficiently manage massive datasets and identify fraud is enhanced by machine learning algorithms and powerful processing capabilities. [15] Additionally, deep learning and machine learning algorithms provide quick and effective answers to issues that arise in real time [10]. Using optimized algorithms such as LightGBM, XGBoost, CatBoost, and logistic regression separately, majority voting combined methods, deep learning, and hyperparameter settings, we present an effective method for detecting credit card fraud in this paper. The method has been tested on publicly available datasets. More fraudulent cases should be detected by an ideal fraud detection system, and the accuracy of those cases should be high; in other words, all results should be correctly detected. This will go a long way toward earning customers' trust and preventing the bank from losing money because of false positives.

3. LITERATURE SURVEY

4. The fact that fraud patterns are so varied and ever-changing is the biggest obstacle to preventing fraud in online transactions. [1] This work presents two new approaches to the problem of fraud pattern detection: fraud islands (link analysis) and a multi-layer machine learning model [10, 15, 20]. To find hidden complex fraud patterns in a network, researchers use link analysis to create "Fraud Islands" and study the connections between various fraudulent organizations. Because fraud patterns are so varied, a multi-layer paradigm is necessary for their handling. There are now several routes that contribute to the determination of fraud labels. These include the decision-making process inside banks, the rejection choices made by human review agents, the fraud alerts generated by banks, and the chargeback requests made by consumers. The bank, the human review team, and the fraud machine learning model are all potential fraud risk prevention forces, and it is reasonable to believe that they may detect distinct fraud patterns. The results of the tests demonstrated that the accuracy of fraud choices may be greatly enhanced by combining a small number of machine learning models that were trained using

various fraud labels [10]. Cases of fraudulent invoicing are on the increase with the exponential growth of health-supporting programs funded by both the public and commercial sectors. [9] Because there are so many moving parts and interdependent variables in healthcare systems—including providers, patients, and services—the detection of fraudulent transactions is a top priority. Therefore, in order to bring accountability to health assistance programs, it is necessary to create smart fraud detection models that can identify fraudulent medical billing instances by identifying the gaps in current processes. In addition, it is important to maximize the client's medical advantages while minimizing the service provider's financial burden. [2] Using sequence mining principles, this research introduces a new process-based fraud detection approach for healthcare insurance claims fraud detection. In place of identifying frauds via the production of service sequences within each specialty, recent study has focused on amount-based analysis or medicine versus illness sequential analysis. Regular sequences of varying pattern lengths are produced using the suggested technique. Each sequence has its own set of confidence values and a corresponding degree of confidence. A comparison is made between the actual patient values and the often occurring sequences and confidence values for each hospital's specialty that are generated by the sequence rule engine [2, 7, 9]. Since these two sequences don't match up with the rule engine's sequences, it finds out when anything is wrong. A local hospital's transactional data from the previous five years, which contains several reported occurrences of fraud, is used to verify the process-based fraud detection technique.

As the economy and stock market have continued to thrive, the use of credit cards has also been on the rise. There has been a corresponding uptick in the scam enterprises. Given this context, detecting fraud has grown in importance. This task is made considerably more difficult by the imbalanced dataset, because the fraction of fraud is far smaller than the genius transaction. In this research, we primarily discuss how to use boosting approaches to deal with the credit card fraud detection issue. We also provide a short comparison of different boosting methods [29, 30]. As the number of online stores and payment methods continues to skyrocket, credit card theft has emerged as a major concern on a worldwide scale. Credit card fraud detection using machine learning algorithms as a data mining approach has recently attracted a lot of attention. But then a lot of problems arise, such unequal class sizes, different types of fraud, and a dearth of publicly accessible data sets. [5] In this study, we evaluate the efficacy of three ML algorithms—Logistic Regression, Random Forest, and Support Vector Machine—in identifying fraudulent activities using actual credit card transaction data [20]. Our use of the SMOTE sampling approach helps to reduce the impact of unequal class sizes. Using incremental learning of chosen ML algorithms in tests, the issue of constantly evolving fraud tendencies is taken into account. Two widely used metrics, recall and precision, are used to assess the methods' efficacy. The financial services industry has a major issue with credit card fraud. Annually, credit card fraud costs businesses and consumers billions of dollars. Confidentiality concerns have prevented several studies from examining actual credit card data. This work presents a method for detecting credit card fraud using machine learning techniques [10, 15, 20]. At initially, we utilize standard models. Subsequently, systems that combine AdaBoost with majority voting techniques are used. The effectiveness of the model is tested using a credit card dataset that is accessible to the public. [6]After that, we look at a credit card data collection that a bank really has. Also, to make sure the algorithms are as strong as possible, we introduce noise to the data samples. The majority voting mechanism successfully detects credit card theft at high rates, according to the testing data.

In the US, healthcare fraud is a costly white-collar crime that affects many people. The public pays for fraud either via higher premiums or catastrophic damage to recipients [2, 7]. To counter this social danger, digital healthcare fraud detection technologies must rapidly advance. Digital healthcare innovations are challenging to deploy due to the heterogeneity and complexity of data systems and health models in the United States. In healthcare fraud detection, the ultimate purpose is to offer investigators leads that may be further investigated in the hopes of recovering losses, recovering funds, or reporting the case to the proper authorities. Systematic reviews and summaries of healthcare fraud detection techniques are provided in this article [7]. The following is a table containing a list of peer-reviewed publications that have been published in this field of study. Each article includes an abstract, major points, conclusions, and data attributes. The possible problems that may arise when using these technologies to actual healthcare data will be covered. To address these deficiencies, the authors suggest other areas for further study in this field.

5. METHODOLOGY

i) Proposed Work:

Using machine learning methods, the project presents a state-of-the-art solution for detecting fraud in financial data. Class weight-tuning and Bayesian optimization, using methods such as [29, 30, 31, 32]CatBoost, LightGBM, and XGBoost, improve its performance. To make sure it can detect and prevent fraudulent activity, the system is evaluated thoroughly using real-world data and important metrics, and deep learning is used to further fine-tune it. Included in it is a Stacking Classifier that, given certain parameters, combines the predictions of RandomForest and LightGBM [17, 28]. This ensemble method improves prediction accuracy by combining the best features of many models; the final estimator is a GradientBoostingClassifier. To further enhance the system's usability and usefulness in real-world fraud detection apps, we have designed a user-friendly Flask framework that is linked with SQLite. This framework has signup and signin functions, which allow for efficient user testing.

ii) Architecture of the System: Credit card transaction information, including characteristics and labels that indicate authenticity or fraud, are input into the system as raw data. In order to get the data ready for machine learning, it is preprocessed using methods like feature extraction and selection. A training set is used for developing models, while a test set is used to evaluate their performance. The dataset is separated into these two parts. To make ML algorithms run more smoothly, hyperparameters are fine-tuned via Bayesian optimization. To guarantee the model's resilience, machine learning methods including XGBoost, CatBoost, LightGBM, and [17] are used to the training data using 5-fold cross-validation. We have also investigated the possibility of including a stacking classifier into the project. To measure how well the algorithms identify credit card fraud with few false positives, we use a variety of assessment indicators.

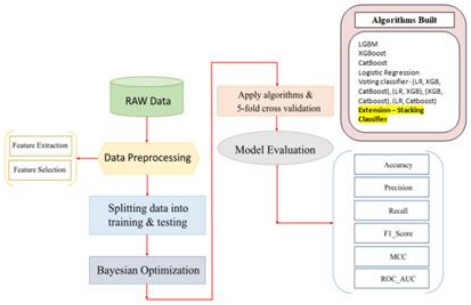
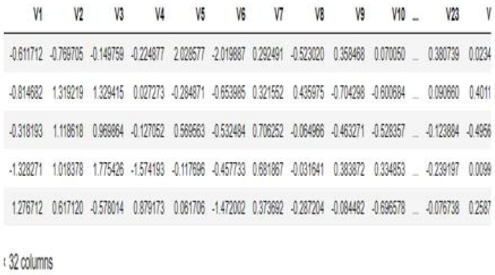

Algorithms Built
LGBM
XGBoost
CatBoost
Logistic Regression
Voting classifier - (LR, XGB, CatBoost), (LR, XGB), (XGB, Catboost), (LR, Catboost)
Extension – Stacking Classifier
RAW Data
Apply algorithms & 5-fold cross validation
Feature Extraction
Feature Selection
Data Preprocessing
Model Evaluation
Splitting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

Fig 1 Proposed architecture

iii) Dataset collection:

CREDIT CARD FRAUD DATASET: To train our machine learning algorithms, we used the Credit Card Fraud Detection dataset that we received from Kaggle. There were other transaction-related characteristics in the original dataset, such as "Amount," "Time," and "V1" through "V28." To provide successful fraud detection training without compromising sensitive information, we have omitted precise data about these original features for privacy and security reasons. The five most important rows in the dataset used to identify credit card fraud are therefore these. With that said, it has 32 columns, some of which are seen 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

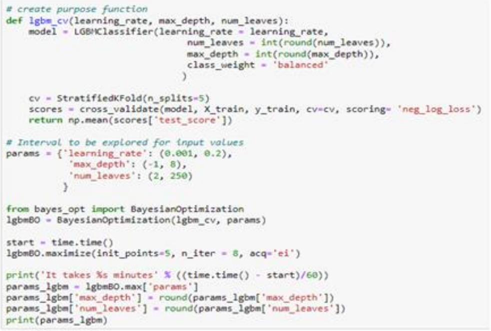
iv) Data Processing:

Processing data entails making sense of raw data for companies. Collecting, organizing, cleaning, validating, analyzing, and transforming data into understandable representations like graphs or papers are all part of data processing. There are three main ways that data may be processed: mechanically, electronically, or by hand. Improving the usefulness of data and making decisions easier are the goals. Companies may then use this information to make better strategic choices and enhance their operations. Software development and other forms of automated data processing are crucial here. Quality management and decision-making may benefit from its ability to transform massive data sets, particularly big data, into actionable insights.

v) Feature selection refers to the process of identifying which characteristics are most relevant, consistent, and free of duplication before building a model. With the proliferation of datasets comes the need to systematically reduce their sizes. The primary objective of feature selection is to decrease the computational cost of modeling while simultaneously improving the performance of a predictive model. An essential part of feature engineering is feature selection, which entails picking out the most relevant characteristics to feed into ML algorithms. By removing superfluous or unimportant characteristics and keeping just the most important ones, feature selection strategies help to decrease the amount of input variables used by machine learning models. Rather than relying on the machine learning model to prioritize features, it is recommended to undertake feature selection beforehand.

the sixth section, algorithms:

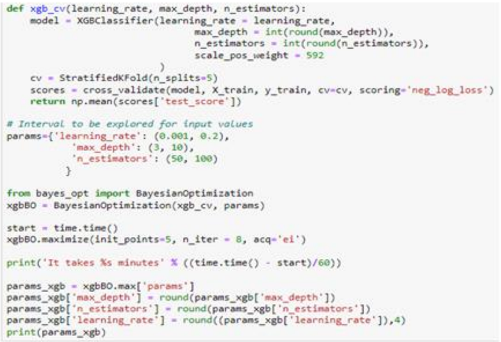
The Light Gradient Boosting Machine, or LGBM for short, is an efficient gradient boosting system that does great job with massive datasets. It's ideal for jobs like fraud detection because to its reputation for speed and accuracy. In order to optimize the boosting process and achieve quicker convergence, LGBM constructs an ensemble of decision trees [28].



```
# 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

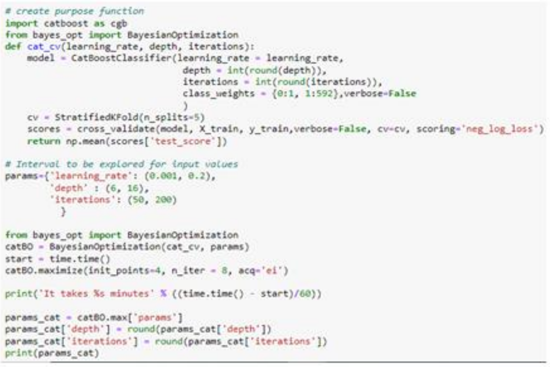
XGBoost (Extreme Gradient Boosting): One other gradient boosting technique with several applications in machine learning is XGBoost. Its performance and resilience have made it famous. Important for fraud detection, XGBoost's regularized gradient boosting architecture makes it adept at managing unbalanced datasets.



```
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

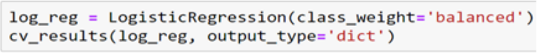
CatBoost (Categorical Boosting): To efficiently deal with categorical characteristics, the developers of the gradient boosting toolkit created CatBoost. This makes it simpler to deal with datasets that include categorical data by automating their treatment. It's practical for handling real-world financial data, resilient, and well-suited to avoiding overfitting [29, 30, 31, 32].



```
# 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

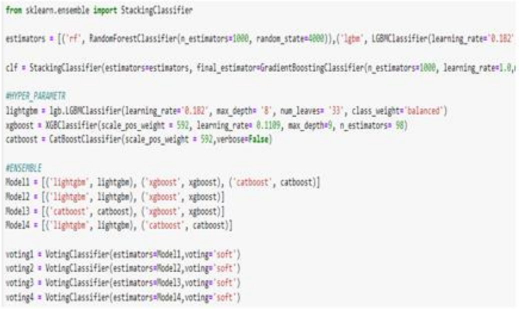
- **Logistic Regression:**



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

Fig 6 Logistic regression

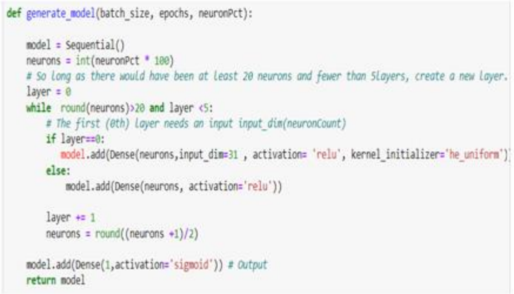
Voting Classifier: A number of machine learning models, including XGBoost, CatBoost, and Logistic Regression, contribute to the Voting Classifier's final forecast. In order to achieve better accuracy and resilience, this ensemble method uses the combined knowledge of several models. We have developed voting classifiers using various algorithm combinations [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: In deep learning, a model that mimics the way the brain works is called a Neural Network. When used in this way, it is able to detect intricate data patterns and correlations. One use of Neural Networks is their capacity to learn complex fraud patterns, particularly in big datasets.



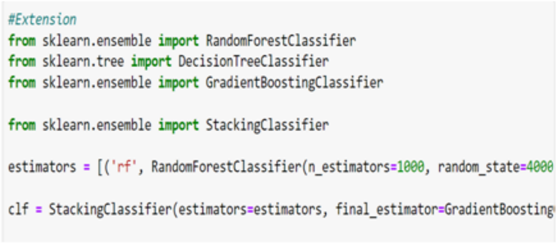
```
def generate_model(batch_size, epochs, neuronPct):

    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:

ISSN 2347–3657

Volume 12, Issue 2, 2024



```
#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

6. EXPERIMENTAL RESULTS

Precision: The accuracy rate, or precision, is the percentage of true positives relative to the total number of occurrences or samples. Consequently, the following is the formula for determining the accuracy:
Preciseness is TP divided by (TP plus FP), which is the sum of true positives and false positives.

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

Recall: The capacity of a model to detect all significant occurrences of a given class is measured by recall, a statistic in machine learning. The completeness of a model in capturing instances of a particular class is shown by the ratio of properly predicted positive observations to the total actual positives.

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

Accuracy:

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

F1 Score:

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

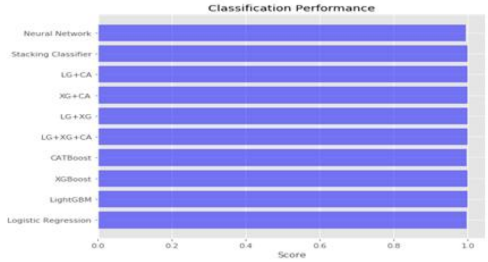
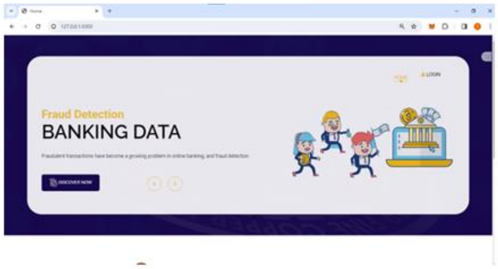
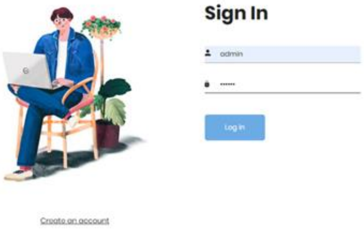


Fig 10 Performance Evaluation



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

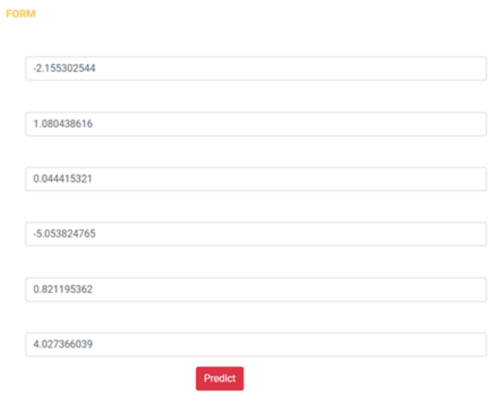
Fig 11 Home page



admin

•••••••

Fig 12 Signin page



-2.155302544

1.080438616

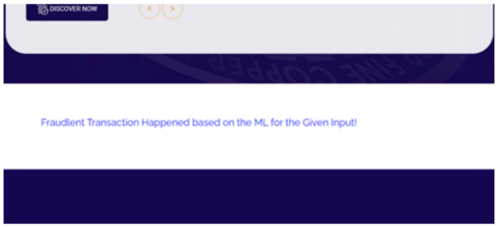
0.044415321

-5.053824765

0.821195362

4.027366039

Fig 13User input



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

Fig 14 Predict result for given input

7. CONCLUSION

By outperforming all other models in terms of accuracy, the Stacking Classifier proved to be the most effective at detecting fraud. The project's versatility was on display when it demonstrated strong performance across many machine learning models, such as neural networks, voting classifiers, and XGBoost, LightGBM, and CatBoost [29, 30, 31, 32]. It is worth noting that the use of varied sampling and scaling strategies greatly enhanced the accuracy of fraud detection. To highlight the efficacy of the ensemble technique, Stacking Classifier was used to considerably increase the accuracy of fraud detection. Making a Flask front-end that is easy to use simplifies authentication and user testing, making it more accessible and practical. Testing the system in Flask, where it received input, ensures that it works as intended and provides a good user experience. items[1,2, 3] Findings from this study pave the path for future uses of sophisticated machine learning methods to solve banking industry fraud detection problems. By delving into other ensemble approaches and optimization tactics, the project's results open up possibilities for further progress. The project's end goal is to improve banking sector security and confidence via increasing fraud detection skills, decreasing financial losses, and guaranteeing safe transactions.

8. FUTURE SCOPE

To further improve the accuracy and resilience of fraud detection, future research will look at integrating more hybrid models with CatBoost [29]. Optimizing the number of trees to enhance the model's efficiency will be the primary focus of future work to fine-tune CatBoost's hyperparameters [33]. To keep the model successful in detecting new fraudulent activity, researchers will concentrate on ways to adjust to shifting fraud trends. To better respond to new threats, researchers are working to make systems more sensitive and adaptable using real-time data. The next step is to improve the model's rationale for making decisions, so we can better understand how it builds trust and how to identify fraud.

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.