



ISSN: 2321-2152

**IJMECE**

*International Journal of modern  
electronics and communication engineering*

E-Mail

[editor.ijmece@gmail.com](mailto:editor.ijmece@gmail.com)

[editor@ijmece.com](mailto:editor@ijmece.com)

[www.ijmece.com](http://www.ijmece.com)

# MACHINE LEARNING TECHNIQUES FOR FINDING FRAUD IN BANK DATA

B AJITH KUMAR<sup>1</sup>, CHERAKALA LAVANYA<sup>2</sup>, A DHANASEKHAR REDDY<sup>3</sup>, TATIREDDY RAVI<sup>4</sup>

<sup>1</sup>Assistant Professor, Department of MCA, Sri Venkatesa Perumal College of Engineering & Technology, Puttur,  
Email: [ajithkumaryadav34@gmail.com](mailto:ajithkumaryadav34@gmail.com)

<sup>2</sup>P.G Scholar, Department of MCA, Sri Venkatesa Perumal College of Engineering & Technology, Puttur, Email:  
[yadavalavanya743@gmail.com](mailto:yadavalavanya743@gmail.com)

<sup>3</sup>Assistant Professor, Department of MCA, Sri Venkatesa Perumal College of Engineering & Technology, Puttur,  
Email: [ghanasekhar918@gmail.com](mailto:ghanasekhar918@gmail.com)

<sup>4</sup>Assistant Professor, Department of CSE, Sri Venkatesa Perumal College of Engineering & Technology, Puttur,  
Email: [tatireddyravi.ai@gmail.com](mailto:tatireddyravi.ai@gmail.com)

**Abstract:** The review centers around ML to detect financial data fraud. Financial institutions should detect and prevent fraud, making this a significant issue. The work presents class weight-tuning hyperparameters for fraud detection. These attributes empower the model recognize lawful and false exchanges, further developing fraud detection accuracy. The concentrate decisively utilizes CatBoost, LightGBM, and XGBoost. Every calculation has remarkable attributes, and their consolidated use further develops fraud detection. The work tweaks hyperparameters utilizing DL. This association further develops the fraud detection system's viability and adaptability, spotting arising fraud systems. The task assesses completely utilizing true information. These tests show that LightGBM and XGBoost outflank different methodologies in numerous classes. This recommends that the proposed technique detects fraud better than others. It utilizes a Stacking Classifier to join RandomForest and LightGBM predictions with determined boundaries. By utilizing differed models, an outfit approach involving a GradientBoostingClassifier as the last assessor further develops prediction accuracy.

**Index terms** - Bayesian optimization, data Mining, deep learning, ensemble learning, hyper parameter, unbalanced data, machine learning.

## 1. INTRODUCTION

The development of monetary organizations and online web based business have expanded monetary exchange volumes as of late. Online banking fraud is extending, and identification has forever been troublesome [1], [2]. Credit card fraud has advanced with charge card improvement. Credit card fraud is constantly refreshed, and fraudsters attempt to appear to be genuine. Fraudsters need to appear to be genuine. They concentrate on fraud detection frameworks and energize them, muddling fraud detection. Analysts are continuously searching for new methodologies or ways of upgrading existing ones [3].

Fraudsters exploit business application security, control, and observing blemishes. Innovation can check fraud [4]. Fraud should be recognized promptly to forestall more fraud [5]. Bogus or unlawful deception for cash or individual increase is fraud. False includes deceitfully utilizing credit cards for

physical or computerized buys. Fraud can happen in advanced exchanges since cardholders regularly supply the card number, lapse date, and check number through telephone or online [6].

Fraud prevention and detection can forestall fraud losses. Proactive fraud counteraction forestalls fraud. Be that as it may, fraud detection is fundamental when a trickster endeavors a deceitful exchange. [7]. Information is classed as legitimate or false in financial fraud detection [8]. Because of the volume and intricacy of monetary information, physically assessing and finding fake exchange designs is unfeasible or tedious. ML based strategies are significant to fraud detection and prediction [9].

ML techniques and high handling limit further develop fraud detection and huge dataset the executives. [15] DL and ML calculations address continuous issues rapidly [10]. This paper proposes a proficient credit card fraud detection technique in view of freely accessible datasets and advanced calculations LightGBM, XGBoost, CatBoost, and strategic relapse, as well as larger part voting joined techniques, DL, and hyperparameter settings. An ideal fraud detection system detects more false cases and ought high precision, i.e., all results to be accurately identified, which constructs client trust and forestalls misfortunes.[35]

## 2. LITERATURE SURVEY

The dynamic and broadened nature of web based business exchange fraud makes avoidance troublesome. [1] This study proposes fraud islands (connect investigation) and multi-layer ML model [10, 15, 20], which can recognize fluctuated fraud designs.

Fraud Islands are made using join examination to uncover confounded fraud examples and test deceitful substance communications. The different fraud designs are dealt with by multi-layer model. Today, banks' declination decisions, manual survey specialists' dismissal decisions, banks' fraud alarms, and clients' chargeback demands conclude fraud marks. Fraud risk avoidance powers (bank, manual survey group, fraud ML model) may get particular fraud designs. Coordinating a couple of ML models prepared with particular fraud marks upgraded fraud judgment precision [10].

False charging episodes are rising dramatically with government and confidential wellbeing upheld plans. [9] Because of perplexing connections between doctors, patients, and administrations, medical services fraud identification is troublesome. In this way, to increment strMLghtforwardness in wellbeing help programs, savvy fraud detection calculations should be created to distinguish holes in existing frameworks and false clinical charging circumstances. Moreover, specialist organization expenses and client health advantages should be improved. [2] This examination proposes a succession mining-based process-based fraud location approach for medical services protection guarantee fraud. Late examination underscores sum based investigation or medication versus MLlment consecutive investigation over grouping making of administrations inside every specialty to distinguish fraud. Our methodology makes regular arrangements with fluctuating example lengths. For each arrangement, certMLnty values and levels are determined. For every emergency clinic strength, the succession rule motor makes continuous groupings and certMLnty values and looks at them to patient qualities [2, 7, 9]. The two groupings abuse the

standard motor, demonstrating anomalies. Process-based fraud location is affirmed utilizing five years of nearby emergency clinic value-based information with many revealed fraud occurrences.

Because of monetary industry development, Mastercard volume has continually expanded. Fraud firms are developing essentially as well. Under these circumstances, fraud detection is turning out to be more helpful. The awkwardness dataset makes this challenge harder on the grounds that fraud is extensively lower than virtuoso exchanges. This work [3] portrays how to utilize helping calculations to recognize Visa fraud and momentarily thinks about them [29, 30].

Because of the ascent of internet business and online installment choices, Visa fraud is a worldwide concern. ML techniques for Visa fraud detection are acquiring prominence. Absence of openly accessible information, seriously lopsided class sizes, changed fraud conduct, and so on are issues. [5] Certifiable charge card exchange information is utilized to evaluate the fraud detection execution of Irregular Woods, Backing Vector Machine, and Strategic Relapse [20]. To adjust class sizes, we utilize Destroyed examining. In tests, steady learning of chosen ML calculations tends to consistently changing fraud patterns. Normal measurements like precision and review measure technique execution.

Mastercard fraud plagues monetary administrations. Consistently, charge card burglary costs billions. Due to mystery, true Mastercard information study is scant. This study identifies Visa fraud utilizing ML [10, 15, 20]. We start with standard models. AdaBoost-larger part casting a ballot half and half methodologies are

then utilized. Public Visa information is used to test the model. [6]A banking foundation's Mastercard information is then assessed. Information tests are additionally uproarious to test calculation flexibility. Positive testing discoveries show that greater part casting a ballot recognizes Visa robbery precisely.

Medical services fraud in the US is an expensive middle class wrongdoing with casualties. Fraud costs the general population in higher charges or significant recipient hurt [2, 7]. To address this social danger, computerized medical care fraud detection frameworks should adjust. US medical care is trying to digitize because of convoluted, heterogeneous information frameworks and different wellbeing models. Medical services fraud detection drives examiners to research and perhaps recover, recover, or allude to the appropriate specialists. Medical services fraud detection procedures and philosophies from the writing are summed up in [7]. This study space's friend checked on papers' significant points, discoveries, and information highlights are recorded. Potential holes in carrying out such frameworks to genuine medical services information will be featured. To cure these inadequacies, the creators propose different review subjects for future specialists.

### 3. METHODOLOGY

#### i) Proposed Work:

The task presents an ML based financial fraud detection system. Class weight-tuning and Bayesian enhancement with CatBoost, LightGBM, and XGBoost further develop execution. The calculation is refined utilizing DL and tried utilizing certifiable information and significant KPIs to identify and

forestall fraud. It includes a Stacking Classifier that joins RandomForest and LightGBM expectations with indicated boundaries [17, 28]. By utilizing fluctuated models, an ensemble approach involving a GradientBoostingClassifier as the last assessor further develops prediction accuracy. An easy to use Flask structure associated with SQLite gives information exchange and signin capabilities to client testing and refining the framework's convenience and common sense in genuine fraud detection applications.[37]

## ii) System Architecture:

The framework begins with credit card transaction data, including fraud and realness marks. ML requires information planning, including feature extraction and selection. The dataset has two subsets: prepared for model structure and tried for execution assessment. Bayesian improvement upgrades ML hyperparameters. To ensure model strength, 5-crease cross-validation is utilized to apply ML calculations like CatBoost, LightGBM, and XGBoost to preparing information. As a venture extension, we considered stacking classifier. The calculations' credit card fraud detection and false positive decrease are assessed utilizing a few rules.

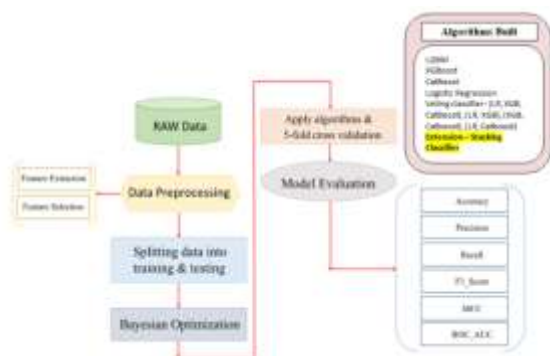


Fig 1 Proposed architecture

## iii) Dataset collection:

**CREDIT CARD FRAUD DATASET:** We prepared ML calculations on Kaggle's Credit Card Fraud Detection dataset. At first, the data contained exchange qualities including "Sum," "Time," and "V1" through "V28." Explicit particulars in regards to these underlying highlights were excluded to safeguard delicate data while permitting fraud detection training. Here are the main 5 credit card fraud detection lines. It has 32 sections, some of which are seen here [6, 17].

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V
-0.611712	-0.7698705	-0.149759	-0.224877	2.028577	-2.018807	0.282494	-0.523020	0.558486	0.070050	...	0.380739	0.0234
-0.014502	1.313218	1.329415	0.027273	-0.294571	-0.653595	0.321552	0.435975	-0.704298	-0.050804	...	0.043650	0.4011
-0.310193	1.110618	0.860864	-0.127052	0.585563	-0.532484	0.705252	-0.044985	-0.463271	-0.528257	...	-0.123804	-0.4858
-1.328271	1.018378	1.775428	-1.574193	-0.117595	-0.457733	0.581867	-0.031541	0.383872	0.344853	...	-0.239197	0.0038
1.276712	0.617128	-0.578814	0.875173	0.081705	-1.472002	0.373862	-0.287204	-0.044482	-0.898578	...	-0.076758	0.2587

32 columns

Fig 2 NSL KDD dataset

## iv) Data Processing:

Data processing transforms crude information into business-helpful data. Information researchers accumulate, put together, clean, check, dissect, and orchestrate information into charts or papers. Information can be handled physically, precisely, or electronically. Data ought to be more significant and decision-production simpler. Organizations might upgrade tasks and settle on basic decisions quicker. PC programming improvement and other robotized data processing innovations add to this. Enormous information can be transformed into significant bits of knowledge for quality administration and independent direction.



#### v) Feature selection:

Feature selection chooses the most steady, non-repetitive, and pertinent elements for model turn of events. As data sets extend in amount and assortment, purposefully bringing down their size is significant. The fundamental reason for feature selection is to increment prescient model execution and limit processing cost.[39]

One of the vital pieces of feature engineering is picking the main attributes for machine learning algorithms. To diminish input factors, feature selection methodologies take out copy or superfluous elements and limit the assortment to those generally critical to the ML model. Rather than permitting the ML model pick the main qualities, feature selection ahead of time enjoys a few benefits.

#### vi) Algorithms:

- **LGBM (Light Gradient Boosting Machine):**

LGBM is a successful gradient boosting system for gigantic datasets. Its speed and precision make it ideal for fraud detection. LGBM upgrades boosting for speedier intermingling through 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': (1, 256)}

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 one more gradient boosting technique that is generally used for different ML issues. Strength and execution are its trademarks. XGBoost handles slanted datasets well utilizing regularized gradient boosting, which is fundamental for fraud detection.

```
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 = 102)

    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, 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):**

The gradient boosting library CatBoost is upgraded for class features. Computerizing classification information makes it more straightforward to manage. It's dependable, handles overfitting great, and functions admirably with monetary information [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, 1.55], 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
pbo = BayesianOptimization(
    cat_cv,
    {'learning_rate': (0.001, 0.1),
     'depth': (5, 16),
     'iterations': (50, 500)}
)

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

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

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

Fig 5 Catboost

- **Logistic Regression:**

An essential binary classification approach is calculated regression. It is a fraud detection benchmark model yet less complex than ensemble approaches like boosting. It's not difficult to get a handle on and uncovers feature significance.

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

Fig 6 Logistic regression

- **Voting Classifier:**

The Voting Classifier utilizes Logistic Regression, XGBoost, and CatBoost to anticipate. This ensemble strategy utilizes many models' insight to expand accuracy and strength. Different calculation mixes have been utilized to assemble voting classifiers [19, 24].

```
from sklearn.ensemble import StackingClassifier
estimators = [
    ('rf', RandomForestClassifier(n_estimators=100, random_state=0)),
    ('lgb', LGBClassifier(learning_rate=0.1))
]
clf = StackingClassifier(estimators=estimators, final_estimator=RandomForestClassifier(n_estimators=100, learning_rate=0.1))

# define models
lgbmodel = LGBClassifier(learning_rate=0.1, max_depth=7, min_samples=10, class_weight='balanced')
xgbmodel = XGBClassifier(scale_pos_weight=1.5, learning_rate=0.1, max_depth=6, n_estimators=50)
catmodel = CatBoostClassifier(scale_pos_weight=1.5, verbose=False)

# define models
Model1 = [lgbmodel, lgbmodel, lgbmodel, lgbmodel, catmodel]
Model2 = [lgbmodel, lgbmodel, lgbmodel, lgbmodel]
Model3 = [catmodel, catmodel, lgbmodel, lgbmodel]
Model4 = [lgbmodel, lgbmodel, lgbmodel, catmodel]

# define models
model1 = VotingClassifier(estimators=Model1, voting='hard')
model2 = VotingClassifier(estimators=Model2, voting='hard')
model3 = VotingClassifier(estimators=Model3, voting='hard')
model4 = VotingClassifier(estimators=Model4, voting='hard')
```

Fig 7 Voting classifier

- **Neural Network:**

Neural Networks are cerebrum enlivened DL models. This permits it to catch muddled information examples and linkages. NN can learn complex fraud designs in tremendous datasets.[44]

```
def generate_model(batch_size, epochs, neuronsPer):
    model = Sequential()
    neurons = int(neuronsPer * 100)
    # So long as there would have been at least 20 neurons and fewer than 5 layers, create a new layer.
    layer = 0
    while round(neurons)/20 and layer < 5:
        # the first (20) layer needs an input_dim(neuronsCount)
        if layer==0:
            model.add(Dense(neurons, input_dim=11, activation='relu', kernel_initializer='he_uniform'))
        else:
            model.add(Dense(neurons, activation='relu'))

        layer += 1
        neurons = round((neurons * 0.75))

    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 ensemble strategy Stacking Classifier joins Random Forest and LightGBM forecasts with indicated boundaries. The last assessor, a GradientBoostingClassifier, mixes the capacities of various models in ensemble learning out how to further develop prediction accuracy.[40]

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

from sklearn.ensemble import StackingClassifier

estimators = [('r4', RandomForestClassifier(n_estimators=1000, random_state=4000))
clf = StackingClassifier(estimators=estimators, final_estimator=GradientBoostingClassifier(n_estimators=1000, random_state=4000))
```

Fig 9 Stacking classifier

#### 4. EXPERIMENTAL RESULTS

**Precision:** Precision estimates the level of positive cases or tests precisely sorted. Precision is determined utilizing the recipe:

$$\text{Precision} = \frac{\text{True positives}}{(\text{True positives} + \text{False positives})} = \frac{TP}{(TP + FP)}$$

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

**Recall:** Machine learning recall assesses a model's ability to perceive all significant examples of a class. It shows a model's culmination in catching occasions of a class by contrasting accurately anticipated positive perceptions with complete positives.

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

**Accuracy:** A test's accuracy is its ability to recognize debilitated from sound cases. To quantify test accuracy, figure the small part of true positive and true negative in completely broke down cases. Numerically, this is:

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

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

**F1 Score:** Machine learning model accuracy is estimated by F1 score. Consolidating model precision and recall scores. The accuracy measurement estimates how frequently a model anticipated accurately all through the dataset.

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

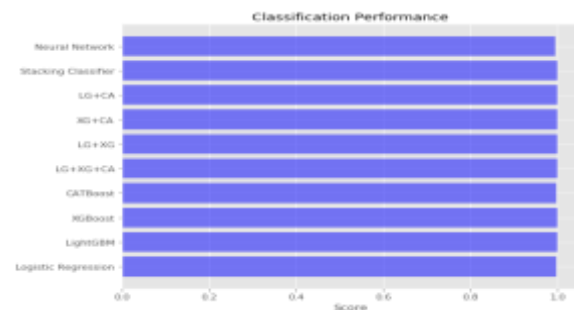


Fig 10 Performance Evaluation



Fig 11 Home page





Fig 12 Signin page

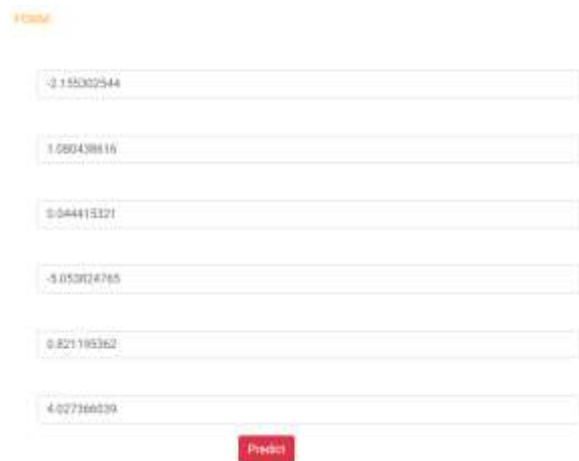


Fig 13 User input



Fig 14 Predict result for given input

## 5. CONCLUSION

Stacking Classifier has the best accuracy among all models, demonstrating its fraud detection strength.

The venture performed well with LightGBM, XGBoost, CatBoost [29, 30, 31, 32], voting classifiers, and neural networks, showing its flexibility. Various testing and scaling improved fraud detection accuracy, highlighting their significance. Stacking Classifier further developed fraud detection accuracy, demonstrating its adequacy. A straightforward Flask front-end works on client testing and validation, making it open and functional. Criticism from Flask testing confirms the framework's usefulness and client experience. [1, 2, 3] The analysis shows that strong ML can be utilized to detect financial fraud, opening the entryway for future applications. The task's outcomes consider ceaseless improvement by researching gathering and streamlining strategies. The drive further develops fraud detection, financial losses, and transaction security, boosting banking sector security and certainty.[42]

## 6. FUTURE SCOPE

Next study will consolidate more hybrid models with CatBoost [29] to further develop fraud detection accuracy and robustness. Future work will enhance CatBoost's hyperparameters, zeroing in on tree build up to further develop model productivity [33]. Examination will zero in on adjusting to moving extortion propensities to keep the model identifying new misrepresentation. Constant information is being investigated to increment framework responsiveness and variation to arising dangers. Further work will explain the model's dynamic cycle to further develop fraud detection and trust building.

## 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. Brkić, "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. (TENSYP)*, 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.
- [35] G. Viswanath, "Hybrid encryption framework for securing big data storage in multi-cloud environment," *Evolutionary intelligence*, vol. 14, 2021, pp. 691–698.
- [36] Viswanath Gudditi, "Adaptive Light Weight Encryption Algorithm for Securing Multi-Cloud Storage," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, 2021, pp. 545–552.
- [37] Viswanath Gudditi, "A Smart Recommendation System for Medicine using Intelligent NLP Techniques," 2022 *International Conference on Automation, Computing and Renewable Systems (ICACRS)*, 2022, pp. 1081–1084.
- [38] G. Viswanath, "Enhancing power unbiased cooperative media access control protocol in manets," *International Journal of Engineering Inventions*, 2014, vol. 4, pp. 8–12.
- [39] Viswanath G, "A Hybrid Particle Swarm Optimization and C4.5 for Network Intrusion Detection and Prevention System", 2024, *International Journal of Computing*, DOI: <https://doi.org/10.47839/ijc.23.1.3442>, vol. 23, 2024, pp. 109–115.
- [40] G. Viswanath, "A Real Time online Food Ordering application based DJANGO Restfull Framework", *Juni Khyat*, vol. 13, 2023, pp. 154–162.
- [41] Gudditi Viswanath, "Distributed Utility-Based Energy Efficient Cooperative Medium Access Control in MANETS", 2014, *International Journal of Engineering Inventions*, vol. 4, pp. 08–12.
- [42] G. Viswanath, "A Real-Time Video Based Vehicle Classification, Detection And Counting

System”, 2023, Industrial Engineering Journal, vol.52, pp.474-480.

[43] G.Viswanath, “A Real- Time Case Scenario Based On Url Phishing Detection Through Login Urls ”, 2023, Material Science Technology, vol.22, pp.103-108.

[44] Manmohan Singh,Susheel Kumar Tiwari, G. Swapna, Kirti Verma, Vikas Prasad, Vinod Patidar, Dharmendra Sharma and Hemant Mewada, “A Drug-Target Interaction Prediction Based on Supervised Probabilistic Classification” published in Journal of Computer Science, Available at:  
<https://pdfs.semanticscholar.org/69ac/f07f2e756b79181e4f1e75f9e0f275a56b8e.pdf>