International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6640
***Article Received**: 24 February 2023 **Revised**: 02 April 2023 **Accepted**: 21 April 2023*

CCNN: An Artificial Intelligent based Classifier to Credit Card Fraud Detection System with Optimized Cognitive Learning Model

L Vetrivendan[1], Ganesh Kumar[2]
[1]Department of Computer Science and Engineering
Galgotias University
Noida, India
vetrivendan21@gmail.com
[2]Department of Computer Science and Engineering
Galgotias University
Noida, India
tganeshphd@yahoo.com

Abstract— Nowadays digital transactions play a vital role in money transaction processes. Last 5 years statistical report portrays the growth of internet money transaction especially credit card and unified payments interface. Mean time increasing numerous banking threats and digital transaction fraud rates also growing significantly. Data engineering techniques provide ultra supports to detect credit card forgery problems in online and offline mode transactions. This credit card fraud detection (CCFD) and prevention-based data processing issues raising because of two major reasons first, classification rate of legitimate and forgery uses is frequently changing, and next one is fraud detection dataset values are vastly asymmetric. Through this research work investigating performance of various existing classifier with our proposed cognitive convolutional neural network (CCNN) classifier. Existing classifiers like Logistic Regression (LR), K-nearest neighbor (KNN), Decision Tree (DT) and Support Vector Machine (SVM). These models are facing various challenges of low performance rate and high complexity because of low hit rate and accuracy. Through this research work we introduce cognitive learning-based CCNN classifier methodology with artificial intelligence for achieve maximum accuracy rate and minimal complexity issues. For experimental data analysis uses dataset of credit card transactions attained from specific region cardholders containing 284500 transactions and its various features. Also, this dataset contains unstructured and non-dimensional data are converted into structured data with the help of over sample and under sample method. Performance analysis shows proposed CCNN classifier model provide significant improvement on accuracy, specificity, sensitivity and hit rate. The results are shown in comparison. After cross-validation, the accuracy of the CCNN classification algorithm model for transaction fraudulent detection archived 99% which using the over-sampling model.

Keywords- CCFD, Machine learning, cross-validation, support vector machine, classification, under sampling.

I. INTRODUCTION

Over the past year of study, we have been following the news on information security and financial fraud as it is essential to all online and offline financial transaction systems. Although fraudulent transactions account for a relatively small percentage of most medium credit card transactions, as soon as a customer is unfortunate enough to have a credit card transaction, the loss of money to the business and a crisis of trust for the customer can ensue. Some reports show that Credit card fraud (CCF) can easily accomplish their purpose. Large amounts of money can transact in a short period without any indication of risk and the owner's permission. Every fraudulent transaction can be legitimized by a fraudster's operation which makes fraud very challenging and difficult to detect [1]. As a result, we are sufficiently motivated to want to improve CCFD by training a pass-through machine learning (ML) classification method. The final purpose is to help this research work to select a better model. The banks want to detect credit card transactions and quickly predict whether the trade is risky, regulators need to delay or hold the transaction, and the marketing needs to be blocked the next time the card used a lot. We think we have ambitions to complete the fraud detection system. Besides, we hope we achieve an opportunity to realize the need for improved customer detection capabilities. The popularity of credit cards has greatly facilitated transactions for both merchants and users but it has also led to many cases of fraud. There are two types of fraud on the market today. Card-present fraud is now less common to buy the other kind of deception, and absent card fraud is currently widespread. They may execute in many ways, usually occurring without the cardholder's knowledge. The maintenance of the security of the Internet database has always been a big problem. A slight leak will cause the threat

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6640
***Article Received**: 24 February 2023 **Revised**: 02 April 2023 **Accepted**: 21 April 2023*

of stolen card information on the user's account. Traditional CCFD models such as manual detection, expert rules, cost analysis models. For example, they might have shortcomings such as low detection accuracy, long detection time, and high maintenance costs. Therefore, financial institutions urgently need a well- designed fraud

II. LITERATURE REVIEW

In the study [2][3] they used a dataset from the European trading market, containing 284807 trades. They used a hybrid technique of under-sampling drinking oversampling, implemented in Python, and used three classifiers for training. The accuracy of KNN and logistic regression was 90.69% and 54.86%. The results from his experimental study indicated that KNN performs better than all other linking techniques. It can provide us with a reference, the reason why the logistic regression is so low, and a way to adjust the KNN accuracy. From the study [4][5], This study is in 2011 and it based on a comparison of ANN and logistic regression (LR) models. The study compares the performance of CCFD while comparing their performance on a test dataset.

A. Credit Card Fraud definition

Initially, we need to understand is: why is it that modern detection systems, anti-fraud detection, are so complicated? The modern detection system, also we call it Anti-fraud programs, for most customers or owners, they probably do not have a clear definition with the CCF. In other words, the purpose of fraud is vague. On a small scale, anti-fraud seems to be a dichotomous problem [6]. However, after repeated deliberation, we found that it is a multi-classification problem because each type of fraud can be treated as a different type. Besides, the single kind of fraud does not exist, and the means of the second phase fraud is always changing. Even now, most of our customers, banks and insurance companies are perennial victims of fraud. They must continually try to update their prediction system. Rather than betting on same model, so fraud detection is also facing this challenge right now.

CCF anatomy

According to Seeja and Zareapoor, there are two main phases for CCFD [7].

- The dataset which we use is labelled so that we can use the more mature supervised learning [27], but there is a disadvantage that it will be slower to update over time.
- There is a significant risk of supervised learning with labels, the model learned from such historical data can only detect frauds that are similar to historical fraud.

To accomplish the task of improving the accuracy of credit card detection, we may need some research to deal with the tags and characteristics of the information we collect, and we may need to do data mining to find information that is beneficial to us.

B. Credit Card fault detection and prevention

This system is designed to prevent any unauthorized credit card transactions from fraudsters and to recover losses and credibility for customers and businesses. Although there are better financial mechanisms, the fraudster is continually updating his techniques. Also, it makes the anti-CCF techniques very challenging; the standard anti CCF methods available in the market today are listed below.

Validation method through merchant trade

The merchants often require a complete list of receipts to identify the user and have added tokenization techniques to protect credit card information by using the referenced card number instead of the current card number. It can make sure that they offer additional information. Also, they may be requested to show them during the merchant transaction, and they are currently used by merchants to combat fraud [8].

C. Using Neural Networks to Detect Online Payment Fraud

Most of the existing techniques based on deep learning and oversampling algorithms for CCFD. The Long Short Term Memory Networks [24] (LSTM) fraud detection model for serial classification of transaction data and integration of synthetic minority class oversampling. The Smote with KNN classification algorithm design and build a KNN-Smote-LSTM based fraud detection network model which can Improve fraud detection performance by continuously filtering out security-generating samples through KNN discriminant classifiers [9].

ML detection: ML is a very effective way to detect fraudulent transactions if his performance is good enough because he determined by choice of features, the training of the data drink testing, and the classification methods of ML. All of these factors contribute to different generation rates. Many studies have shown that using ML classification algorithms to detect CCF has resulted in better accuracy. They have also compared the results of different algorithms and other studies and agreed that ML detection is the right choice.

D. Decision tree (DT)

The use of decision tree is judges the feasibility of the decision analysis method. Then we know that because this decision branch is drawn as a graph much like the trunk of a tree, we name it a decision tree (figure 1).

Decision trees are a primary classification and regression method, and learning. Classification tree (decision tree) is a very commonly used classification method. Similar to the dataset classification problem mentioned in this paper, the decision tree is a technique that is often used to analyze data and can also be used to make predictions. That is why we chose it for the training of the fraud detection system [5]. That

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6640
***Article Received**: 24 February 2023 **Revised**: 02 April 2023 **Accepted**: 21 April 2023*

is a simple decision tree classification model: the red boxes are features.

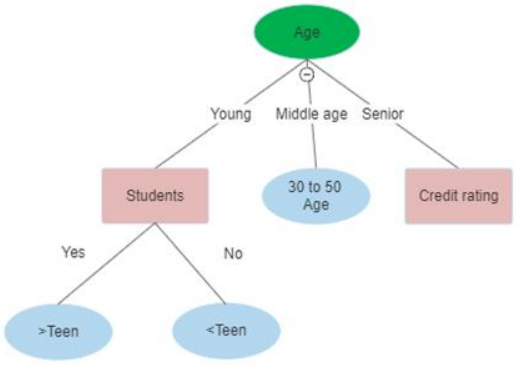

Figure 1: Prediction procedure with decision tree

E. KNN Classification

The term K nearest neighbor mean K nearest neighbor which says that its closest K neighboring values can represent each sample. The nearest neighbor algorithm is a method of classifying every record in a data set. The implementation principle of KNN nearest neighbor classification algorithm is: to determine the Category of unknown samples by taking all the examples of known types as a reference and at the same time calculate vector values between proposed models and all the available pieces, from which the nearest K has known examples are selected, according to the rule of majority-majority-voting, the unknown samples and the K nearest models belong to a category with more categories [12].

$$d\big((x_1, x_2, \dots x_n), (y_1, y_2, \dots y_n)\big) = \left(\sum_{i-1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}} \quad (1)$$

The K value of the KNN algorithm in 'scikit-learn' is adjusted by the neighbor's parameter, and the default value is 5. As shown in the figure 2 below, how do people determine which Category a green circle should belong to red, green and blue. If K=3, the green process will be judged to belong to the red triangle class because the proportion of red triangles is 2/3, and if K =5, the green circle will be considered to belong to the blue square class because the ratio of blue squares is 3/5 [11].

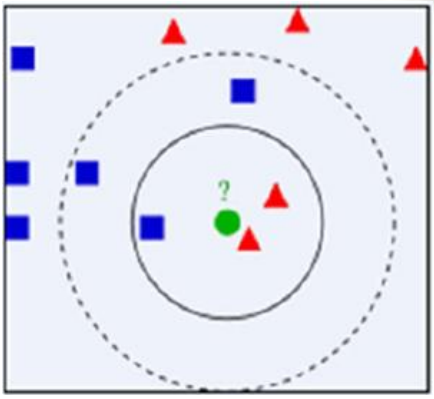

Figure 2: Kth nearest neighbor classifier

F. Logistic regression algorithm

Use logistic regression [13][14] to detect CCF. Logistic regression is the classical and the best bicategorical algorithm which is preferred when dealing with classification problems, especially bicategorical ones. The choice of algorithm is based on the principle of simplicity before complexity. Logistic regression is also an excellent choice because it is a recognized statistical method used to predict the outcome of a binomial or polynomial. A multinomial logistic regression algorithm can regenerate the model. It will be a better classification algorithm when the target field or data is a set field with two or more possible values.

The advantage of logistic regression is that he is faster to process and is suitable for bicategorical problems. It is also more straightforward for any beginner to understand and directly see the weights of each feature. Then it is easier to update the model and incorporate new data for different problems [15]. Furthermore, it has a disadvantage. There is a limit to the data and the adaptability of the scene. Not as adaptable as the decision tree algorithm. But this is an issue that we can also determine the actual situation whether the logistic regression has a better ability to adapt to an extensive data set of credit card transactions [16].

The main methods of logistic regression method:

It is to look for some risk factor, then in this research work, they want to find a particular transaction factor or reasons that are suspected of being fraudulent.

Prediction: Predicting the probability of fraud under other independent variables, based on different algorithmic models.

Judgment: It is somewhat similar to prediction. It is also based on different models to see how likely it is that a transaction is a risk factor in a situation where fraud falls into a specific category.

III. CCF IDENTIFICATION

The identification of CCFD is currently facing challenging because of most people not familiar with CCF. After all, most of the scam comes out through the valid pathway following the banks as well as financial companies, and the only difference is that they are unauthorized third-party pathways [26]. The recent credit fraud, as well as becomes more challenging to identify. Because if there has anyone who knows them credit card number, as well as expiration date, he can make a transaction on the website without them permission. Fraudsters will get more information about people's finances, and they will also have more opportunities to make fraudulent transactions by swiping credit cards, rather than just the ones we see.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6640
***Article Received**: 24 February 2023 **Revised**: 02 April 2023 **Accepted**: 21 April 2023*

A. Consequences of credit card problems

Credit card security problems and process directly concern the user and the financial company; it is a reason we keep focus CCF this year. The following are examples of fraud transaction outcomes.

1. Economic losses to users and businesses
2. Customer Personal Information Breach and Corporate Disclosure Enterprise trust crisis in information security.

ML classifiers

In this research work, we used a total of five classifications methods CCNN, KNN, SVM, DT, Logistic and Linear regression. These classification algorithm methods are widely used for problems such as differential training dataset. Also, it commonly used in classification learning. That is the reason I compare them in the same training dataset. Also, it can be a cross-sectional comparison with other current studies in the final results.

IV. PROPOSED COGNITIVE CONVOLUTIONAL NEURAL NETWORK-BASED CLASSIFIER (CCNA)

A. Finding the h-function (i.e., the prediction function)

Constructing the predictive function h(x), the logistic function, or also known as the sigmoid function, we generally the basic process to build the predictive analysis, where training data utilized to calculate the vector and its aggregate values, as well as the best parameters. The basic form of the function shown in figure 3.

$$g(z) = \frac{1}{1+e^{-z}} \quad (2)$$

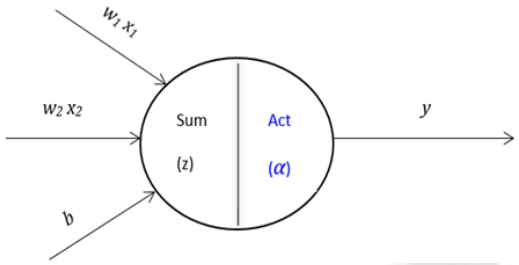

Figure 3: Expression of Activation function

B. Constructing the J-function (loss function)

The second step is that we need to construct the loss function-j. In general, there will be m samples, each with n characteristics. [22] [23] The Cost and J functions are as follows, and they are derived based on maximum likelihood estimation.

$$h_\theta(x), y = \begin{cases} -\log(h_\theta(x)) & if\ y = 1 \\ -\log(1 - h_\theta(x)) & if\ y = 0 \end{cases}$$

$$J_\theta = \frac{1}{m}\sum_{i=1}^{m}(y_i \log\ h_\theta(x_i) + (1 - y_i)(1 - \log h_\theta(x_i))) \quad (3)$$

The J-function minimal and find the regression parameter (θ)

The final step is that we, using gradient descent, solve for the minimum value of θ. The process of updating θ can then be summarized as follows

$$\theta_j := \theta_j - \frac{1}{m}(h_\theta(x_i) - y_i)x_i^j \quad (4)$$

Cognitive convolutional Neural Network

For CCNA, [30] a ML library open-sourced and artificial intelligence-based Gradient Boosting Categorical Features. The name CCNA comes from two words "Cognitive learning" and "Convolution". As mentioned earlier, the library is a universal library of gradient boosting algorithms which contains many tree type algorithms [25]. For example, it can handle a variety of category-type data well and is a library of gradient boosting algorithms that can handle category-type features well. That is the reason why we finally chose this algorithm. We wanted to compare the performance of the comprehensive library with the first four individual classification algorithms, including DT and cat boost, also possesses some of these features.

The CCNN has some of the following advantages:

- The CCNN has a unique way of dealing with categorical features with cognitive learning [29]. First, it does some statistics on the categories and calculates the frequency of a type, such as the fraudulent transaction class in this question and then adds hyper-parameters to generate new numerical features.
- The CCNN algorithm works with artificial intelligence (AI) based cognitive learning which adapts the context based on the structural data. It can handle both Category and numerical features and uses combined category features that can take advantage of the links between elements which significantly enriches the feature dimension. However, CCNN has been optimized to use other algorithms to prevent overfitting of the model. That is why the proposed algorithm can rival any advanced AI based cognitive learning algorithm in terms of performance.
- CCNN is easy to use: CCNN provides a Python interface for integration with scikit, as well as R and command-line interfaces which facilitate quick calls and reduce the number of calls. Also holds a custom loss function which also reflects his extensibility.

D. CCNA algorithm:

Algorithm 1: Cognitive Convolution neural network

Input: training data, activation module, bias. $(x_i, y_i) \subseteq N$, predict function and loss function

Iteration: hidden layer (oversampling, slicing)

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6640
***Article Received**: 24 February 2023 **Revised**: 02 April 2023 **Accepted**: 21 April 2023*

$\mathrm{CCNN}_{\mathrm{best}}$
$=$ cognitive with fuzzy rate of all attributes from CCFD dataset as nodes and edges
$\mathrm{bst}_{\mathrm{value}} = \mathrm{CCNN}_{\mathrm{best}}$ with joint probability distribution (best score) of predictive attributes
do
$features\ fail\ =\ end$
$CCNN_{score} = \mathrm{CCNN}_{\mathrm{best}}$
$rate = bst\_value$
$CCNN_{deviation}$
$= set\ of\ outcomes\ and\ its\ feature\ engineering\ value$
$(reversing\ an\ edge\ based\ cognitive\ decisions)$
$for\ bst\_value\ in\ CCNN\_deviation$
do
$if(bst\_value > rate)$
$rate = bst_{value}$
$CCNN$
$= calculate \begin{pmatrix} F: rate, feature\ extraction, possible \\ combinations, hit\ rate \end{pmatrix}$
$end\ if$
$rate\ compare\ with\ next\ values\ an\ find\ CCNNbest$
$end = true$
$cognparam(CCNN, conditional\ probability\ table, bst_{value})$
$retrun\ \mathrm{CCNN}_{\mathrm{best}}$ (5)

E. Synthetic Minority Over-Sampling Technique (SMOTE):

SMOTE [14] algorithm is used to increase the quality of random oversampling technique where synthetic samples are generated for the minority classes. The class-imbalance problem that we need to solve next in this research work refers unbalanced distribution of various necessary credit card transaction classes in training process [22][23]. For example, for a binary problem with 1000 training samples, ideally, the number of positive and negative models are similar [18]; if there are 995 positive samples and only five negative samples, it means there is class- imbalance. There is also the case for the dataset in this research work. For now, there are three main approaches.

Adjusting the value of θ

Adjust the value of θ according to the proportion of both types of samples in CCFD training values. It is done based on the assumptions made about the training set, as described above. However, whether this assumption holds in the given task is open to discussion.

Over sampling

The classes with a small number of samples inside the training set (few types) are oversampled, and new models are synthesized to mitigate class imbalance.

Under sampling

Under-sampling of classes with huge count of samples inside the training set (most categories), discarding some examples to mitigate class imbalance.

In this research work, we use oversampling and under sampling to perform comparison operations. At the same time, we can also compare the results to analyze whether the two methods are more suitable for this research work's dataset, and what are the advantages and disadvantages of each technique [17]. The core idea of SMOTE in a nutshell is to interpolate between minority class samples to generate additional models.

$$x_{new} = x_i + (mean\ x_i - x_i) * \delta \quad (6)$$

Where mean x_i is the elected k-nearest neighbour point, and δ ∈ [0,1] is a random number. An example of a SMOTE-generated sample, using k-nearest neighbor, is shown in the following figure 4which shows that the SMOTE-generated model generally lies on the line connected by x_i and $mean\ x_i$

Figure 4: SMOTE formation sample (under and over)

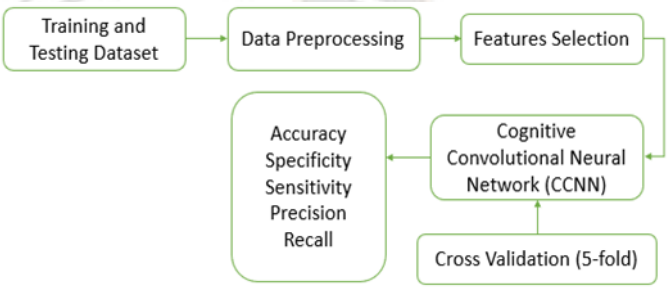

Figure 5: Overall framework of Cognitive Convolutional Neural Networks model

Dataset preparation and pre-processing

There is one dataset of CCFD from the kaggle.com. The dataset that contains data from credit cardholders using credit cards for transactions in September 2013. Also, this dataset is complete shows all transactions that took place over two days. This dataset has a memory size of 166Mb. It is a straightforward piece of data mining and self-classification done in the format of packet format.cvs. We can put it directly on our local hard drive or a network drive and use it for direct

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6640
***Article Received**: 24 February 2023 **Revised**: 02 April 2023 **Accepted**: 21 April 2023*

access. There will be more details on the content of the relationship dataset.

General components of the CCFD dataset

From this research analysis of CCFD dataset we obtained, the first thing we can do is open directly, and it was showing that in two days European cardholders made a total of 284,908 transactions via credit cards, of which only 492 were fraud transactions. The CCFD dataset portrays a highly unbalanced fraudulent transaction profile.

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	0.090794	-0.551600	-0.617801	-0.991390	-0.311169	1.468177
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	-0.166974	1.612727	1.065235	0.489095	-0.143772	0.635558
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	0.207643	0.624501	0.066084	0.717293	-0.165946	2.345865
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	-0.054952	-0.226487	0.178228	0.507757	-0.287924	-0.631418
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	0.753074	-0.822843	0.538196	1.345852	-1.119670	0.175121

	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
	1.468177	-0.470401	0.207971	0.025791	0.403993	0.251412	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0
	0.635558	0.463917	-0.114805	-0.183361	-0.145783	-0.069083	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0
	2.345865	-2.890083	1.109969	-0.121359	-2.261857	0.524980	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
	-0.631418	-1.059647	-0.684093	1.965775	-1.232622	-0.208038	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0
	0.175121	-0.451449	-0.237033	-0.038195	0.803487	0.408542	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99	0

Figure 6: Original CCF dataset picture

Table 1: Attributes of the European dataset

S.No	Features	Functions
1	Overall/Individual Time	Overall round-trip time value and overall transactions time
2	Overall Transactional Value	Every transactional value
3	Classification threshold	Anomaly detection is 1 if not means 0

Then performed the data review process in the pre-processing data section of the data as the code is shown in table 1, data. IsNull() checks for missing values and the result is 0, so the data set is a good one that doesn't need to be processed for complementary values and can be used straight away.

$$data.isnull().sum().max() \quad (7)$$

By observing the statistical information of the data: it was found that the mean, maximum, minimum, median, etc. of Time and Amount are very different from V1-V28, and the mean values of V1-V28 and Class are concentrated around 0

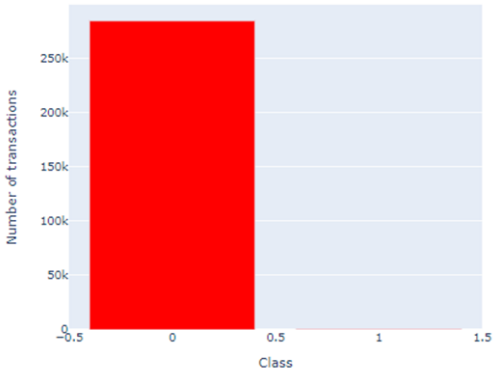

Figure 7: Number of categories of inspection sample

The variance of the data is in the range of 0-1. It means that the information is evenly distributed, and the data of the amount dimension is very unevenly distributed, and the data of the Amount column is too big floating. At the same time, the scaling is different from V1-V28, so in the process of ML, we need to ensure that the difference between the eigenvalues cannot be too large, so we need to carry out feature scaling standardization on the amount. Also, we observed that the Time class which is a counting function, is not very useful for this research work, and that after all the Amounts are standardized, the delete operation can also be performed. Then we counted the number of categories of regular and fraudulent consumption, as shown in the figure 7 above. 0 indicates typical consumption, 1 tells fraudulent consumption, and the histogram shows that the amount of fraudulent data is minimal, while regular consumption is enormous. It is important to note that if we build the model directly with this unbalanced data, the model will be inferior at predicting the small number of samples, so we will later balance the examples using the sampling Up/Throughout Technique.

V FEATURE ENGINEERING

In this feature engineering, we will complete the feature engineering by constructing a diagram to get a comprehensive view of the overall distribution of the data. Also, it might need to extract as many features as possible from the raw data for use by the algorithm and the model, and to integrate, select, and scale the elements for better performance. In this research work, through proposed model improve the accuracy and precision of subsequent model training by conducting feature engineering on CCFD dataset. Figure 8 portrays that a comparison of time dimensions of the fraud and standard classes shows that the time distribution of regular transactions varies with some regularity. At the same time, there is no obvious time pattern for fraud transactions.

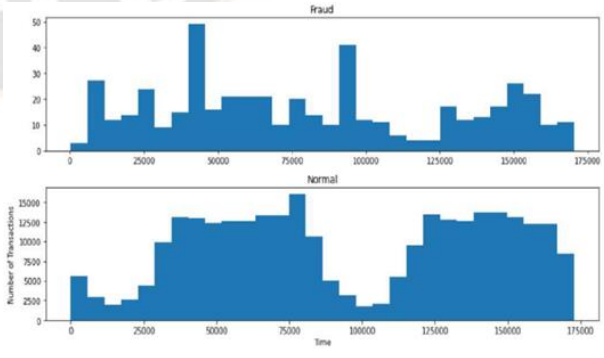

Figure 8: Fraud and regular classes dimensionality comparison

Then there is a comparison between order amounts, where both types of transactions exist in the form of long-tail distributions, but in terms of the number of fraudulent orders

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6640
***Article Received**: 24 February 2023 **Revised**: 02 April 2023 **Accepted**: 21 April 2023*

are mostly small orders, generally less than 1000, while the positive transactions range from 0-15000. Then, Figure 9 shows that plotting a scatter plot of time versus the amount. And it shows that regular transactions are evenly distributed across points in time, and outliers for transaction amounts are less frequent. In contrast, fraudulent transactions are scattered across time, and outliers occur more frequently.

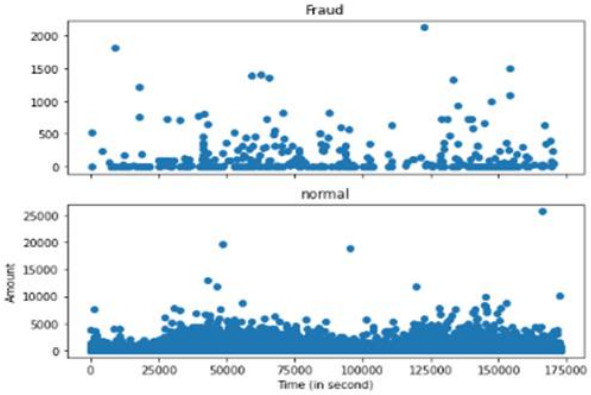

Figure 9: Scatter plot graph (time versus amount)

Next, the distribution of each of the remaining PCA-processed features will be exported, and the distribution of each element within the standard and fraud classes will be observed. We observed that the distribution of V6, V8, V13, V20, V22, V23, V24, V25, V26 is very similar in both categories, and the similar shape of the distribution means that the feature has little impact on the final prediction results, so it is deleted

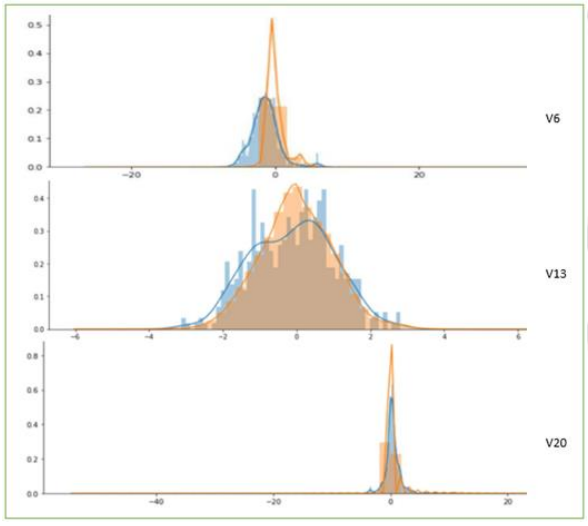

Figure 10: Various transactions comparison

The last step, because we mentioned that the data in the Amount column floats too much, in the process of doing ML. We need to ensure that the eigenvalue difference cannot be too large, the pre-process on CCFD dataset is amount and standardize the data. Normalized the two dimensions amount and hour by using the mean-standard deviation method.

Table 2: Data after standardization

	V1	V2	V3	…V28	Class	Norm_ Amount
0	-1.3598	-0.0727	2.5363	-0.0210	0	0.2449
1	1.1918	0.2661	0.1664	0.0147	0	-0.3424
2	-1.3583	-1.3401	1.7732	-0.0597	0	1.1606
3	-0.9662	-0.1852	1.7929	0.0614	0	0.1405
4	-1.1582	0.8777	1.5487	0.2151	1	-0.0734
[5 rows X 30 columns]						

VI IMPLEMENTATION

Python libraries for data science and ML.

This article uses the Python programming language for implementation. As you know the same, Python language, in addition to using basic pandas, NumPy and other open-source libraries, user data analysis and data mining the most essential one Python library. Then he is an efficient and straightforward open-source library. It is built on NumPy and other Python libraries on top [19]. And he contains, classification, regression, clustering, dimensionality reduction, model selection and drinking pre-processing and other functions. Can save developers a lot of time and work. From this research work, the library of proposed classifier should be improved. It is an all-purpose algorithm library, and to use this library, you need to download and install the cat boost package first.

Segmentation/reservation of the original training set

Before start training, preprocessing techniques have to apply on dataset like noise removal, data reduction, data transformation, data cleaning, split and reserve processing. The purpose of dynamite is that since based on the needs of samples for training and testing which will change the original dataset, need to reserve a copy of the data first, and the StratifiedShuffleSplit method used to shuffle dataset ramdomly. It is the combined form of StratifiedKFold and ShuffleSplit which ensures that each fold has the same proportion of samples for each Category while messing up the models randomly and dividing up the train/test pairs based on parameters. Because only in this way can maintain the original imbalance of the test set which is essential because these test set values used in validation of the prediction results.

$$sss = StratifiedShuffleSplit(n_{splits} = 5, test_{size} = 2, random_{state} = 42) \quad (8)$$

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6640
***Article Received**: 24 February 2023 **Revised**: 02 April 2023 **Accepted**: 21 April 2023*

Sampling process

The target column Class presents a massive sample imbalance which can cause problems for model learning. In this research work, used the SMOTE (Synthetic Minority Oversampling Technique) to handle the sample imbalance.

Under sampling: Start with under sampling which is actually very simple, just randomly draw the same number of samples from a huge count of pieces reduced into necessary minimal count of samples. Here generates a new dataset called data_new, and then train the machine with various necessary feature selection process. After using under sampling, the proportion of standard and fraudulent transactions was 50 per cent and 50 per cent, and after sample reduction, the final sample size of trades was 984.

Table 3: The dataset after under sampling

S.No	Time	V1	V2	V3	V4	V5	V6	V10	V13	V20	V23	V28	Amount	Class
0	0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.090794	-0.99139	0.251412	-0.110474	-0.021053	149.62	0
1	0	1.191857	0.266151	0.16648	0.448154	0.060018	-0.082361	-0.166974	0.489095	-0.069083	0.101288	0.014724	2.69	0
2	1	-1.358354	-1.340163	1.773209	0.37978	-0.503198	1.800499	0.207643	0.717293	0.52498	0.909412	-0.059752	378.66	0
3	1	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	-0.054952	0.507757	-0.208038	-0.190321	0.061458	123.5	0
4	2	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.753074	1.345852	0.408542	-0.137458	0.215153	69.99	0

Let's use the sns.countplot function to see how the sample categories are classified after sampling.

Oversampling

From the study which includes the principle of oversampling is increasing positive samples count, makes count of positive and negative models. Then they were learning to process the data, constructing oversampled data [20]. After oversampling the data set, samples count of '1' is 287454. Samples count is '0' has also increased to 287454. 50% of each, for a total sample size of 574908.

The SMOTE algorithm for up-sampling, and as an up-sampling technique, the SMOTE algorithm does not simply copy the original small number of samples. But it can select an interval for each of its features that

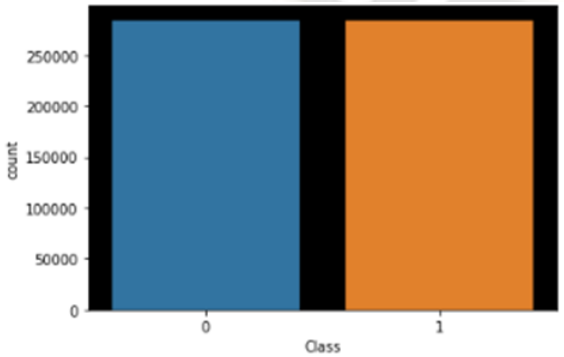

Figure 11: Sample classification

fluctuates by a small margin, performs new feature generation, and combines these features to generate new samples of that class. The models developed using this technique are much more common-sense. SMOTE is already well encapsulated in the same imblearn package.

Since our original logistic regression is built based on the down sampled dataset, the selected parameters may not be suitable for the up sampled data. So, need to find the optimal parameters for it again, for imbalanced random search uses 'RandomizedSearchCV' for tuning, compared to GridSearchCV, RandonizedSearchCV will not try all the parameters so that it will consume less memory and time.

Individual classifiers evaluation

Before building the model, need to do two functions, they are slice and dice the dataset. Slice the dataset into $x_{train}, x_{test}, y_{train}, y_{test},$

$$x_{train}, x_{test}, y_{train}, y_{test} = train\_test\_split(x_{new}, y_{new}, test_{size} = .2, random_{state} = 42) \quad (9)$$

prepared four pre-selected models, observed their predictive performance, and chose the best performing model. The five models are LR, KNN, SVM and DT and CCNNs.

Table 4: First training accuracy

CCF Prediction Classifications	Accuracy
LR Classification	94%
KNN Classification	93%
SVM Classification	93%
DR Classification	90%
Cognitive CNN	95.6%

Through this research work found that the logistic regression model and CCNN work better, and it is a surprising bonus that the simple model of logistic regression is no worse than the complex model. The performance of proposed classification CCNN model is as good as ever; after all, it is a combination of various optimized algorithms. But here we're using models with default parameters, and improve the quality of parameters for each model and then verify their accuracy. And then we're building the model from this using the accuracy, not the recall (%). The reason for CCF that data from post-sampling category balancing is now used, and accuracy has a better assessment and is more convincing

Classifier evaluation with cross-validation

The shortcoming of our model training is that our model training and testing are conducted on the same data set which leads to overfitting of the model. So, dividing overall dataset and samples into various unique cluster classes. The cross-validation method partitions the data set. The following figure shows the parameter setting code for the five models. Maximum cross validation models use grid search to

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6640
***Article Received**: 24 February 2023 **Revised**: 02 April 2023 **Accepted**: 21 April 2023*

construct a candidate set of parameters. Then grid search will exhaust various combinations of parameters to find the best location of settings according to the scoring mechanism of the set evaluation. In grid search, cognitive process adjusts two parameters, C and kernel, where 'C' is the penalty parameter C. If the default value is 1.0, the higher C is equal to the penalty relaxation variable. The relaxation variable value is very nearer to zero, i.e., the penalty for misclassification increases. It tends to be the case that the training set is fully split into pairs which is very accurate when testing the training set but has weak generalization ability.

The kernel arguments represent the form of the kernel function which is 'rbf' by default, but can also be 'linear', 'poly', or 'kernel'., 'rbf', 'sigmoid', 'precomputed', conducted experiments with 5-fold cv, and model accuracy was assessed using f1 -We set the range for C to be [0.01, 0.1, 1, 10, 100], and the range for kernel to be ['linear', 'poly', 'linear', and 'poly']. The best outcome can get with 'kernel' = 'linear', 'C' = 0.01, and accuracy of parametric model instead of 'kernel' = 'linear', 'rbf', 'sigmoid'].

Table 5: Cross-validation of results after parameterization

Classifier	Accuracy (before cross validation)	Accuracy (after cross validation)	Difference
Logistic Regression (LR)	94%	94.78%	0.78%
Knowledge nearest neighbor (KNN)	93%	93.52%	0.52%
Support Vector Machine (SVM)	93%	93.14%	0.14%
Decision Tree Classifier (DTC)	90%	92.25%	0.25%
Cognitive CNN (CCNN)	95.6%	95.93%	0.33%

We can see that the accuracy of each model is improved to a certain extent after the parameter adjustment. And the results show that logistic regression is still the most suitable model for this task, followed by a vector machine, KNN, CCNN, and the worst is the decision tree. The accuracy of the SVM classifier is 94.78%. The default parameter SVM classifier in the same test set on the accuracy is 94.0%, Table 6 portrays confusion metrics of CCFD, after the tuning model accuracy improved 0.78%. Move to the next, the k-nearest (KNN) accuracy after cross-validation is 93.52%. Also, the accuracy before that is 93.0% which improved 0.52% in after parameter setting. The accuracy rate of SVM is 93.14% which is not much higher than that before parameters were not adjusted, only 0.14% higher than that before 93.0%. Then the accuracy rate was the second to last. The decision tree was the worst in the detection performance before. But after the adjustment, it was much improved, from 90% to 92.25%. Unfortunately, its accuracy was still at the bottom. Finally, our CCNN Classifier didn't get much of a cognitive decision, either, at 0.33%. From 95.6% to 95.93%. But it is still the third most accurate. However, the accuracy difference between logistic regression and vector machine is not massive; we can observe the degree of fit of each model according to their learning curve, to choose the best model. According to the research, we can see that the test set accuracy of logistic regression and the training set accuracy are always close to each other which means that the models are not falling into overfitting or under fitting. In contrast, the training set accuracy of the vector machine is higher, with some slight overfitting, to this point, we choose logistic regression as our prediction model for this task.

Model predicts real data

Next, we need to build a good down sampling model to predict our real data which will use the normal_data we reserved at the beginning and the StratifiedShuffleSplit with the parameters we set. An unbalanced test set is the only way the predictions will make sense. There are also several new methods we'll use in 'imblearn', including NearMiss which is a wrapped down sampling method in 'imblearn', make_pipeline which is similar to the pipeline mechanism in sklearn, but this is set up separately for sampling, and the model which selects the optimal model that we start tuning the parameters, and then train it. We then print the individual scores of the post-prediction model, and the graphs show that the recall is good. Still, the accuracy is low which means that we trained the model to make the wrong positive sample operation to filter out all the negative samples as much as possible. It is the result we got using the oversampled data. Next, we will use the over-sampled data for smooth.

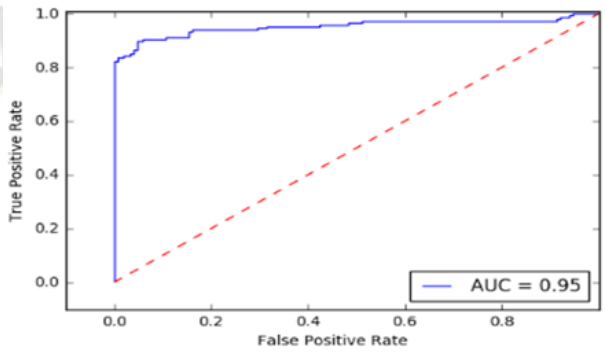

Figure 12: ROC-AUC curve

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6640
***Article Received**: 24 February 2023 **Revised**: 02 April 2023 **Accepted**: 21 April 2023*

Table 6: Indicators for classifiers (under sampling)

OverSampling	Score
Accuracy	0.98658743791
Recall	0.93698766595
Precision	0.98154796687
F1_Score	0.98778924651
ROC_AUC	0.96368741569

Test set/confusion matrix

We next have an oversampling method, the rest is the same as just now, the same process for training, and we end up with a logistic regression model training oversampling.

Table 7: Indicators for classifiers (oversampling)

Under Sampling	Score
Accuracy	0.9822398582369592
Recall	0.9269133570886076
Precision	0.9584832501705173
F1_Score	0.9183663057992376
ROC_AUC	0.9497326523286519

As you can see from the various metrics of the classifiers in Figure 13, accuracy of oversampling method has enhanced suggestively equated to the previous under sampling way. Still, at the same time, there is a slight decrease in the corresponding recall rate. But even so, we still need to make a final prediction on the test set and then draw a confusion matrix of the two for comparison.

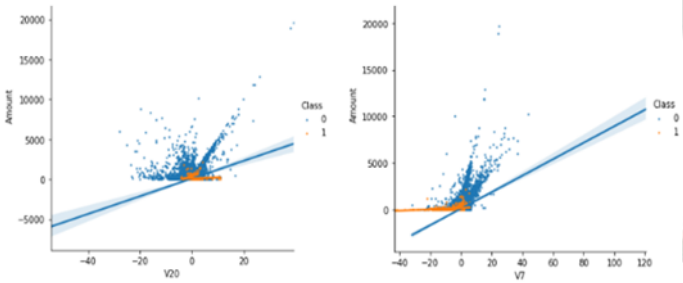

Figure 13: The learning curve of the models

Next, we performed the same operation as above using the same over-sampling method, and we came up with the results. We plotted the confusion matrix to compare the differences between the two sampling methods. As we see in the figure 13 above, we have performed up-sampled and under sampled confusion matrix calculations using our logistic regression model.

The first is the confusion matrix plot for under-sampling. As you can see, our logistic regression model is reasonably of the standard transactions are predicted to be fraudulent which makes the classification of standard samples irregular. While such a model is good enough to predict the fraudulent sample we need, whether it is commercially viable from the standpoint of our research work is open to question.

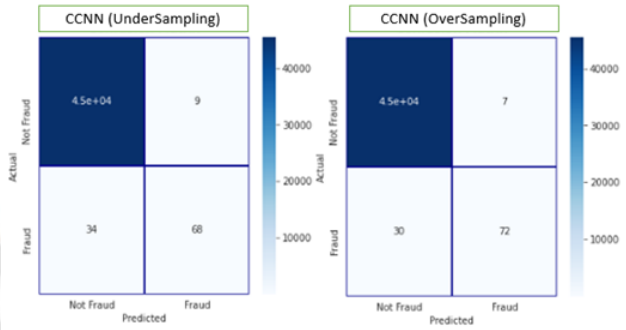

Figure 14: Logistic Regression with oversample & under sample

Since more than half of the ordinary transactions identified as fraudulent lead to failed transactions, this loss is significant, suggesting that we may need a better model. The up-sampling confusion matrix was then plotted, and the up-sampling predictions were similar to what we expected, even though there were a few unidentified frauds. Still, the number of false positives for regular trades was much lower which is probably more in line with our expectations. Conclusion, this anti-fraud training scheme, we used 'sklearn', 'imblearn', 'Keras' and other scientific computing packages. And it applied techniques such as cross-validation, grid/random search, up/down Sampling, learning curve, confusion matrix, etc., and built logistic regression, neural network, decision tree, KNN, vector machine and proposed cognitive learning models. Finally, the down-sampled logistic regression model was chosen as the final model which has a recall rate of 87% and an accuracy of 97%.

Result and Conclusion

Dataset of the bank credit card is from kaggle.com. Also, we are pre-processing and feature engineering scales and selects features, and uses the smote algorithm (under sampling and down sampling). Then we build an anti-fraud prediction model based on the five algorithms: LR, KNN, SVM, DT and experience & fuzzy (CCNN). The model can predict whether a user has made fraudulent purchases. Then we used a confusion matrix to compare the results of the two sampling methods. The best solution is logistic regression (under sampling) which is more in line with our expectations. It also achieves an accuracy of 99.00%. Then although credit card spoofing detection, most of the current research is still using decision tree and logistic regression test. But in this research work, I think two points where we added SVM and universal algorithm CCNN, to make training comparison together. I also believe meaningful results emerged. Our proposed model did not perform poorly, and also, we dealt with the

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6640
***Article Received**: 24 February 2023 **Revised**: 02 April 2023 **Accepted**: 21 April 2023*

sample imbalance problem to get significant marks. Finally, while KNN and CCNN perform well, it is also possible to get a better notation if they are trained later on for integration. Secondly, the training of SVM algorithms usually takes a long time, and if we are still increasing the amount of data, we may process the results differently.

Comparative analysis

In the present study, though, we conducted a comparative analysis. However, it is only limited to the study of these classifier algorithms and a single data set. Although the feature selection in credit card detection is similar, the collection method and method selection are different, so different research subjects will have very different results. For example, according to the study [5], KNN is the most accurate classifier algorithm, but logistic regression is the worst in comparison. It may have something to do with the process and purpose of the experiment.

Comparative analysis without cross-validation

We compared the performance results of this experiment with those of previous studies. The aim is to look for products on CCF, although different techniques are used. But our goals are the same. While improving the accuracy of detection, many factors need to be considered such as data set size, using a classifier and final evaluation method. Are essential factors in determining accuracy.

Table 9 is a comparison of the parameters of two different studies; We were able to see that the highest accuracy was 98.92%, and also using the highest number of samples. This research chooses NB, KNN and LR classifier (Awoyemi, J. O 2017).

Table 8: Comparison of the result of different studies without cross-validation

Author	Classifier	Sample Size	Accuracy
Sahin and Duman 2011 [5]	ANN and Logistics Regression	Train Value: 2723 Test Value: 1168	94.51
Shoufei Han 2020 [2]	LR, SVM, KNN, DT and CatBoost	Train Value: 688 Test Value: 295	94.0%
Awoyemi, J.O 2017 [3]	NB, KNN and LR	Train Value: 159238 Test Value: 68236	97.37%
El Barakaz Fatima 2021 [4]	NN, DT and LR	Train Value: 103587 Test Value: 2548	95.84%
Proposed research work	Cognitive Convolutional Neural Network	Training Value: 286500 Test Value: 56925	98.92%

The next largest number of datasets, again using a neural network (El Barakaz Fatima 2021), also used the original maximum number of datasets, with an accuracy of 98.92%, meaning that the high number of samples, KNN and Bayesian algorithms were optimal before the tuning cross-validation was performed.

Comparative analysis with cross-validation

When we have used cross-validation and some other manipulations in different studies, it means that our data set and accuracy will change in some way.

Table 9: Comparison of the result of different studies with cross-validation

Author	Classifier	Sample Size	Accuracy
Sahin and Duman 2011 [5]	ANN and Logistics Regression	Train Value: 160000 Test Value: 40000	94.7%
Shoufei Han 2020 [2]	LR, SVM, KNN, DT and CatBoost	Train Value: 160000 Test Value: 40000	94.82%
Awoyemi, J.O 2017 [3]	NB, KNN and LR	Train Value: 160000 Test Value: 40000	97.75%
El Barakaz Fatima 2021 [4]	NN, DT and LR	Train Value: 160000 Test Value: 40000	95.93%
Proposed research work	Cognitive Convolutional Neural Network	Train Value: 160000 Test Value: 40000	98.87%

Although cross-validation is not the most significant factor in determining classifier technology; however, as can be seen from the table, it still has an impact on the different classifiers. We see that. The highest accuracy rate is 98.87% achieved by proposed mode. Then the other studies have increased their accuracy rate accordingly, from which it is also evident that there is another point where the number of datasets is

also an essential factor in the accuracy rate. When the training dataset is increased in the first (Sahin and Duman 2011) and the second (Shoufei Han, 2020) study, there is an increase in the accuracy rate (the most massive increase from 98.22% to 98.97%), and in the subsequent third (Awoyemi, J. O 2017) and fourth (El Barakaz Fatima 2021) research, the accuracy rate has almost stabilized and remains 98.85% and 98.93%. CCNN model accuracy rate stabilized with 98.22% to 98.47%. Classification performance changes little after a

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6640
***Article Received**: 24 February 2023 **Revised**: 02 April 2023 **Accepted**: 21 April 2023*

specific size and remains at a more stable value. However, it should be reminded that in the actual experiment, when the training scale increases, although the classification performance is improved, the training time also will be doubled, the corresponding feature potential growth, the classification time will also increase. Therefore, in future experiments, classification performance and time requirements should be considered together.

VII CONCLUSION AND FUTURE WORK

This research is all about analyzing CCFD models based on different ML classification algorithms. The goal is to be in this training and testing. To find out the best way to process the dataset and the best ML classification algorithm for the dataset of this credit card transaction. So, to achieve this, we chose five different classifiers, respectively. Between them, ten different combinations of algorithms and sampling methods were used to evaluate their predicted performance as a way to get better results for CCFD. Finally, we cross-validated the technique applied to all the individual classifiers to obtain more accurate results.

We also have some findings for this study: Using oversampling to deal with a too unbalanced credit card transaction dataset in the confusion matrix ended up with the same results as we expected. Logistic regression, as one of the simpler few algorithms, still has their advantages in targeting differential data processing, followed by the SVM algorithm. There is proposed fault detection model CCNN algorithm which both perform well. We can compare to the previously mentioned literature for the model training and testing, this study obtains an optimal ML algorithm for CCFD - logistic regression (oversampling) and also achieves high accuracy results. This research work was more successful in completing the training of the CCFD model, but there are many areas for improvement in future work.

After completing the training of the optimal model, we can try to combine two or more. Classifiers with training and evaluating the detection performance. It can provide more possibilities. Use deep learning similar to neural networks. DL is having various deep functionalities from ML in that it is unsupervised learning. It uses unstructured or unlabeled data and does not require the developer to tell it what to look for in the data. It is then possible to train CCFD models in a simplified way. Although we try to use CCNN which is an excellent algorithm, due to the limit time, after adjusting the parameters, the performance can be more optimized. In the data source, as we are using someone else's original dataset possibly. At a later stage, if we then extract more data from the network. The amount of data is gradually increasing which may be useful for training. The final predictive performance of the model is also improved. In other words, the detection accuracy is enhanced by a large data set. The classifier of ML is tested for different types of attacks. And analyses its performance under attack. And then use this. Make appropriate measures to improve its security. Using the existing mature and effective classification methods, we can enhance credit card detection fraud detection performance. Then we use the current bank's credit card system to evaluate whether this model was achieved high accurate, as a way to test the real CCFD transactional data.

REFERENCES

[1] Fawaz Khaled Alarfaj, Iqra Malik, Hikmat Ullah Khan, Naif Almusallam, Muhammad., "Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms", IEEE Access, Vol 10, ISSN: 2169-3536, pp-39700 – 39715, DOI: 10.1109/ACCESS.2022.3166891, 2022.

[2] Shoufei Han, Kun Zhu, MengChu Zhou, Xinye Cai, "Information-Utilization-Method-Assisted Multimodal Multi-Objective Optimization and Application to Credit Card Fraud Detection", IEEE Transactions on Computational Social Systems, Vol 8, Issue 4, pp – 856-869, DOI: 10.1109/TCSS.2021.3061439, 2021.

[3] J. O. Awoyemi, A. O. Adetunmbi, S. A. Oluwadare, "Credit Card Fraud Detection Using Machine Learning Techniques: A Comparative Analysis", International Conference on Computing Networking and Informatics (ICCNI), pp – 1-9, doi.org/10.1109/iccni.2017.8123782, 2017.

[4] El Barakaz Fatima, Boutkhoum Omar, El Moutaouakkil Abdelmajid, Furqan Rustam, "Minimizing the Overlapping Degree to Improve Class-Imbalanced Learning Under Sparse Feature Selection: Application to Fraud Detection", IEEE Access, Vol 9, ISSN: 2169-3536, pp-28101-28110, DOI: 10.1109/ACCESS.2021.3056285, 2021.

[5] Y. Sahin, E. Duman, "Detecting credit card fraud by ANN and logistic regression", International Symposium on Innovations in Intelligent Systems and Applications, DOI: 10.1109/INISTA.2011.5946108, 2011.

[6] Jin-Yi Cai, Michael Kowalczyk, Tyson Williams, "Gadgets and Anti-Gadgets Leading to a Complexity Dichotomy", ACM Transactions on Computation Theory, Vol 11, Issue 2, pp – 1-26, https://doi.org/10.1145/3305272, 2019.

[7] Seeja, K. and Zareapoor, M. J. T. S. W. J. 2014. FraudMiner: A novel credit card fraud detection model based on frequent itemset mining. 2014.

[8] Reem M. Own, Sameh A. Salem, Amr E. Mohamed, "TCCFD: An Efficient Tree-based Framework for Credit Card Fraud Detection", 16th International Conference on Computer Engineering and Systems (ICCES), 10.1109/ICCES54031.2021.9686121, 2021.

[9] Houssem Sifaou,Abla Kammoun,Mohamed-Slim Alouini, "High-dimensional linear discriminant analysis classifier for spiked covariance model", The Journal of Machine Learning Research , Vol 21, Issue 1, pp - 4508–4531, 2020.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6640
***Article Received**: 24 February 2023 **Revised**: 02 April 2023 **Accepted**: 21 April 2023*

[10] Snehal Patil, Harshada Somavanshi, Jyoti Gaikwad, Amruta Deshmane, Rinku Badgujar, "Credit Card Fraud Detection Using Decision Tree Induction Algorithm", International Journal of Computer Science and Mobile Computing, Vol 4, Issue 4, pp – 92-95, 2015.

[11] Anjali Singh Rathore, Ankit Kumar, Depanshi Tomar, Vasudha Goyal, Kaamya Sarda, Dinesh Vij, "Credit Card Fraud Detection using Machine Learning", 2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART), DOI: 10.1109/SMART52563.2021.9676262, 2021.

[12] 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.

[13] H. Tingfei, C. Guangquan, and H. Kuihua, ''Using variational auto encoding in credit card fraud detection,'' IEEE Access, vol. 8, pp. 149841–149853, 2020. F. Wang, Z. Li, F. He, R. Wang, W. Yu, and F. Nie, ''Feature learning viewpoint of AdaBoost and a new algorithm,'' IEEE Access, vol. 7, pp. 149890–149899, 2019

[14] Emmanuel Ileberi, Yanxia Sun, Zenghui Wang, "Performance Evaluation of Machine Learning Methods for Credit Card Fraud Detection Using SMOTE and AdaBoost", IEEE Access, Vol 9, ISSN: 2169-3536, pp - 165286 – 165294, DOI: 10.1109/ACCESS.2021.3134330. 2021.

[15] J. Feng, H. Xu, S. Mannor, and S. Yan, ''Robust logistic regression a classification,'' in Proc. Adv. Neural Inf. Process. Syst., vol. 27, pp. = 253 – 2014.

[16] K. Kirasich, T. Smith, and B. Sadler, ''Random Forest vs logistic regression: Binary classification for heterogeneous datasets,'' SMU Data Sci. Rev., vol. 1, no. 3, p. 9, 2018.

[17] Alghamdi, M. et al. 2017. Predicting diabetes mellitus using SMOTE and ensemble machine learning approach: The Henry Ford Exercise Testing (FIT) project. 12(7), p. e0179805, 2017.

[18] Han, H. et al. eds. 2005. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. International conference on intelligent computing. Springer, 2005.

[19] T. T. Wong and P. Y. Yeh, ''Reliable accuracy estimates from k-fold cross validation,'' IEEE Trans. Knowl. Data Eng., vol. 32, no. 8, pp. 1586–1594, Apr. 2019.

[20] D. Elreedy and A. F. Atiya, ''A comprehensive analysis of synthetic minority oversampling technique (SMOTE) for handling class imbalance,''Inf. Sci., vol. 505, pp. 32–64, Dec. 2019

[21] M. C. M. Oo and T. Thein, ''An efficient predictive analytics system for high dimensional big data,'' J. King Saud Univ. Comput. Inf. Sci., vol. 34, no. 1, pp. 1521–1532, Jan. 2022.

[22] S. N. Kalid, K.-H. Ng, G.-K. Tong, and K.-C. Khor, ''A multiple classifiers system for anomaly detection in credit card data with unbalanced andoverlapped classes,'' IEEE Access, vol. 8, pp. 28210–28221, 2020.

[23] I. D. Mienye and Y. Sun, ''Performance analysis of cost-sensitive learning methods with application to imbalanced medical data,'' Informat. Med. Unlocked, vol. 25, Art. no. 100690, 2021.

[24] B. Wiese and C. Omlin, ''Credit card transactions, fraud detection, and machine learning: Modelling time with LSTM recurrent neural networks,'' in Innovations in Neural Information Paradigms and Applications. Springer, pp. 231–268, 2009.

[25] F. Wang, Z. Li, F. He, R. Wang, W. Yu, and F. Nie, ''Feature learning viewpoint of AdaBoost and a new algorithm,'' IEEE Access, vol. 7, pp. 149890–149899, 2019.

[26] S. P. Maniraj, A. Saini, S. Ahmed, and S. Sarkar, ''Credit card fraud detection using machine learning and data science,'' Int. J. Res. Appl. Sci. Eng. Technol., vol. 8, no. 9, pp. 3788–3792, Jul. 2021.

[27] S. Khatri, A. Arora, and A. P. Agrawal, ''Supervised machine learning algorithms for credit card fraud detection: A comparison,'' in Proc.10th Int. Conf Cloud Comput., Data Sci. Eng. (Confluence), pp. 680–683, jan 2020.

[28] T. Hengl, M. Nussbaum, M. N. Wright, G. B. M. Heuvelink, and B. Gräler, ''Random Forest as a generic framework for predictive modeling of spatial and spatio-temporal variables,'' PeerJ, vol. 6, p. e5518, Aug. 2018.

[29] N. F. Ryman-Tubb, P. Krause, and W. Garn, ''How artificial intelligence and machine learning research impacts payment card fraud detection: A survey and industry benchmark,'' Eng. Appl. Artif. Intell., vol. 76, pp. 130–157, doi: 10.1016/j.engappai.2018.07.008, 2018.

[30] S. S. Lad, A. C. Adamuthe, ''Malware classification with improved convolutional neural network model,'' Int. J. Comput. Netw. Inf. Secur., vol. 12, no. 6, pp. 30–43, doi: 10.5815/ijcnis.2020.06.03, 2021.