ISSN: 2584-0495 Vol. 2, Issue 9, pp. 1211-1217

International Journal of Microsystems and IoT

ISSN: (Online) Journal homepage: https://www.ijmit.org

Fraud Detection Using Machine Learning and Community Detection Algorithm

Ashish Ranjan, Sateesh Kumar Awasthi

Cite as: Ranjan, A., & Awasthi, S. K. (2024). Fraud Detection Using Machine Learning and Community Detection Algorithm. International Journal of Microsystems and Iot, 2(9), 1211-1217. https://doi.org/10.5281/zenodo.14102397

Published online: 23 Sept 2024

DOI: https://doi.org/10.5281/zenodo.14102397

International Journal of Microsystems and IoT
Vol. 2, Issue 9, pp.1211-1217; DOI: https://doi.org/10.5281/zenodo.14102397

Fraud Detection Using Machine Learning and Community Detection Algorithm

Ashish Ranjan, Sateesh Kumar Awasthi

Department of Electronics and Communication Engineering, Dr. B.R. Ambedkar National Institute of Technology, Jalandhar, India

ABSTRACT
Online fraud has indeed been a growing problem in recent years, causing significant financial losses for individuals, merchants, and banks. Machine learning has demonstrated its effectiveness in detecting and mitigating credit card fraud. This paper aims to review various online credit card techniques for fraud detection utilizing Machine Learning algorithms and evaluate them based on performance metrics such as precision, accuracy, and specificity. Additionally, the paper proposesa Fraud Detection System (FDS) that employs a supervised Random Forest algorithm to enhance fraud detection accuracy. The suggested system employs a learning-to-rank strategy to efficiently prioritize alerts and tackles the issue of concept driftin fraud detection. This ensures that the system stays efficientand current in identifying changing fraud patterns.

KEYWORDS
Support Vector Machine (SVM); Random Forest (RF); Fraud Detection System (FDS); Graphical Neural Networks (GNN); K- Nearest Neighbor (KNN)

1. INTRODUCTION

A significant obstacle linked with credit card fraud involves the misuse of credit cards and other payment cards as unau-theorized sources of funding for transactions. One unlawful method of obtaining things and money is fraud. Obtaining goods without paying for them or withdrawing funds from an account without authorization could be the aim of such illicit transactions. Finding this kind of fraud is difficult and could put businesses. In general, an investigator is not able to review every transaction in the real-world FDS [1].

In the above-mentioned scenario, the FDS monitors all authorized transactions and identifies the most suspicious ones for further investigation. The investigator confirms these alerts and advises FDS as to whether the transaction was fraudulent or approved. It takes [25] time and money to verify every warning on a daily basis. Therefore, the investigator can only authenticate a limited number of notifications daily. Until the customer finds the leftover transactions and reports them as fraudulent, they are left unchecked. Furthermore, fraudster's methods and cardholder's spending patterns evolve with time. Concept drift is the term used to describe this shift in credit card transactions [1], [6]. As a result, identifying credit card fraud is typically challenging. Machine Learning is widely recognized as one of the most effective approaches for identifying fraudulent activity. It detects credit card theft by using a regression and classification approach. Basically, the Machine learning algorithms are classified into supervised [7], [9], and unsupervised [8] learning methods. Supervised learning methods rely on labeled transactions to train the classifier, while unsupervised learning algorithms utilize peer group analysis, categorizing consumers according to their profiles and identifying fraud through their spending behaviors.

Numerous learning techniques, such as neural networks [7], Logistic Regression, decision trees [3], Naive Bayes [5], Support Vector Machine (SVM) [4], K-Nearest Neighbor (KNN) [5], and Random Forest (RF) [1], [2], are being presented for credit card fraud detection. This study evaluates the effectiveness of the aforementioned algorithms based on their capacity to determine if a transaction was fraudulent or approved. Accuracy, precision, and specificity of performance measures are used in the comparison. The outcome shows that the Random Forest algorithm out-compete other methods in terms of accuracy, specificity, and precision.

2. Related Work

Mittal and Tyagi [10] investigated several supervised and unsupervised machines learning models, including Logistic Regression, KNN, Naive Bayes, SVM, Gradient Boosted Trees, RF, and Artificial Neural Networks (ANN), to predict credit card fraud. Specifically, they addressed the problem of class imbalance in the dataset to guarantee a fair comparison of model performance.
Another fraud detection approach involves social net- work analysis techniques like community detection. *Soltani, Nguyen, Yang, Faghani and Yagoub*. [11] used weighted and directed network modeling to identify money launder-ing groups among customers. Similarly, *Ma, Zhang, Wang and Pozdnoukhov*. [12] developed a graph-based system to detect risky customers, leveraging transaction data to construct account-link graphs and identify suspicious communities.
Zhdanova, Repp, Rieke, Gaber and Hemery. [17] conducted an examination of fraudulent transactions within the mobile money framework, utilizing a synthetic database generated by a simulator. They used random forest algorithms, C4.5 decision trees, and PART decision tables to create

prediction models. This model's performance was assessed using criteria like recall scores, precision, and confusion matrix. In the preliminary assessments, the maximum recall scores and accuracy attained were 36.65% and 96.05%, respectively. The Subsequent focus on money laundering activities with a chain length exceeding three criteria (recall scores, precision, and confusion matrix) led to notable improvements, with accuracy and recall scores rising to 99.8% and 90.1%, respectively.

Integrating Machine Learning models with social network analysis enhances fraud detection accuracy. This integrated approach enhances understanding of individual behaviors and network structures, leading to better identification of fraudulent activities.

3. Preliminaries

Another data mining technique called classification uses categories to help with more precise analysis and forecasts. It involves forecasting a specific outcome based on provided inputs [13]. Several algorithms are available for identifying whether a transaction is fraudulent or not.[21] We have examineda selection of these algorithms - Logistic regression predicts fraud probability based on past transactions. KNN classifies new transactions based on their closest past neighbors (fraud or not). Decision trees create a series of questions about a transaction to predict if it's fraud. All can be used for fraud detection, but the best choice depends on your data and needs,K-Nearest Neighbors (KNN) is a fraud detection algorithm that compares a new transaction to the closest past transactions in the data. Imagine voting among your closest neighbors - KNN does the same for transactions. It assigns a "fraud" or "not fraud" label based on the majority vote of those closest neighbors. It's easy to understand but requires choosing the right number of neighbors (k) and can be slow with massive datasets, and Decision trees [16] fight fraud like a choose- your-own-adventure story. It asks a series of questions about a transaction (amount, location, etc.). Each answer leads down a branch, ultimately reaching a "fraud" or "legit" conclusion. Easy to understand, it works with various data types, but can be too specific and require adjustments to avoid being fooled by slight changes.

1.1 Logistic Regression

Logistic regression, a prevalent form of regression analysis, is employed to predict the likelihood of a binary event (whether it occurs or not) based on various independent variables. This method is highly effective when the outcomeis categorical and presents two potential results, typically encoded as 0 and 1.

The sigmoid function, sometimes referred to as the logistic function, is used by the logistic model, to convert the log odds (logit) of the event happening into probabilities ranging from 0 to 1. The logistic function can be defined as:

$$P(Y = 1|X) = 1/(1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n)})$$

Here:

- $P(Y = 1 | X)$ is the likelihood that the event will occur depending on each of the independent variable's values.$X_1, X_2, \ldots, X_n$.[19]
- $\beta_0, \beta_1, \ldots, \beta_n$ are the parameters to be estimated through the logistic regression.
- $X_1, X_2, \ldots, X_n$ are the independent random variables.

In logistic regression, the objective is to estimate the coefficients $\beta_0, \beta_1, \ldots, \beta_n$ in a way that maximizes the likelihood of the observed data, often employing techniques such as maximum likelihood estimation.

This approach allows us to predict the probability of the event happening i.e., success or failure based on the given values of the independent variables, making logistic regression a valuable tool in various fields such as medicines, finance, and marketing.

1.2 K Nearest neighbor

A versatile and widely used technique is the k-nearest neighbor (KNN) algorithm applied supervised machine learning technique, frequently employed for both classification and regression purposes. Its popularity is notably high in classification assignments. In the KNN algorithm, during the classification of a new data point also called a testing query, the system searches for the k closest data points known as neighbors within the training dataset, determined by a selecteddistance metric such as Euclidean distance. Subsequently, the classes of these neighbors are employed in predicting the class of the testing query through either majority voting (for classification) or averaging (for regression). The algorithm's simplicity and flexibility make it a powerful tool in various applications [14].

1.3 Decision Tree

A traditional tree consists of a root, branches, and leaves. In the context of Decision Trees, every part of this structure, including the leaf, branch, and root nodes, is duplicated. During evaluation, every internal node in the tree represents a test on a particular attribute. The outcome of the test guides the traversal down the branches, culminating in a class label assigned at the leaf node.[18] The apex of the tree is the rootnode, acting as the parent to all other nodes. Regarding the attributes of Decision Trees, the ID3 algorithm is exclusively simulated using the WEKA tool, and the dataset is solely categorical. For simulation, ID3 cannot handle continuous datasets. Similarly, both CART and C4.5 share the same attributes as ID3, differing only in their capability to accept continuous datasets for simulation purposes.

1.4 Community Detection

Community detection is a foundational task in network analysis, focusing on the identification of subgroups or communities among nodes within a network, where nodes share more connections within their group than with nodes outside the group. Recommendation systems, biological network analysis, social network analysis, and FDS implementation are just a few of its many uses.[20]

1.5 GNN

Graphical Neural Networks (GNNs) are adept at handling graph data, where nodes denote entities and edges signify relationships. Utilizing message passing, nodes iteratively share information, refining their representations based on local insights. GNNs derive node embeddings capturing attributes and graph structure, essential for tasks like node classification. They enable graph-level operations such as pooling for holistic graph representations. Common architectures include Graph Convolutional Networks (GCNs) and Graph Attention Networks (GATs), applied [24] across social networks, bioinformatics, and recommendation systems.

4. Proposed Methods

1.1 Using ML in Online Fraud Detection

In this research work, we utilize Anaconda Notebooks, a hosted JupyterLab service that enables us to reliably run JupyterLab notebooks online. We import our data from an online platform i.e. Kaggle. Here the primary goals of our proposed system are:

1. Training the model by incorporating feedback and delayedsamples, and combining their likelihood to detect alerts.
2. Using machine learning methods to address problems withidea drift and class imbalance.
3. Constructing a learning-to-rank methodology to improvealert precision.
4. Introducing presentation metrics relevant to real-world Fraud Detection Systems (FDS).

1.2 Identifying fraudulent groups using graph analysis

In the other approach, transactions can be depicted as a graph, with users as nodes and interactions as edges. Conventional algorithms such as XGBoost and DLRM typically analyze nodes or edges independently. In contrast, graph- based approaches take into account the local context and structure, which includes neighboring nodes and edges. In the conventional graph domain, statistical methods aggregate features from adjacent nodes or edges. Techniques such as the Louvain method identify user communities. Yet, they may lack the expressivity to fully comprehend the original graph. Graph neural networks (GNNs) address this by incorporating local structural and feature contexts within the model. Through aggregation and message passing, information is propagated to neighboring nodes. Multiple[22] graph convolution layers allow nodes to gather information from distant nodes, expandingthe model's receptive field. Graph Neural Networks (GNNs) effectively manage complex transaction sequences utilized by fraudsters to obscure fraudulent activities in fraud detection tasks. They can adapt to changing patterns through iterative model retraining.

Here, the data set contains:

- 24 million distinct transactions
- 4,999 distinct merchants
- 9,999 distinct cards
- 29,999 fraudulent samples (0.1% of total transactions).

Table.1 Description of A Dataset

S.no.	Field name	Description
1	User	A user's transaction identifier
2	Source customer id's	Transaction source account ID
3	MCC	Merchant Category Code
4	Transaction type	The transaction type
5	Amount	Transaction amount
6	Use Chip	Chip usage in transaction
7	Merchant name	Merchant name used in the transaction
8	Merchant City	Merchant location(city)
9	Merchant State	Merchant location(state)
10	Is Fraud?	Fraud detection status

A. Implementation

1) Machine Learning workflow with various Algorithms:

<u>Algorithm For Random Forest</u>

```
1)  procedure
2)  Load $n$ records ($n = 1296675$).
3)  Convert column(K) to boolean ($k = 23$).
4)  Remove rows containing missing values from the dataframe.
5)  Extract relevant features.
6)  while $x =$ fraud Train_df and $y =$ fraud Train df ['is fraud'] do
7)      Remove 'is fraud' from matrix $X$.
8)  end while
9)  if $cc\ num = 4767265376804500$
10)     return true: fraudulent transaction
11) else
12)     return false: non-fraudulent transaction
13) end if
14) Apply random forest.
15) while $n\ estimate = 100$ and $random\ state(R) = 42$ do
16)     Divide the training and testing sets (with 80% for training and 20% for testing) of the dataset $x$ and the target variable $y$, where $R = 42$.
17)     Predict $x\_\ test$ using trained RF model.
18)     Compute:
19)         Confusion matrix, Accuracy, precision, specificity.
20) end while
```

```
21) end procedure
```

Algorithm for KNN

```
1)  procedure
2)  Load $n$ records ($n$ = 1296675)
3)  Convert column to boolean ($k$ = 23).
4)  Remove rows containing missing values from the dataframe.
5)  Extract relevant features.
6)  while $x$ = fraud Train df and $y$ = fraud Train_ df['is fraud'] do
7)      Remove 'is fraud' from matrix $X$.
8)  end while
9)  if $cc\ num$ = 4767265376804500
10)     return true: fraudulent transaction
11) else
12)     return false: non-fraudulent transaction
13) end if
14) Apply KNN.
15) while $n\_\ neighbour$ = 20 do
16)     Train the model
17)     Predict $x\_\ test$ using trained KNN model.
18)     Compute:
19)         Confusion matrix, Accuracy, precision, specificity.
20) end while
21) end procedure
```

Algorithm for Logistic Regression

```
1)  procedure
2)  Load $n$ records ($n$ = 1296675).
3)  Convert column to boolean ($k$ = 23).
4)  Remove rows containing missing values from the dataframe.
5)  Extract relevant features.
6)  while $x$ = fraud Train_ df and $y$ = fraud Train df['is fraud'] do
7)      Remove 'is fraud' from matrix $X$.
8)  end while
9)  if $cc\_\ num$ = 4767265376804500
10)     return true: fraudulent transaction
11) else
12)     return false: non-fraudulent transaction
13) end if
14) Apply Logistic regression
15) while $max\_\ iter$ = 1000 do
16)     Train a logistic regression classifier by employing the fit() function with the training dataset (X_ train) along with their respective labels (y_ train).
17)     Predict $x\_\ test$ using trained Logistic model.
18)     Compute:
19)         Confusion matrix, Accuracy, precision, specificity.
20) end while
21) end procedure
```

Algorithm for Decision Tree

```
1)  procedure
2)  Load $n$ records ($n$ = 1296675).
3)  Convert column to boolean ($k$ = 23).
4)  Remove rows containing missing values from the dataframe.
5)  Extract relevant features.
6)  while $x$ = fraud Train_ df and $y$ = fraud Train df['is fraud'] do
7)      Remove 'is fraud' from matrix $X$.
8)  end while
9)  if $cc\ num$ = 4767265376804500
10)     return true: fraudulent transaction
11) else
12)     return false: non-fraudulent transaction
13) end if
14) Apply Decision Tree
15) while random state(R) = 42 do
16)     Train our decision tree model.
17)     Predict $x\ test$ using trained Decision Tree model.
18)     Compute:
19)         Confusion matrix, Accuracy, precision, specificity.
20) end while
21) end procedure
```

2) Steps for GNN: Fraud detection can be enhanced by creating a multigraph of transactions and applying a Graph.

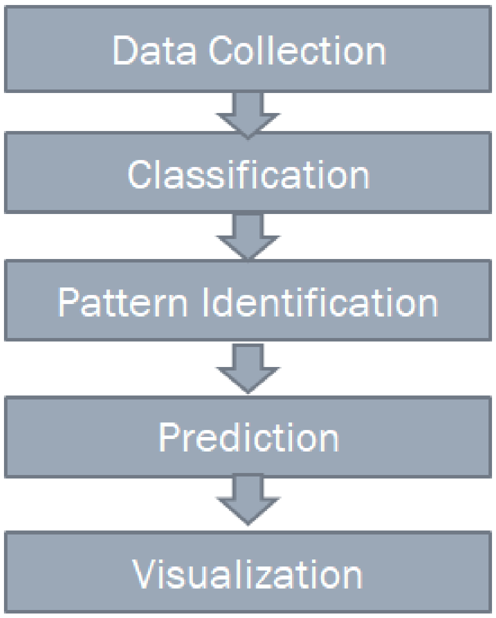


Fig. 1 Shows the process.

Neural Network to its edge list, a process outlined in thesesteps:

1. Organize the transaction data: Gather and structure the transaction information in a way that allows for the creation of edges in the multigraph. One approach isto represent each transaction as a tuple (node1, node2, attributes), where node1 and node2 denote the sender and recipient, and attributes encompass details like the amount, timestamp, and transaction type stored in a dictionary.
2. Construct the multigraph: Utilize the transaction data to form a multigraph employing the NetworkX library. Employ the add edge () method to introduce edges to the multigraph, where each edge corresponds to a specific transaction. Extract the edges list and their features: Use the edges () method of the multigraph to extract the edges list and their features, which will be used as input to the GNN.
3. Retrieve the list of edges and their attributes: Employ the edges () method of the multigraph to extract both the list of edges and their associated features, serving as input for the Graph Neural Network (GNN).
4. Implement a GNN on the edge list: To utilize a Graph Neural Network (GNN) on the edge list, one can employ a Graph Neural Network library like PyTorch Geometric, Deep Graph Library (DGL), or Spektral. With the use of the learned features, the GNN will obtain representations of the multigraph edges, allowing for the categorization of those edges as fraudulent or not.

Evaluation: To measure the efficiency of the Graph Neural Network (GNN), you can divide the data into training andtesting portions. Following that, the testing subset canbe employed to assess the model's accuracy, precision, recall, and F1-score.

5. Results and Discussion / Comparison

1.1 Comparative examination of Machine learning algorithmseffectiveness in online fraud detection.

Here are the results for learning algorithms utilized in online fraud detection, compared in a table. Their specificity, accuracy, and precision are the main points of comparison [15]. The table perfectly shows that Random Forest accuracy surpasses that of all other algorithms, on the parameters of accuracy, precision, and specificity. Consequently, the suggested system utilizing the Random Forest algorithm is anticipated to demonstrate improved accuracy when applied to a larger volume of training data. The evaluation of different machine learning methods heavily depends on confusion matrix parameters. These parameters measure various outcomes like true positives, true negatives, false positives, and false negatives within a single confusion matrix. Subsequently, essential evaluation metrics such as accuracy, recall, and precision are calculated based on these values.

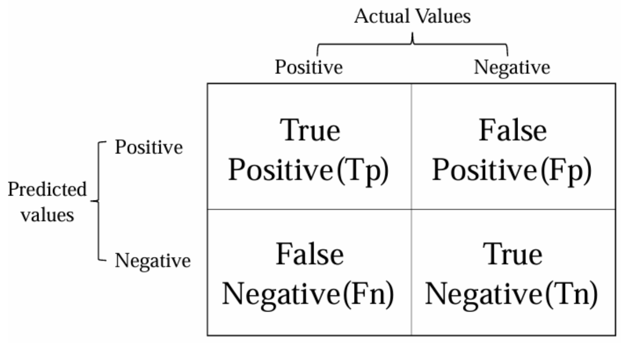


Fig. 2 Confusion matrix.

$$Accuracy = \frac{Tp + Tn}{Tp + Tn + Fp + Fn} \quad (1)$$

$$Precision = \frac{Tp}{Tp + Fp} \quad (2)$$

$$Specificity = \frac{Tn}{Tn + Fn} \quad (3)$$

Table. 2 Comparison of Techniques for Machine Leaning

Categories	Accuracy	Precision	Specificity
Random forest	0.995	0.932	0.968
Logical Regression	0.993	0.915	0.968
KNN	0.932	0.450	0.961
Decision Tree	0.908	0.911	0.912

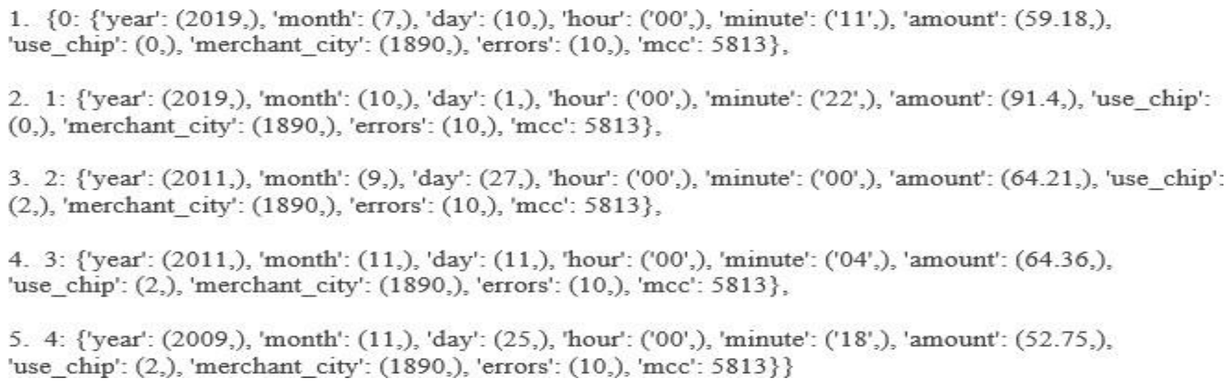
1.2 Detection Epoch and Losses in Fraud Network using GNN

Overall, the main result is the trained GNN model capable of detecting fraud in credit card transactions based on the transaction graph structure and properties. The model's effectiveness can be further evaluated using additional metrics and testing datasets. Certainly! Let's dive into the results related to epoch and loss during the training process:

1. Epoch: In this section, the training loop iterates over a fixed number of epochs (num epochs), with each epoch comprising multiple iterations or batches. During each epoch, the model is exposed to the entire dataset, allowing it to learn fromthe data multiple times. With each pass through the dataset, the model updates its parameters based on the observed patterns, aiming to enhance its ability to make accurate predictions. By monitoring the training progress across epochs, We learn more about the way the model performs over time. This ongoing assessment helps us determine whether the model is converging towards a satisfactory solution, where its predictive capabilities effectively capture the underlying patterns in the data. The Epoch" column in the provided output indicates thenumber of training epochs. In the output, the epochs range from 0 to 200, with increments of 20, corresponding to the iterations through the entire dataset during training.
2. Loss: In this code, the binary cross-entropy loss (BCE-WithLogitsLoss) is employed as a measure of the model's performance [23] on the training data. This loss function is widely utilized in binary classification tasks, such as fraud

detection, where the objective is to distinguish between two classes (fraudulent and non-fraudulent transactions). The loss

value i



```
1. {0: {'year': (2019,), 'month': (7,), 'day': (10,), 'hour': ('00',), 'minute': ('11',), 'amount': (59.18,),
'use_chip': (0,), 'merchant_city': (1890,), 'errors': (10,), 'mcc': 5813},

2. 1: {'year': (2019,), 'month': (10,), 'day': (1,), 'hour': ('00',), 'minute': ('22',), 'amount': (91.4,), 'use_chip':
(0,), 'merchant_city': (1890,), 'errors': (10,), 'mcc': 5813},

3. 2: {'year': (2011,), 'month': (9,), 'day': (27,), 'hour': ('00',), 'minute': ('00',), 'amount': (64.21,), 'use_chip':
(2,), 'merchant_city': (1890,), 'errors': (10,), 'mcc': 5813},

4. 3: {'year': (2011,), 'month': (11,), 'day': (11,), 'hour': ('00',), 'minute': ('04',), 'amount': (64.36,),
'use_chip': (2,), 'merchant_city': (1890,), 'errors': (10,), 'mcc': 5813},

5. 4: {'year': (2009,), 'month': (11,), 'day': (25,), 'hour': ('00',), 'minute': ('18',), 'amount': (52.75,),
'use_chip': (2,), 'merchant_city': (1890,), 'errors': (10,), 'mcc': 5813}}
```

Fig. 3 We retrieve the properties of a small sample of edges in our graph (G) and print their properties.

calculated for the entire dataset or batches of data and utilized to adjust the model's parameters via backpropagation. A diminishing loss across epochs indicates that the model is learning from the data and enhancing its ability to discriminate between fraudulent and non-fraudulent transactions. By monitoring the loss, we can evaluate the efficiency of the model's training process and detect any issues such as overfitting or underfit- ting, thereby facilitating the optimization of the model's performance. In summary, tracking the loss over epochs allows us to understand how the model is learning from the training data and whether it is converging towards optimal performance. The "Loss" column in the provided output displays the value of the loss function calculated at each epoch. The loss value represents the discrepancy between the model predicted probabilities and the actual labels in the training data. Lower loss values indicate better agreement between predictions and actual labels. By observing the loss values across epochs, we can track how the loss decreases over time. Ideally, we would expect the lossto decrease gradually as the model learns from the data and improves its predictive capabilities. In the provided output, theloss decreases from an initial value of 75.22 to a final value of 0.07 over the course of 200 epochs, indicating that the modelis learning effectively and converging toward a satisfactory solution.

Table.3 Shows the Epochs and Losses

Sno.	Epochs	Losses
1	Epoch: 0	Loss: 0.1819741725921630
2	Epoch: 20	Loss: 1.0938960313796997
3	Epoch: 40	Loss: 1.1777929067611694
4	Epoch: 60	Loss: 1.0725549459457397
5	Epoch: 80	Loss: 0.7495688199996948
6	Epoch: 100	Loss: 0.5570971965789795
7	Epoch: 120	Loss: 0.5782715678215027
8	Epoch: 140	Loss: 0.9718276858329773
9	Epoch: 160	Loss: 0.8122223615646362
10	Epoch: 180	Loss: 0.5796378254890442
11	Epoch: 200	Loss: 0.3332529366016388

So, after finding Epoch and losses the model's outputs are converted to binary labels by applying the sigmoid function and rounding the results. After that, the model's accuracy is determined by comparing the predicted and true labels, and the result is printed.

So, we get the Accuracy: 0.99883 which is better than above mentioned Machine learning Algorithms.

6. Conclusion

This paper investigates two research domains, in the Machine Learning segment, our focus lies on comparing algorithms, and evaluating their performance in terms of accuracy, precision, and specificity. Utilizing a confusion matrix, we meticulously analyze these parameters. Remarkably, our findings underscore the superiority of Random Forest over alternative algorithms across these metrics. Furthermore, our exploration extends to the application of Random Forest on extensive datasets. We observe that leveraging Random Forest on a large volume of data contributes significantlyto enhancing accuracy, offering a more refined and precise outcome. This highlights the efficacy of Random Forest in handling substantial datasets and underscores its potential for improving predictive performance in various applications. In the second approach, the training process demonstrates effective convergence as indicated by the gradual decrease in the binary cross-entropy loss from 75.22 to 0.07 over 200 epochs. This suggests that the Graph Neural Network (GNN) modelis successfully learning from the transaction graph's structure and properties to identify fraud in credit card transactions.The consistent decrease in loss signifies improved model performance and alignment with actual labels, validating its efficiency in capturing fraudulent patterns. Further evaluation with additional metrics and testing datasets would providecomprehensive validation of the model's effectiveness in real-world fraud detection scenarios. So, at last we conclude that on the basis of Accuracy GNN model is comparatively better than other algorithms.

REFERENCES

[1] Jalinus. N Nabavi R A and Mardin A (2017). The seven steps of project- based learning model to enhance productive competences of vocational students. International Conference on Technology and Vocational Teachers (ICTVT). 251–256

[2] Pozzolo A D, Boracchi G, Caelen O, Alippi C and Bontempi G 2018 Credit card fraud detection: A realistic modeling and a novel learning strategy. IEEE Transactions on Neural Networks and Learning Systems.29(8): 3784–3797

[3] Sahin Y and Duman E 2011 Detecting credit card fraud by ANN and logistic regression. International symposium on innovations in intelligentsystems and applications. 315–319

[4] Sahin Y, Bulkan S and Duman E 2013 A cost-sensitive decision tree approach for fraud detection. Expert Systems with Applications. 40(15): 5916–5923

[5] Sahin Y and Duman E 2011 Detecting credit card fraud by decision tree and support vector machines. International MultiConference of Engineers and Computer Scientists. 1(1): 442-447

[6] Kiran S, Guru J, Kumar R, Kumar N, Katariya D and Sharma M 2018 Credit card fraud detection using naive bayes model based andknn classifier. International Journal of Advance Research, Ideas and Innovations in Technology. 4(3): 44

[7] Wang S, Minku L L and Yao X 2015 Resampling-based ensemble methods for online class imbalance learning. IEEE Transactions on Knowledge and Data Engineering. 27(5): 1356–1368

[8] Carminati M, Caron R, Maggi F and Zanero S 2015 Banksealer: A deci- sion support system for online banking fraud analysis and investigation. computers and security. 53: 175–186

[9] Elwell R and Polikar R 2011 Incremental learning of concept drift in nonstationary environments. IEEE Transactions on Neural Networks. 22(10): 1517–1531

[10] Mittal S and Tyagi S 2019 Performance evaluation of machine learning algorithms for credit card fraud detection. 9th International Conference on Cloud Computing, Data Science and Engineering. 320–324

[11] Soltani R, Nguyen U T, Yang Y and Faghani M R 2016 A new algorithm for money laundering detection based on structural similarity, in 2016. IEEE 7th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON). 1-7

[12] Ma J, Zhang D, Wang Y, Zhang Y and Pozdnoukhov A 2018 GraphRAD: A graph- based risky account detection system. KDD MLG conference. 9: 1–9

[13] Llaha O 2020 Crime analysis and prediction using Machine learning. 43th International convention on information, communication and elec- tronic technology (MIPRO). 496–501

[14] Sam U, Moses G and Olajide T 2023 Credit card fraud detection using machine learning algorithms.

[15] Patel H and Prajapati P 2018 Study and analysis of decision tree based classification algorithms. International Journal of Computer Sciences and Engineering. 6(10): 74–78

[16] Yeruva S, Harshitha M S, Kavya M, Deepa M S and Sahithi S 2023 Credit card fraud detection using machine learning. International Journal of Engineering and Advanced Technology. 12: 25–30.

[17] Zhdanova M, Repp, J, Rieke R and Gaber C 2014 No smurfs: Revealing fraud chains in mobile money transfers. International Conference on Availability, Reliability and Security, ARESstudents. International Conference on Technology and Vocational Teachers (ICTVT). 251–256

[18] Y. Thakur, B. Raj, and S. S. Gill, "Design and performance analysis of pentacene organic field effect transistor with high-Kdielectric materials." Optoelectronics and Advanced Materials - Rapid Communications, 17, 335 (2023).

[19] Thakur Y, Raj B, Raj B. Design of Pentacene Thin-Film Transistor Based Hydrogen Gas Sensor with High-K Dielectric Materials for High Sensitivity. ECS Journal of Solid State Science and Technology 2024;13:047005. https://doi.org/10.1149/2162-8777/ad3d86.

[20] D. Sarma, W. Alam, I. Saha, M. N. Alam, M. J. Alam, and S. Hossain, "Bank Fraud Detection using Community Detection Algorithm," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Jul. 2020, doi: 10.1109/icirca48905.2020.9182954.

[21] Roja, G., Kakarla, A., & Jacob, T. P. (2022, April). Cyber Patrolling using Machine Learning. In 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS) (pp. 75-78). IEEE.

[22] Ahmed, Bashar Qasem Hezam. Coding Standards Violations Impact on Software Faults. MS thesis. King Fahd University of Petroleum and Minerals (Saudi Arabia), 2014.

[23] Ndama, Oussama, and El Mokhtar En-Naimi. "Optimisation and Resampling methods for Handling Imbalanced Datasets in Credit Card Fraud Detection using Artificial Neural Networks." Proceedings of the 6th International Conference on Networking, Intelligent Systems & Security. 2023.

[24] M. Albashrawi, "Detecting Financial Fraud Using Data Mining Techniques: A Decade Review from 2004 to 2015," Journal of Data Science, vol. 14, no. 3, pp. 553 570, Mar. 2021, doi: 10.6339/jds.201607_14(3).0010.

[25] Dornadula, Vaishnavi Nath, and Sa Geetha. "Credit card fraud detection using machine learning algorithms." *Procedia computer science* 165 (2019): 631-641.

AUTHORS

Sateesh Kumar Awasthi received his B.E. degree from G. B. Pant Engineering College, Pauri (Garhwal), M.Tech degree from UPTU Lucknow in VLSI domain and Ph.d from IIT Kanpur in SPCOM. He is currently Assistant professor in Electronics and Communication Engineering at National Institute of Technology, Jalandhar, India. His research interests are Peer-to-Peer Networks, Complex Networks, Wireless Sensor Networks, Social Networks, Application of Linear Algebra and Game Theory in Networks, Statistical Signal Processing, Convex Optimization.

E-mail: awasthisk@nitj.ac.in

Ashish Ranjan received his B.tech degree from JSS Academy of Technical Education, Noida in (2021) and pursuing M.tech from National Institute of Technology Jalandhar, India in Electronics and Communication branch. His research area interest is Machine Learning, Community detection, Verilog. He also has completed training from Asic guru under guidance of Puneet Kumar (Sr. Director Synopsis) in Design verification domain.

E-mail: ashishranjan5659@gmail.com