

A Deep Learning Ensemble With Data Resampling For Credit Card Fraud Detection

Dheeraj Chalasani¹, Dr. Y.Rama Krishna²

^{1,2}CSE, KL University, Vijayawada

^{1,2}2401050188cse@gmail.com, yramakrishna@kluniversity.in

Abstract

Credit cards are extremely important in today's digital economy. Its use has been significantly increased in recent years due to credit card theft. Credit card fraud detection has ML classifiers to do their best, using the fact that credit card holders are constantly changing and classroom imbalances are problematic. To solve this problem, this study is a robust methodology of DL with "long short-term memory (LSTM) and Repeat Units (GRU)" as a fundamental student of emergency architecture, and is "multilayer" than meta ears. The technique of hybrid synthetic minority and the modified method of the closest neighbor (SMOTE-EN) is used to increase the distribution of the class in the data file more and at the same time. Experimental findings showed that integration of the proposed DL set with Small-en-Accessible showed that the integration of "sensitivity of 1,000 and specificity of 0.997 recorded other frequently used classifiers and ML methods in the literature. It also covers more complex file models such as stacking and voting classifiers. Test them with both the original and the Small-en data". The SQLite Integration Framework also allows users to register, register, and test. This will help your project work better and interact with users.

Keywords: Credit card, deep learning, ensemble learning, fraud detection, machine learning, neural network.

1. Introduction

The financial sector was heavily influenced by the progress of information technology, leading to the expansion of electronic trading platforms (electronic trade). "The recent emergence of the Roman pandemic coronavirus (Covid-19)" has been greatly driven by the electronic trading sector, underscoring the need for a digital environment [1] [2]. "One of the biggest issues with modern electronic trade is the large number of examples of credit card fraud [3]. Credit card fraud has also been on the rise in the past decade, and is a major problem for banks and other financial institutions" [4]. The rise in CCF is linked to the growth of electronic trade and other "online transactions. Therefore, credit card fraud detection (CCFD) is extremely important for financial companies to avoid losing money".

Companies in the financial sector can achieve high benefits by using "Artificial Intelligence (AI)" and ML to increase productivity, reduce costs, and happier customers [5]. There are many ML-based techniques to find credit card scams. For example, Malik et al. [6] "they examined the use of hybrid models in CCFD. The hybrid model was manufactured by assembling a variety of MLs. Experimental results show that the integration of the hybrid model, Adaboost and LGBM", achieved excellent classification performance. Alfaiz and Fati [7] evaluated the performance and strategies of ML classifiers for identifying data fraud with credit cards in a similar study. This study used classifiers such as "LGBM, XGBoost, Random Forest, Category Enhancement (CATBOOST), Logistics Regression, and Naive Bayes. The results showed that the Catboost classifier" worked better than the other methods, combined with the underlined k-non-closer district.

At the same time, it was difficult to create strong CCFD models based on ML for many reasons. First, traditional classifiers look only at the details of the transaction, such as the amount, the country and the type of transaction. They do not look at the order of transactions, which is useful for finding fraud formulas [8], [9]. Second, credit card statistics are quite unbalanced because there are much more real transactions than fake [10]. Classification of imbalance is the problem of predictive modeling characterized by uneven distribution of samples between classes [11]. Most of the class is a class that makes up most of the data file and the minority class is one that forms a smaller part of the data file. It is difficult to classify imbalance, because most ML techniques have been built with the idea

that classes will be evenly distributed. As a result, the use of unbalanced data sets, such as credit card data set, provides “models with suboptimal classification, especially for minority class, namely fraudulent transactions”. In addition, it is also important to correctly identify samples from the minority class in situations with unbalanced classification [12].

“Deep Learning (DL)” And ML recently dominated ML [13] [1], [15] [16]. This can be used to provide a good forecast of the outcome in complex situations and overcome obstacles for CCF detection. DL is a type of ML, usually composed of neural networks with many layers [17]. Various tasks in ML based on sequential modeling DL models used “recurrent neuronal networks (RNNs) [18] [19], [20]. Shen et al. [21]” They observed that sequential modeling methods such as RNNs usually have superior performance compared to traditional ML models. At the same time, simplification of RNN-based models is responsive to the issue of disappearance gradients. This occurs when the RNN does not send important information about the model output to the layer near the end of the start [22]. “RNNs based on LSTM and GRU were developed as solutions to the problem of disappearing gradients and showed praise performance in different sequence classifications [8] [23]”.

1.1 Literature survey

Numerous research efforts in “Deep Neural Networks (DNNs)” have recognized credit card fraud, which focuses on improving the accuracy of points and alleviating unnecessary distortions through the development of various network designs or ML [1]. Quantifying uncertainty is essential in point estimation. This is because it allows for reliable systems to reduce model strain and avoid suboptimal decisions that result from inadequate trust. It is important to assess the uncertainty associated with DNN predictions associated with actual CCF recognition scenarios for northern reasons. (a) The fraudster will continuously change its strategy. This work represents three techniques for quantifying uncertainty (UQ) - Monte Carlo Spanche, Ensemble, Ensemble Monte Carlo Spoce - Identify cards containing transaction data. Additionally, we evaluate estimates of prediction uncertainty using a UQ interference matrix and several performance indicators. Experimental results show that the ensemble better captures the uncertainty associated with the generated predictions. It also shows that the proposed UQ approach provides more information on point forecasting and supports the fraud prevention process.

With new technologies and communication options, such as contactless payments, becoming more and more common, credit card fraud is becoming an increasingly problematic. “This article [2] provides a comprehensive analysis of innovative research findings focused on detection and prediction of fraudulent credit card transactions from 2015 to 2021”. Reviews of 0 related publications are organized by topics covered (such as class or elemental issues). Our study shows that there was no insufficient investigation by DL, which underscores the need for further research findings. Our work has led to current research questions and is manifested on future research trips. This is a great source for academic and industrial scientists who evaluate financial fraud marking systems and find powerful solutions.

Fraud has become one of the main risks for electronic trading businesses as the electronic trade has grown. [3] fraudulent actions seriously undermine the evaluation systems of electronic trading platforms and negatively affect the experience of buying consumer. It is quite useful to find fraud on electronic trading websites. However, this work is not easy, because fraudsters do things that are against the law. Current fraud detection systems used in the electronic trading industry are susceptible to degradation and lack of adaptability to developing fraud formula because they rely on established fraudulent behavior as supervisory data to identify suspicious activities. This article is a competitive graphic system of “detection of neural network -based fraud (CGNN) (EFRAUDCOM)” designed to identify fraudulent activities on one of the largest platforms of electronic trading “Taobao”. In EFRAUDcom, (1) of the “competing graphical neural network (CGNN)” serves as the basic component, allowing direct user behavior of users “by modeling the distribution of normal and fraudulent behavior; (2)” Some normal behavior is used as weak information about supervision that helps CGNN to create a profile for normal behavior that shows greater stability than fraudulent behavior [31,32]. Relying on fraudulent behavior will be eradicated, allowing Efraudcom to identify fraudulent activities in the presence of new fraud formulas; (3) The term regularization of mutual information may improve CGNN by maximizing the distinction between normal and fraudulent behavior. “Efraudcom is integrated into the prototype system and the system's performance is evaluated by complex experiments”. Tests on two files of public data sets and two Taobao data sets show that the proposed deep CGNN frame is better to look for fraud

than other baseline lines. A case study using Taobao data sets confirms that CGNN is durable despite improving fraud behavior.

When creating reliable detection systems of “credit card fraud (CCF)”, the problem of unbalanced data sets is a big worry. This article examines and evaluates the recent development of ML algorithms and “deep reinforcement learning (DRL) applied to credit card detection systems (CCF)”, which include both fraudulent and non -fragile categories. Smote and Adasyn are two ways to rework the CCF data file that is not balanced. [4] Then the ML methods are used on this balanced data file to create systems that can find CCF. Then DRL is used to develop detection algorithms that work with an unbalanced CCF data file. Different categorization measures are used to complete how well these ML and DRL models work. We have found the reliability of ML models through empirical studies using two methods of resampling and DRL models to identify CCF. When the original CCF data sets are carried out using Smote and Adasyn before “the training/test division, the ML models get a very high level of accuracy over 99%. However, these ML [4]” models show worse results when only these techniques are used to rework only CCF training sets. For example, logistics regression using “Adasyn has an accuracy of 1.81% and the F1 score 3.55%. Our research shows that the DRL model does not work well and receives only 34.8% of the time”.

Over the years, the damage caused by financial crimes causes to banks and other financial organizations much worse. Many methods of personal and hybrid ML have been used to find crimes such as CCF. However, these methods have a major problem as they are not examining other hybrid gorms in the same data file. This study [6] presents and examines seven models of hybrid-ML to identify malware using real data files. The hybrid model was installed in steps. First, the best single algorithm from the first phase was used to help you find fraud on your credit card. The hybrid methods were then created using the best single algorithm from the first phase. “Our results show that the hybrid model Adaboost + LGBM is optimal, and therefore is optimal. future research should focus on studying hybridization and algorithms of various forms of the credit card sector”.

2. Methodology

2.1 Proposed Work:

The proposed technique uses DL files to create a strong way to find credit card fraud. As students of basic neural networks, the stacking set of “long short-term memory (LSTM) and the gates of recurrent units (GRU) of neural networks and as a meta-learner multilayer perception (MLP)”. This method works well to solve problems when changing shopping habits and class imbalances when detecting fraud in credit card fraud. The system uses “a hybrid synthetic minority technique and modified the closest neighbor (Smote-En)” method to solve the problem of the class imbalance. Experimental results show that it is more sensitive and specific than traditional ML approaches, making it a great alternative to detecting fraud in real time. “The proposed system is compared with the models Adaboost, Random Forest, MLP, LSTM and GRU [8], [23] and [24]”. Next, we add more advanced file methods, including stack classifiers consisting of random forests and MLP, and more advanced file methods, including adaboost and random audio classifiers consisting of random. These models have been tested on both the original dataset and the Small Een improved dataset. Flasks with SQLite integration have also been created to make it easier for users to register, register and select features. This supplement is enhanced by the project by thoroughly checking multiple classifiers and adding a user-friendly interface to get simple interactions and tests.

2.2 System Architecture:

The system begins with the collection of credit card transactions that contain information about both normal and possible false purchases. The data obtained undergo preliminary processing that includes data cleaning, missing numbers and data change to ensure that it is in good quality. Data sampling strategies are used to correct class imbalances. This means an excessive sample (fraud transactions) of minority classes (fraud transactions) with techniques such as Small-En [28] [29]. Select functions using the approach to find the most useful features or features for finding scams. This limits the number of “dimensions and focuses on the data properties that are most important for classification. The selected function is used as input for ML and DL. These classifiers learn to characterize the difference between legitimate and fraudulent transactions” by considering preliminary and tasting data. The system is equipped with a check phase to see how the classifier works well. This usually means using a separate data record for reviews of how well a model can be generalized. “We use metrics such as accuracy,

accuracy, download, score F1, ROC curve, AUC, sensitivity, and more to see how well the classifier works". This test is run on real and fake transactions to see how well the system is working. The system uses ratings to determine whether a new credit card transaction is healthy or fraudulent.

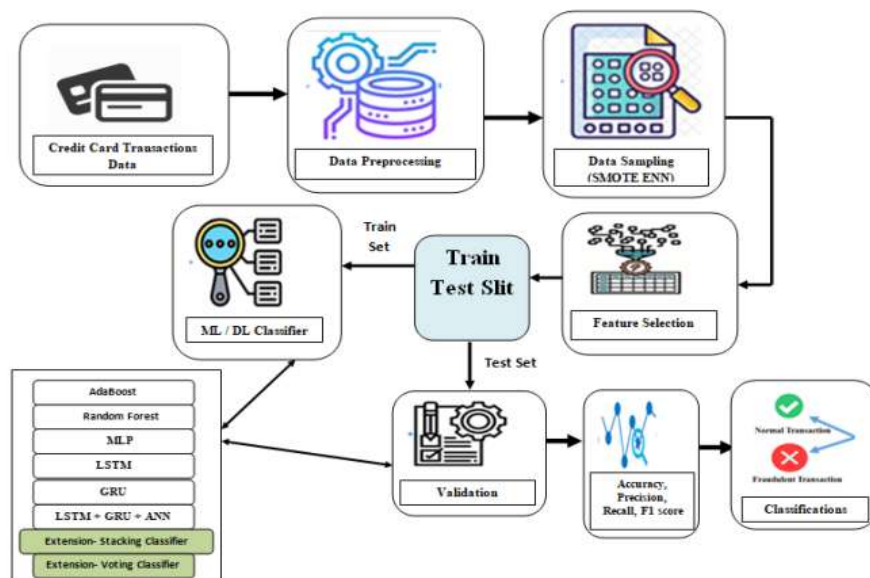


Fig 1: System Architecture"

2.3 Dataset collection:

Research uses Kaggle data file and uses access to data augmentation to solve the problem using cards with cards, along with exploratory data analysis and correlation analysis to increase information understanding. These methods help show how the data spreads, where there are remote values and how the variables are connected to each other, which makes it easier to interpret data and later develop models. We trained ML algorithms using Kaggle Credit Card Detection Data Fresh [17]. The "amount", "time" and "V1" on "V28" were some of the attributes related to the transaction that were initially in the data file. For the protection of private information, details concerning the original characteristics have been kept secret.

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	...	V23	V
-0.611712	-0.766705	-0.149759	-0.224877	2.028577	-2.019887	0.292491	-0.523020	0.358458	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	0.969064	-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	-0.064482	-0.696578	...	-0.076738	0.2587

32 columns

Fig 2: Dataset

2.4 Data Processing:

Data processing is the process of converting RAW data into information useful to the enterprise. Data scientists usually work on formats, such as graphics or text that allow people to read data collection, organization, cleaning, control, analysis, and information changes. There are three types of data processing: manual machines and computers. The aim is to increase more useful information and help people make decisions. This helps organizations to run better and make intelligent decisions at the right moment. The development of computer software and other automated data processing technologies are a big part. It can help you understand many data, even huge data, so you can make better decisions and control quality.

2.5 Feature selection:

The selection of features is the process of finding the most important, consistent and unique features that you can use to create a model. Since the data sets are increasing and more diverse, it is necessary to reduce their size methodically. The basic purpose of choosing functions is to make it difficult to make the predictive model better and the cost of running is less.

Functional selection is an important part of engineering. This is the process of selecting the most important features used in ML algorithms. To reduce the amount of input variables, the approach to selecting functions will alleviate properties that are useless or too similar to others. They also reduce a set of functions to those that are most useful for ML. The key advantages of selecting functions in advance instead of making the ML model to find out which functions are most important are:

2.6 Algorithms:

“Adaboost”, Adaptive Stringning is an ML technology that makes it more accurate to put together some simple models. Tree Decision - Start with a simple model, such as a manufacturing tree, and always train a new model. This increases weight due to data points where the previous model did not work. Adaboost is a more powerful file for these models. This helps the project improve the detection of credit card learning from previous models and improve overall performance [36].

```
from sklearn.ensemble import AdaBoostClassifier

# instantiate the model
ada = AdaBoostClassifier(n_estimators=100, random_state=0)

ada.fit(X_train, y_train)

y_pred = ada.predict(X_test)
```

Fig 3: Adaboost

“Random Forest” ML ways to use more than a tree decision - where. Train a group of decisions to create a tree for random data and combine estimates to create a single forecast. This file method improves accuracy, limits excessive amounts, and works well in both classification and regression problems.

```

from sklearn.ensemble import RandomForestClassifier

# instantiate the model
forest = RandomForestClassifier(max_depth=2, random_state=0)

forest.fit(X_train, y_train)

y_pred = forest.predict(X_test)

```

Fig 4: Random forest

This research uses a certain “artificial neural network (ANN) called a multilayer perceptron (MLP) to find credit card fraud. It has many layers of neurons that are mutually connected and cooperated” on the processing of inputs and learning complex patterns. MLP changes its internal settings during training to make sure its predictions are as accurate as possible. MLP is a good tool for finding fake credit card transactions because it can adapt and find non-linear data relationships.

```

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20,
                                                    random_state=0)
print('X_train size: {}, X_test size: {}'.format(X_train.shape, X_test.shape))

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
results = []

```

Fig 5: MLP

LSTMS aims to solve problems that have common RNN when working with sequential data. You can acquire and sustain a diverse array of sequences that are effective in professional settings, “including speech recognition, natural language processing, and temporal chain analysis. [4]” LSTM employs many cells, gates, and conditions to retain and transmit temporal information. This enables the imitation of intricate addiction and the manifestations of sequential data.

```

inputs1=Input((1,11))
att_in=LSTM(50,return_sequences=True,dropout=0.3,recurrent_dropout=0.2)(inputs1)
att_in_1=LSTM(50,return_sequences=True,dropout=0.3,recurrent_dropout=0.2)(att_in)
att_out=attention()(att_in_1)
outputs1=Dense(1,activation='sigmoid',trainable=True)(att_out)
model1=Model(inputs1,outputs1)

```

Fig 6: LSTN

The stacking classifier is a way to use ML to create a model that is a stronger and more accurate combination of predictive capabilities of several basic classifiers. “The stacking classifier system uses two basic classifiers: random forest and multilayer perceptron (MLP). The classifier for strengthening a Light Gradient Boosting Machine (LGBM)” creates a final prediction. The stacking classifier seeks to increase the overall prediction performance by using different capabilities of these classifiers. This file method could be useful for solving difficult classification tasks and complicated data files by combining the experience of various basic classifiers.

```
estimators = [('rf', RandomForestClassifier(n_estimators=10)),('mlp', MLPClassifier(random_state=1, max_iter=500))]
clf1 = StackingClassifier(estimators=estimators, final_estimator=LGBMClassifier(n_estimators=10))

clf1.fit(X_train,y_train)

y_pred = clf1.predict(X_test)
```

Fig 7: Stacking classifier

“**Gated Recirrent Unit (GRU)** is a type of recurring neural network (RNN)”, which is very good to handle sequential data. It is comparable to the “Long-term Short-Term memory (LSTM)” but is calculated more efficiently. [8] The strong fact is that GRUs can find dependencies and expressions in order of low computing power. This occurs in less critical gateway systems that restrict information flow, maintain critical data, and are less important. GRU is a popular option for applications that require “sequential data such as natural language processing, time series analysis,” and healthy recognition. ML applications work simple and well, so many people choose it in ML applications.

```
inputs1=Input((1,11))
att_in=GRU(50,return_sequences=True,dropout=0.3,recurrent_dropout=0.2)(inputs1)
att_in_1=GRU(50,return_sequences=True,dropout=0.3,recurrent_dropout=0.2)(att_in)
att_out=attention()(att_in_1)
outputs1=Dense(1,activation='sigmoid',trainable=True)(att_out)
model1=Model(inputs1,outputs1)
```

Fig 8: GRU

In this research, a strong model of the file is created by assembling a “Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and an Artificial Neural Network (ANN) Multilayer Perceptron(MLP). LSTM and GRU are two forms of recurring neural networks (RNN)”, which It's very good to recognize a sequence and how it depends on each other. LSTMs are excellent when connected over long distances, and GRU is superior when calculating [8] [23] [2]. By adding MLP as a metaleaf, the class files improve complex formulas for credit card transaction data. This combination is known to be able to find both short-term and long-term addictions, and is much more accurate and useful for project recognition.

```

inputs1=Input((1,11))
att_in=LSTM(50,return_sequences=True,dropout=0.3,recurrent_dropout=0.2)(inputs1)
att_in_1=GRU(50,return_sequences=True,dropout=0.3,recurrent_dropout=0.2)(att_in)
att_out=attention()(att_in_1)
outputs = Dense(8, activation='relu')(att_out)
outputs = Dense(6, activation='relu')(att_out)
outputs1=Dense(1,activation='sigmoid',trainable=True)(att_out)
model1=Model(inputs1,outputs1)

```

Fig 9: LSTM + GRU + ANN

The algorithm of the soft voting classifier is one of the many algorithms that make up file learning in ML. This method uses predictions of several separate classifiers to come up with the final prediction. Instead of feeding the same weight instead of feeding the same weight, we use the same weight and use estimates of the probability that “each classifier of different classes shows. This algorithm combines these probability estimates” to give predictions to a safe classifier. This leads to more accurate and refined final predictions. To search for credit card fraud, the system works better with the best features of each model.

```

ecclf2 = VotingClassifier(estimators=[('ad', clf1), ('rf', clf2)], voting='soft')
ecclf2.fit(X_train, y_train)
y_pred = ecclf2.predict(X_test)

```

Fig 10: Voting classifier

3. Experimental results

Precision: Accuracy focuses on samples of all individuals who are characterized as a percentage of correctly classified cases or positive. Therefore, the equation for determining the accuracy is:

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

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

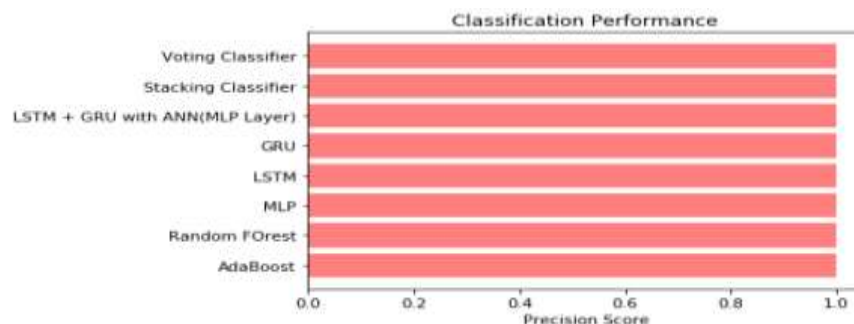


Fig 11: Precision comparison graph

Recall: In ML, an appeal is a measure that measures how well a model can find all relevant examples of a particular class. This is the proportion of approximate positive comments correct for specific real positivity, and shows how well the model contains all examples of a particular class.

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

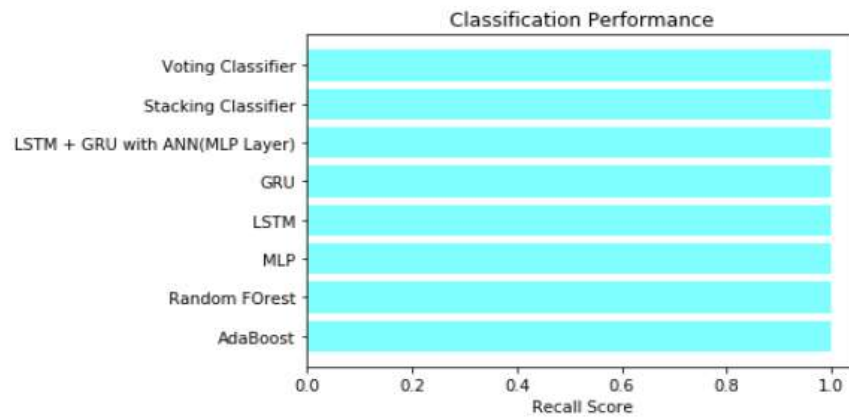


Fig 12: Recall comparison graph

Accuracy: In the classification task, the accuracy is the percentage of the right predictions. It shows how correct the model predictions are overall.

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

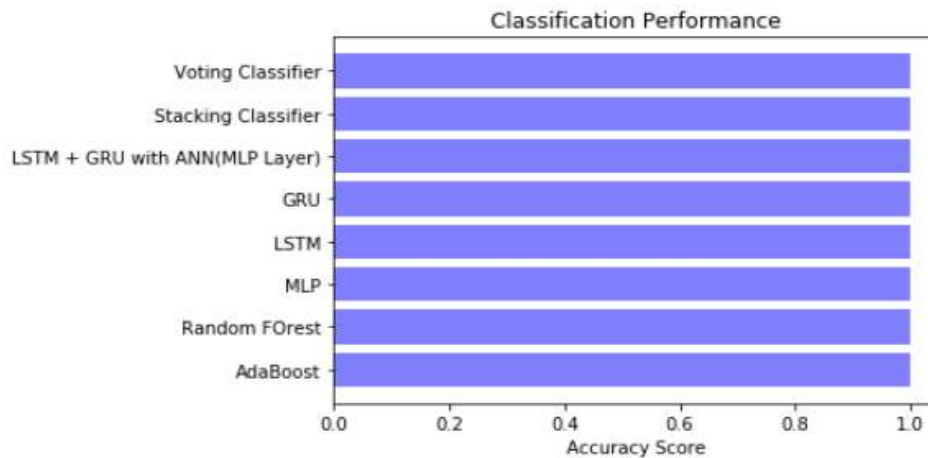


Fig 13: Accuracy graph

F1 Score: F1 points are a harmonious average of accuracy and memory. “This is a balanced metric that takes into account both false positives and false negatives”. Therefore, it works well with data records that are not evenly distributed.

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

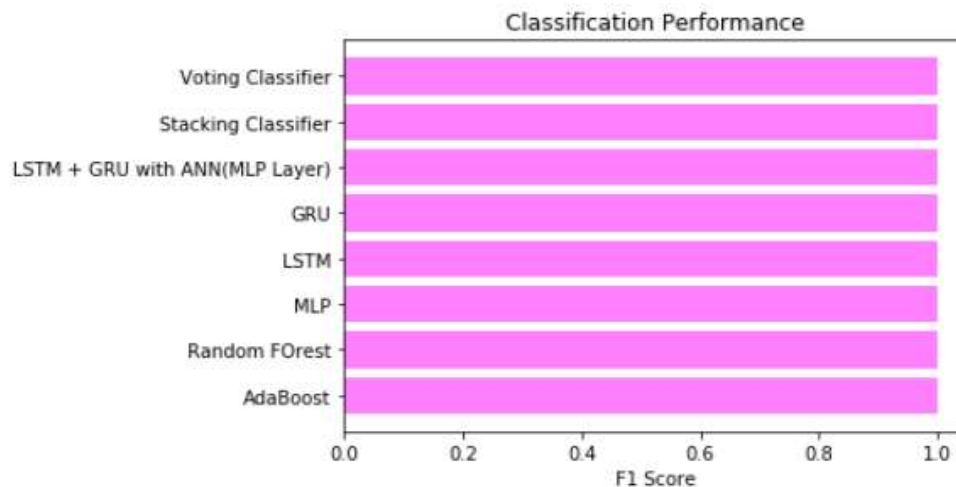


Fig 14: F1Score

	ML Model	Accuracy	Precision	Recall	F1-Score
0	AdaBoost	0.999	0.999	0.999	0.999
1	Random Forest	0.999	0.999	0.999	0.999
2	MLP	0.999	0.999	0.999	0.999
3	LSTM	0.998	1.000	0.998	0.999
4	GRU	0.998	1.000	0.998	0.999
5	LSTM+ GRU with ANN(MLP Layer)	0.998	1.000	0.998	0.999
6	Extension- Stacking Classifier	1.000	0.999	0.999	0.999
7	Extension- Voting Classifier	1.000	1.000	1.000	1.000

Fig 11: Performance Evaluation original dataset

	ML Model	Accuracy	Precision	Recall	F1-Score
0	AdaBoost	0.952	0.953	0.952	0.952
1	Random Forest	0.931	0.938	0.931	0.931
2	MLP	0.999	0.999	0.999	0.999
3	LSTM	0.499	1.000	0.499	0.666
4	GRU	0.499	1.000	0.499	0.666
5	LSTM + GRU with ANN(MLP Layer)	0.499	1.000	0.499	0.666
6	Extension- Stacking Classifier	1.000	1.000	1.000	1.000
7	Extension- Voting Classifier	1.000	1.000	1.000	1.000

Fig 12: Performance Evaluation SMOTE-ENN dataset

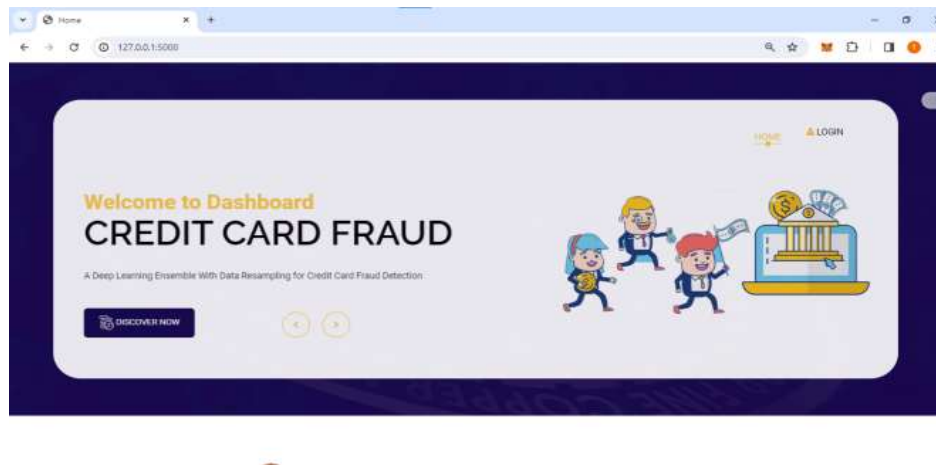


Fig 13: Home page



Fig 14: Login page

Fig 15: User input



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

Fig 16: Predict result for given input

4. Conclusion

This program effectively addresses the escalating issue of credit card fraud in the digital era, offering a substantial answer as an increasing number of individuals globally rely on digital transactions. This research employs diverse sampling techniques and data salting strategies to optimize the size of data files within the ML model. This indicates the need of meticulously organizing your data to enhance your model's effectiveness. The design and evaluation of several models such as “Adaboost, Random Forest, MLP, LSTM, GRU and LSTM + GRU + MLP showed its efficiency [8], [23], [24]”. While voting classifiers have overcome others, including adjustment and stacking classifiers as an extension of research showed an increase in accuracy. Adding file approaches caused fraud detection system to be much more accurate and reliable. The study has gained great results by focusing on teamwork between models, which shows that there is still room for improvement in this industry. Incorporating a user-friendly front-end interface with a Flask framework with user approval signifies that your project is committed to accessibility and support. This strategy guarantees the system's utility for the user by facilitating the identification and categorization of fraudulent transactions.

Future scope

Future research can examine the expansion of various LSTM integrations with various classifiers such as Randallswald, Utilization of logistic regression to enhance the precision of credit card recognition, alongside SVM [3]. An analysis of primary functions in future study will identify the most critical variables for detecting credit card fraud and enhancing efficient work recognition systems. The following studies can examine the analysis of risk factors to understand the basic components that contribute to credit card fraud. This knowledge helps create better opportunities to find things. Due to the proposed approach to deep nurturing of ensembles, scientists need to

examine the various model architectures, optimization approaches, and methods for tuning high parameters in order to better design the system. The proposed methodology can be employed to identify many forms of fraud beyond credit card theft, including insurance fraud and online trading fraud. This enhances the array of solutions to mitigate fraud.. Searching for ways to design and run Real Times can also help you find and stop financial transaction fraud.

References

1. M. Habibpour, H. Gharoun, M. Mehdipour, A. Tajally, H. Asgharnezhad, A. Shamsi, A. Khosravi, M. Shafie-Khah, S. Nahavandi, and J. P. S. Catalao, "Uncertainty-aware credit card fraud detection using deep learning," 2021, arXiv:2107.13508.
2. A. Cherif, A. Badhib, H. Ammar, S. Alshehri, M. Kalkatawi, and A. Imine, "Credit card fraud detection in the era of disruptive technologies: A systematic review," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 35, no. 1, pp. 145–174, Jan. 2023, doi: 10.1016/j.jksuci.2022.11.008.
3. G. Zhang, Z. Li, J. Huang, J. Wu, C. Zhou, and J. Yang, "eFraudCom: An e-commerce fraud detection system via competitive graph neural networks," *ACM Trans. Inf. Syst.*, vol. 40, no. 3, pp. 1–27, Mar. 2022, doi: 10.1145/3474379.
4. T. K. Dang, T. C. Tran, L. M. Tuan, and M. V. Tiep, "Machine learning based on resampling approaches and deep reinforcement learning for credit card fraud detection systems," *Appl. Sci.*, vol. 11, no. 21, p. 10004, Oct. 2021, doi: 10.3390/app112110004.
5. J. Chaquet-Ulledemolins, F.-J. Gimeno-Blanes, S. Moral-Rubio, S. MuñozRomero, and J.-L. Rojo-Álvarez, "On the black-box challenge for fraud detection using machine learning (I): Linear models and informative feature selection," *Appl. Sci.*, vol. 12, no. 7, p. 3328, Mar. 2022, doi: 10.3390/app12073328.
6. 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, doi: 10.3390/math10091480.
7. N. S. Alfaiz and S. M. Fati, "Enhanced credit card fraud detection model using machine learning," *Electronics*, vol. 11, no. 4, p. 662, Feb. 2022, doi: 10.3390/electronics11040662.
8. I. Benchaji, S. Douzi, B. El Ouahidi, and J. Jaafari, "Enhanced credit card fraud detection based on attention mechanism and LSTM deep model," *J. Big Data*, vol. 8, no. 1, p. 151, Dec. 2021, doi: 10.1186/s40537-021-00541-8.
9. E. Esenogho, I. D. Mienye, T. G. Swart, K. Aruleba, and G. Obaido, "A neural network ensemble with feature engineering for improved credit card fraud detection," *IEEE Access*, vol. 10, pp. 16400–16407, 2022, doi: 10.1109/ACCESS.2022.3148298.
10. E. Btoush, X. Zhou, R. Gururaian, K. Chan, and X. Tao, "A survey on credit card fraud detection techniques in banking industry for cyber security," in *Proc. 8th Int. Conf. Behav. Social Comput. (BESC)*, Oct. 2021, pp. 1–7, doi: 10.1109/BESC53957.2021.9635559.
11. I. D. Mienye and Y. Sun, "Performance analysis of cost-sensitive learning methods with application to imbalanced medical data," *Inform. Med. Unlocked*, vol. 25, Jan. 2021, Art. no. 100690, doi: 10.1016/j.imu.2021.100690.
12. S. A. Ebiaredoh-Mienye, T. G. Swart, E. Esenogho, and I. D. Mienye, "A machine learning method with filter-based feature selection for improved prediction of chronic kidney disease," *Bioengineering*, vol. 9, no. 8, p. 350, Jul. 2022, doi: 10.3390/bioengineering9080350.
13. C. Ho, Z. Zhao, X. F. Chen, J. Sauer, S. A. Saraf, R. Jialdasani, K. Taghipour, A. Sathe, L.-Y. Khor, K.-H. Lim, and W.-Q. Leow, "A promising deep learning-assistive algorithm for histopathological screening of colorectal cancer," *Sci. Rep.*, vol. 12, no. 1, pp. 1–9, Feb. 2022, doi: 10.1038/s41598-022-06264-x.
14. P. Goel, R. Jain, A. Nayyar, S. Singhal, and M. Srivastava, "Sarcasm detection using deep learning and ensemble learning," *Multimedia Tools Appl.*, vol. 81, no. 30, pp. 43229–43252, Dec. 2022, doi: 10.1007/s11042-022-12930-z.
15. I. D. Mienye, P. Kenneth Aina, I. D. Emmanuel, and E. Esenogho, "Sparse noise minimization in image classification using genetic algorithm and DenseNet," in *Proc. Conf. Inf. Commun. Technol. Soc. (ICTAS)*, Mar. 2021, pp. 103–108, doi: 10.1109/ICTAS50802.2021.9395014.
16. R. T. Aruleba, T. A. Adekiya, N. Ayawei, G. Obaido, K. Aruleba, I. D. Mienye, I. Aruleba, and B. Ogbuokiri, "COVID-19 diagnosis: A review of rapid antigen, RT-PCR and artificial intelligence methods," *Bioengineering*, vol. 9, no. 4, p. 153, Apr. 2022, doi: 10.3390/bioengineering9040153.

17. G. Nguyen, S. Dlugolinsky, M. Bobák, V. Tran, L. L. García, I. Heredia, P. Malík, and L. Hluchý, "Machine learning and deep learning frameworks and libraries for large-scale data mining: A survey," *Artif. Intell. Rev.*, vol. 52, no. 1, pp. 77–124, Jun. 2019. [Online]. Available: <http://link.springer.com/10.1007/s10462-018-09679-z>
18. S. O. Alhumoud and A. A. Al Wazrah, "Arabic sentiment analysis using recurrent neural networks: A review," *Artif. Intell. Rev.*, vol. 55, no. 1, pp. 707–748, Jan. 2022, doi: 10.1007/s10462-021-09989-9.
19. Z. Zhong, Y. Gao, Y. Zheng, B. Zheng, and I. Sato, "Real-world video deblurring: A benchmark dataset and an efficient recurrent neural network," *Int. J. Comput. Vis.*, vol. 131, no. 1, pp. 284–301, Jan. 2023, doi: 10.1007/s11263-022-01705-6.
20. J. Van Gompel, D. Spina, and C. Devellder, "Satellite based fault diagnosis of photovoltaic systems using recurrent neural networks," *Appl. Energy*, vol. 305, Jan. 2022, Art. no. 117874, doi: 10.1016/j.apenergy.2021.117874.
21. F. Shen, X. Zhao, G. Kou, and F. E. Alsaadi, "A new deep learning ensemble credit risk evaluation model with an improved synthetic minority oversampling technique," *Appl. Soft Comput.*, vol. 98, Jan. 2021, Art. no. 106852, doi: 10.1016/j.asoc.2020.106852.
22. A. Tsantekidis, N. Passalis, and A. Tefas, "Chapter 5—Recurrent neural networks," in *Deep Learning for Robot Perception and Cognition*, A. Iosifidis and A. Tefas, Eds. New York, NY, USA: Academic, 2022, pp. 101–115, doi: 10.1016/B978-0-32-385787-1.00010-5.
23. Y. Xie, G. Liu, C. Yan, C. Jiang, M. Zhou, and M. Li, "Learning transactional behavioral representations for credit card fraud detection," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Oct. 5, 2022, doi: 10.1109/TNNLS.2022.3208967.
24. Y.-C. Wei, Y.-X. Lai, and M.-E. Wu, "An evaluation of deep learning models for chargeback fraud detection in online games," *Cluster Comput.*, vol. 26, pp. 927–943, Jul. 2022, doi: 10.1007/s10586-022-03674-4.
25. S. Mishra, K. Shaw, D. Mishra, S. Patil, K. Kotecha, S. Kumar, and S. Bajaj, "Improving the accuracy of ensemble machine learning classification models using a novel bit-fusion algorithm for healthcare AI systems," *Frontiers Public Health*, vol. 10, May 2022, Art. no. 858282. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fpubh.2022.858282>
26. I. D. Mienye, G. Obaído, K. Aruleba, and O. A. Dada, "Enhanced prediction of chronic kidney disease using feature selection and boosted classifiers," in *Intelligent Systems Design and Applications*. Cham, Switzerland: Springer, 2022, pp. 527–537, doi: 10.1007/978-3-030-96308-8_49.
27. N. L. Fitriyani, M. Syafrudin, G. Alfian, C.-K. Yang, J. Rhee, and S. M. Ulyah, "Chronic disease prediction model using integration of DBSCAN, SMOTE-ENN, and random forest," in *Proc. ASU Int. Conf. Emerg. Technol. Sustainability Intell. Syst. (ICETIS)*, Jun. 2022, pp. 289–294, doi: 10.1109/ICETIS55481.2022.9888806.
28. J. Yang and J. Guan, "A heart disease prediction model based on feature optimization and smote-Xgboost algorithm," *Information*, vol. 13, no. 10, p. 475, Oct. 2022, doi: 10.3390/info13100475.
29. D. S. Sisodia, N. K. Reddy, and S. Bhandari, "Performance evaluation of class balancing techniques for credit card fraud detection," in *Proc. IEEE Int. Conf. Power, Control, Signals Instrum. Eng. (ICPCSI)*, Sep. 2017, pp. 2747–2752, doi: 10.1109/ICPCSI.2017.8392219.
30. H. Guan, Y. Zhang, M. Xian, H. D. Cheng, and X. Tang, "SMOTE-WENN: Solving class imbalance and small sample problems by oversampling and distance scaling," *Int. J. Speech Technol.*, vol. 51, no. 3, pp. 1394–1409, Mar. 2021, doi: 10.1007/s10489-020-01852-8.
31. L. Ni, J. Li, H. Xu, X. Wang, and J. Zhang, "Fraud feature boosting mechanism and spiral oversampling balancing technique for credit card fraud detection," *IEEE Trans. Computat. Social Syst.*, early access, Feb. 13, 2023, doi: 10.1109/TCSS.2023.3242149.
32. H. Fanai and H. Abbasimehr, "A novel combined approach based on deep autoencoder and deep classifiers for credit card fraud detection," *Exp. Syst. Appl.*, vol. 217, May 2023, Art. no. 119562, doi: 10.1016/j.eswa.2023.119562.
33. F. Itoo, 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, Aug. 2021, doi: 10.1007/s41870-020-00430-y.
34. S. K. Saddam Hussain, E. Sai Charan Reddy, K. G. Akshay, and T. Akanksha, "Fraud detection in credit card transactions using SVM and random forest algorithms," in *Proc. 5th Int. Conf. I-SMAC (IoT Social, Mobile, Analytics Cloud) (I-SMAC)*, Nov. 2021, pp. 1013–1017, doi: 10.1109/I-SMAC52330.2021.9640631.
35. I. D. Mienye, Y. Sun, Z. Wang, "Prediction performance of improved decision tree-based algorithms: A review," *Proc. Manuf.*, vol. 35, pp. 698–703, Jan. 2019, doi: 10.1016/j.promfg.2019.06.011.

36. 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, doi: 10.1109/ACCESS.2018.2806420.
37. T.-H. Lin and J.-R. Jiang, "Credit card fraud detection with autoencoder and probabilistic random forest," *Mathematics*, vol. 9, no. 21, p. 2683, Oct. 2021, doi: 10.3390/math9212683.
38. A. Rb and S. K. Kr, "Credit card fraud detection using artificial neural network," *Global Transitions Proc.*, vol. 2, no. 1, pp. 35–41, Jun. 2021, doi: 10.1016/j.gltp.2021.01.006.
39. S. C. Dubey, K. S. Mundhe, and A. A. Kadam, "Credit card fraud detection using artificial neural network and BackPropagation," in *Proc. 4th Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, May 2020, pp. 268–273, doi: 10.1109/ICICCS48265.2020.9120957.
40. O. N. Akande, S. Misra, H. B. Akande, J. Oluranti, and R. Damasevicius, "A supervised approach to credit card fraud detection using an artificial neural network," in *Applied Informatics*. Cham, Switzerland: Springer, 2021, pp. 13–25, doi: 10.1007/978-3-030-89654-6_2.