International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Detecting Fraudulent Patterns: Real-Time Identification using Machine Learning

[1]Dr. Manoj Tarambale, [2]Dr. Ketaki Naik, [3]Rahul Manohar Patil, [4]Rajendra V. Patil, [5]Dr. Shailesh Shivaji Deore, [6]Dr. Mahua Bhowmik

Submitted: 05/12/2023 **Revised**: 15/01/2024 **Accepted**: 28/01/2024

Abstract: The difficulty of identifying fraudulent activity in real-time has grown in importance in the age of digital transactions and networked technologies. The comprehensive strategy presented in this work uses the strength of machine learning techniques to address this pressing problem.Our study focuses on creating and implementing a reliable, real-time fraud detection system that can change with changing fraud patterns while maintaining a high degree of accuracy. To analyse huge amounts of transactional data in real-time, we suggest a system that combines multiple machine learning approaches, such as K Nearest Neighbour, Logistic regression, Naive Bayes model.Our system's capacity for constant learning and adaptation is at its core. Anomaly detection methods are used to find out-of-the-ordinary trends in transaction data, and historical data is used to train prediction models that can predict fraudulent behaviour. In order to identify anomalies at the individual level, the system also uses user behaviour analysis, which improves accuracy and lowers false positives.The proposed machine learning method is highly accurate and quick at detecting fraudulent activity, making it appropriate for use healthcare domain. Our system offers a strong defence against the constantly changing terrain of fraudulent activities by upgrading its knowledge base and reacting to new fraud trends, protecting both businesses and customers.

Keywords: *Machine Learning, fraud detection, Healthcare, KNN, LR, NB*

1. Introduction

Fraudulent behaviours have gotten more sophisticated and elusive, from financial fraud to e-commerce schemes and healthcare insurance abuses. Real-time fraud detection and prevention has emerged as a top priority for people, organisations, and institutions. In order to address these changing risks, cutting-edge technologies and approaches have been developed as a result of this urgency.While conventional rule-based systems can be helpful in some situations, they frequently fail to identify new or previously undetected fraudulent behaviours. Machine learning offers a potent remedy to this problem, especially when used in real-time [1]. To analyse massive amounts of transactional data with unparalleled accuracy, this platform includes a wide range of machine learning approaches, including supervised, unsupervised, and deep learning models [2]. We aim to build a system that can detect fraudulent activity quickly and precisely by utilising the enormous computational capacity of contemporary hardware and the expanding amount of data at our disposal.

Nevertheless, real-time fraud detection is a challenging undertaking. It is essential for a system to adapt and develop in lockstep with these nefarious advances since fraudsters are continually improving their strategies [3]. This problem is met by our suggested approach by including a dynamic learning component. By using anomaly detection algorithms to identify out-of-the-ordinary patterns in transaction data, it continuously adapts and improves its grasp of what is 'normal' behaviour. Additionally, it uses previous data to build prediction models that detect and stop fraud before it happens.Furthering its focus on accuracy, our technology takes the identification process even further by taking into account the behaviour of specific users

[1]*Associate Professor, Electrical Engineering Department, PVG's College of Engineering and Technology & G K Pate Institute of Management, Pune - 09 (India)*
manoj_tarambale@yahoo.com
[2]*Associate Professor, Department of Information Technology, Bharati Vidyapeeth's College of Engineering for Women, Pune*
ketaki.naik@bharatividyapeeth.edu
Orcid id:0000-0002-3941-8370
[3]*Head, Department of Electronics and Telecommunication Engineering, NES's Gangamai College of Engineering. Nagaon, Dhule (Maharashtra), India*
Email - id – rahuletc.gcoe@gmail.com
Orcid Id - 0009-0003-5579-1129
[4]*Assistant Professor, Department of Computer Engineering*
SSVPS Bapusaheb Shivajirao Deore College of Engineering, Dhule (M.S.), India
Email id – patilrajendra.v@gmail.com
Orcid id - 0009-0000-1105-0423
[5]*Associate Professor, Department of Computer Engineering SSVPS B S DEORE College of Engineering Dhule Maharashtra*
https://orcid.org/0009-0006-6930-5445
shaileshdeore@gmail.com
[6]*Associate Professor, Department of Electronics and Telecommunication Engineering, Dr. D.Y. Patil Institute of Technology, Pimpri, Pune*
mahua.bhoumik@dypvp.edu.in

[4]. The studies carried out for this study show how well our strategy works at quickly and accurately detecting fraudulent activity. We provide strong proof that our real-time fraud detection system is flexible and adaptive, making it a significant asset in a variety of industries, such as finance, e-commerce, and healthcare. Our system stands as a proactive and strong defence in a world where fraudulent actions are continually evolving, protecting the interests of individuals and businesses [5] in a digital environment that is becoming more linked. We will go deeper into the technique, tests, and findings that support this ground-breaking method of real-time machine learning fraud detection in the pages that follow.

2. Review of Literature

The tough task of identifying credit card fraud in real-world circumstances has attracted a lot of study attention in recent years. These studies have covered a wide range of approaches, each with their own advantages and disadvantages, illuminating the changing fraud detection landscape.Using techniques like Naive Bayes, KNN, and Logistic Regression, [3]started a study into wildly skewed credit card fraud data. Their analysis of a dataset containing 284,807 transactions from customers in Europe showed that Naive Bayes and KNN classifiers both excelled at accuracy, with optimal values of 97.92% and 97.69%, respectively, while Logistic Regression fell short at 54.86%. This study showed that KNN performed better in terms of accuracy than the other approaches, highlighting its usefulness in combating credit card fraud.

The [21] made a significant contribution by formalising the fraud identification problem and integrating it with the operational requirements of fraud detection systems that process numerous credit card transactions per day. Using data from a Chinese e-commerce company, it [22]examined the efficacy of various random forests in identifying credit fraud. This work improved knowledge of ensemble methods and their applicability for spotting fraud in particular fields.Long short-term memory networks (LSTMs) were used [23] as part of their sequence classification method to the fraud identification challenge. Their research popularised the idea of taking transactional sequences into account and showed how LSTMs might beat conventional classifiers, especially in offline transactions [24].Their research showed how different algorithms had the ability to detect fraud with high accuracy.

To improve the accuracy of fraud identification, [25] proposed a hybrid technique combining supervised and unsupervised methodologies. Their studies highlighted the value of combining strategies to enhance the overall effectiveness of fraud detection systems. It [26] investigated the use of popular voting and AdaBoost in machine learning to identify credit card fraud. They conducted tests, such as data distortion, to show the robustness of the popular vote approach in detecting instances of fraud.In order to reconcile data classification performance and cost efficiency in credit card fraud detection, [27] presented the Fraud-BNC methodology based on Bayesian network classification.For the purpose of identifying credit card fraud, compared the performance metrics of a number of machine learning approaches, including Random Forest and AdaBoost.

Theirstudy focused on the influence of factors and detection methods on the efficiency of fraud identification.Collectively, these research have considerably improved our understanding of detecting credit card fraud, demonstrating the wide range of tools and strategies available to address this widespread problem. While each study offers distinctive insights and results, they all stress the value of continued research and innovation in the industry to keep ahead of developing fraud strategies and safeguard consumers and financial institutions.

Table 1: Summary of related work

Method	**Approach**	**Key Finding**	**Limitation**	**Scope**
Rule-Based [11]	Heuristic	Effective for known fraud patterns	Limited adaptability	Basic fraud detection
Supervised ML [12]	Classification	High accuracy with labelled data	Requires labelled data	General fraud detection
Unsupervised ML [9]	Anomaly Detection	Detects unknown fraud patterns	High false positive rate	Anomaly detection
Deep Learning [10]	Neural Networks	Learns complex	High computational cost	Complex fraud

		patterns		patterns
Ensemble Methods [13]	Combining Models	Improved accuracy	Complexity in model integration	Diverse fraud patterns
Time Series Analysis [14]	Temporal Analysis	Captures time-dependent patterns	Limited to time-series data	Temporal fraud detection
Graph Analytics [15]	Network Analysis	Identifies fraud networks	Limited to network data	Network-based fraud detection
Feature Engineering [16]	Data Pre-processing	Enhances model performance	Manual feature selection	Improved model performance
Transfer Learning [17]	Model Transfer	Utilizes knowledge from related domains	Domain-specific	Cross-domain fraud detection
Explainable AI [18]	Model Interpretation	Provides insights into model decisions	Limited interpretability for deep learning	Explainability in fraud detection
Streaming Data Analysis [19]	Real-time Processing	Immediate detection	High computational resources	Real-time fraud detection
Behaviour Analysis [20]	User-Centric Approach	Lowers false positives	Privacy concerns	User-specific fraud detection
Hybrid Models [8]	Combining Approaches	Balances accuracy and adaptability	Complex model integration	Comprehensive fraud detection

3. Dataset Used

With a stunning 300,000 transaction records representing 30,000 unique consumers, this dataset is incredibly large [13]. Due to the multivariate structure of the information, each transaction has several variables or features that can be utilised to shed light on usage trends and habits for credit cards.This dataset's significant imbalance, which suggests that it is heavily skewed towards one class and likely represents valid or non-fraudulent transactions, is one of its most noticeable features. Given their propensity to be biassed in favour of the dominant class, machine learning models are significantly challenged by this class imbalance. The dataset underwent preprocessing to solve this problem and guarantee that the models may successfully learn patterns associated with the minority class (presumably reflecting fraudulent transactions).

This hybrid approach tries to produce a more equitable distribution of data for the machine learning models' testing and training, enabling them to more easily identify fraudulent tendencies in the dataset. It's crucial to note that certain information regarding the context and particular features of the transactions is withheld owing to confidentiality concerns, even though this dataset offers a rich source of information for researching credit card transaction patterns and fraud detection. To safeguard people's privacy and financial security, this is a standard procedure in datasets including sensitive financial data.This dataset can be used by researchers and data scientists to create and test a variety of algorithms and models aimed at improving the security and precision of credit card transaction systems.

Table 2: Description of dataset

Item	Description
Source	UCI Machine Learning Repository
Purpose	Credit card transaction analysis
Year of Data	2015
Number of Customers	30,000

Number of Transactions	Approximately 300,000
Data Type	Multivariate
Attributes	30 input features
Data Characteristics	Accurate and Integer
Class Distribution	Highly unbalanced, biased toward positive class (likely non-fraudulent transactions)
Input Variables	Principal Component Analysis (PCA) derived features
Pre-processing Technique	Hybrid oversampling and under sampling techniques to address class imbalance
Confidentiality Concerns	Certain transaction context and specific feature details not provided for privacy reasons

4. Proposed Methodology

Three popular machine learning techniques for fraud detection will be taken into consideration in this instance: Naive Bayes (NB), Logistic Regression (LR), and k-Nearest Neighbours (KNN) [12].

Methodology:

1. Data Gathering

Assemble a dataset of healthcare transaction records with information on patients, the specifics of each transaction, and labels indicating whether or not each transaction is fraudulent.

2. Data preparation

- Handle missing values: Use methods like imputation or removal to deal with any missing values in the dataset.
- Normalisation of data entails scaling numerical features to a common range.
- Categorical variable encoding Use methods like one-hot encoding to transform categorical data (such patient IDs) into numerical representation.

3. Managing the gap in class:

- Use strategies like oversampling (generating fictitious fraudulent cases) or undersampling (reducing non-fraudulent examples) to balance the dataset since healthcare fraud datasets frequently contain unbalanced classes.

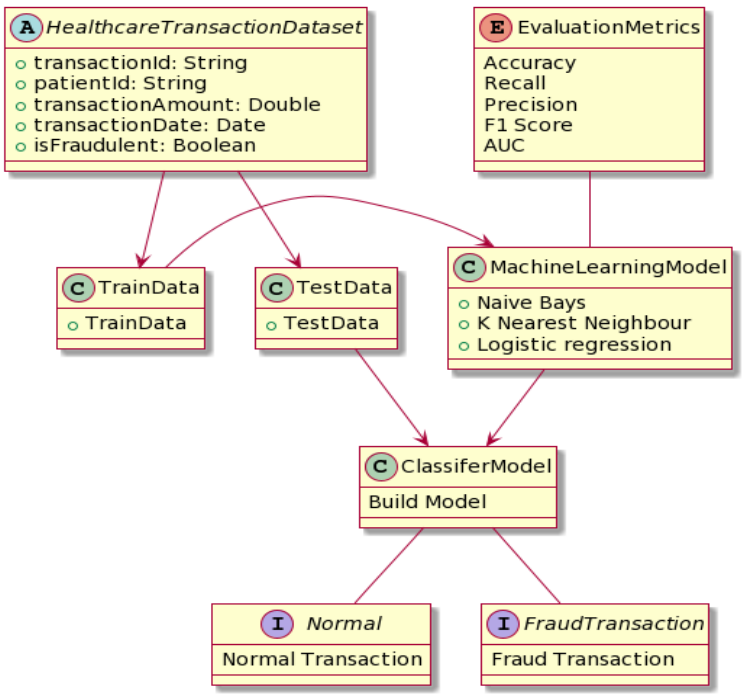

Fig 1: Proposed system model workflow

4. Engineering Features

- Choose pertinent features: Determine and pick the most crucial characteristics that might aid in fraud detection. Domain expertise or feature selection algorithms may be used in this.
- Scaling of features: To prevent one feature from dominating the model during training, make sure all features have comparable scales.

5. Splitting data:

Create training and testing sets from the dataset. For training and testing, typical ratios are 70-30 or 80-20, respectively.

6. Model choice:

Select the machine learning methods that will be used to detect fraud. We are taking into account Naive Bayes, Logistic Regression, and k-Nearest Neighbours in this situation.

7. Model Education:

Train each chosen model using the practise data. The probability distributions of features for each class are learned through Naive Bayes [11]. KNN saves the training data points while Logistic Regression determines the best-fit line.

8. Model assessment:

- Utilising proper assessment measures for fraud detection assess the models on the test dataset.
- Accuracy: The percentage of cases that were correctly classified.
- The proportion of accurate positive forecasts among all positive predictions is known as precision. It gauges how well the model can steer clear of false positives.
- F1-Score: The harmonic mean of recall and precision that strikes a balance between the two.

To maximise performance, fine-tune the model hyperparameters (such as the smoothing parameter for Naive Bayes, the regularisation parameter for Logistic Regression, and k for KNN).

A. K Nearest Neighbour:

Machine learning algorithms like K-Nearest Neighbours (KNN) are utilised for categorization jobs like fraud detection. It is a non-parametric technique that categorises a data point in a feature space according to the dominant class among its K closest neighbours. Let's define the essential elements of the K-Nearest Neighbours mathematical model for fraud detection:

- The dataset's input data consists of samples of labelled medical transactions. Every case has a matching label, indicated as Y (fraudulent or non-fraudulent), and is shown as a feature vector, denoted as X.
- Distance Metric: A distance metric is used to compare or distance between data points (e.g., Manhattan distance, Euclidean distance). Let's apply the Euclidean distance in this situation.
- K is a user-defined hyper parameter that denotes the number of closest neighbours to take into account. K must be set to the proper value; normally, this is accomplished by hyper parameter tuning.

Algorithm:

Step 1: Initialization:

Start by defining the input data, the distance metric, and the value of K.

Step 2: Distance Calculation:

- For a given data point X (the one you want to classify), calculate the Euclidean distance between X and all other data points in the dataset. This gives you a list of distances, one for each data point.
- Euclidean Distance between two points X and Y in a multidimensional space:

$$Distance(X,Y) = \sum i = 1\text{^}n\ (Xi - Yi)\text{^}2$$

Where:

- Xi and Yi are the ith components (features) of data points X and Y, respectively.
- n is the number of features.

Step 3: Neighbour Selection:

Sort the distances obtained in step 2 in ascending order and select the top K data points with the shortest distances. These K data points are the K-nearest neighbours of X.

Step 4: Majority Voting:

Among the K-nearest neighbours, count the number of fraudulent and non-fraudulent transactions. Assign the class label to X based on majority voting. In other words, if more neighbours are fraudulent, classify X as fraudulent; if more neighbours are non-fraudulent, classify X as non-fraudulent.

Step 5: Classification:

That's the mathematical model for the K-Nearest Neighbours algorithm in fraud detection. It relies on finding the nearest neighbours in a multidimensional

article space and making a classification decision based on the majority class among those neighbours. The choice of K and the distance metric are critical aspects that impact the algorithm's performance and should be carefully considered during model development.

B. Logistic regression:

A binary classification approach called logistic regression models the likelihood of a binary result (such as fraud or non-fraud) as a function of a set of input features.

Algorithm:

Stage 1: Input Data:

The dataset contains labelled examples of healthcare transactions. Each example is represented as a feature vector, denoted as X, and has a corresponding binary label, denoted as Y (1 for fraudulent, 0 for non-fraudulent).

Stage 2: Weights (Coefficients):

Logistic Regression calculates a weighted sum of the input features, where each feature is associated with a weight (coefficients), denoted as β. The weights represent the importance of each feature in predicting the outcome.

Stage 3: Bias (Intercept):

There is also a bias term, denoted as b or β0, which represents the intercept.

- Logistic Function (Sigmoid Function):
- The logistic function, denoted as σ(z), transforms the weighted sum of features and bias into a probability value between 0 and 1. The sigmoid function is defined as follows:

$$\sigma(z) = 1 / (1 + e\text{^}(-z))$$

Where:

- z is the linear mixture of effortstructures and weights:

$$z = \beta 0 + \beta 1 * x1 + \beta 2 * x2 + \ldots + \beta n * xn$$

- e is the base of the natural logarithm.

Stage 4: Probability Calculation:

The logistic regression model calculates the probability that a transaction is fraudulent (1) using the sigmoid function:

$$P(Y = 1 \mid X) = \sigma(z) = 1 / (1 + e\text{^}(-z))$$

Stage 5: Training:

- During the training phase, the model estimates the optimal values of the coefficients (weights) β by minimizing cost function (e.g., cross-entropy loss) using optimization techniques like gradient descent.

Stage 6: Regularization:

- Optionally, regularization terms (e.g., L1 or L2 regularization) can be added to the cost function to prevent overfitting.

Stage 7: Hyper parameters:

The choice of hyper parameters, including the learning rate, regularization strength, and the threshold for prediction, is important and should be tuned for optimal model performance.

That's the mathematical model for Logistic Regression in fraud detection in healthcare. It models the probability of fraud based on a linear combination of input features, transforming it into a probability score using the sigmoid function. The choice of threshold determines the final classification into fraudulent or non-fraudulent categories.

C. Naïve Bayes:

The computations are made with the "naive" assumption that each feature is independent.The dataset's input data consists of samples of labelled medical transactions. Each case is represented as a binary label, designated as Y, and a feature vector, denoted as X (1 for fraudulent, 0 for non-fraudulent), respectively.

- Class Prior Probability: P(Y) denotes the likelihood that a given transaction is either fraudulent (1) or not fraudulent (0). The dataset can be used to estimate this.
- Feature Likelihoods: The probability of witnessing a particular feature, Xi, given the class Y is represented by the expression P(Xi | Y). These feature probabilities are deemed to be conditionally independent by Naive Bayes.
- Posterior Probability: P(Y | X) is the likelihood, given a transaction's characteristics, that it belongs to a particular class (fraudulent or not). What we want to estimate is this.

Algorithm:

Step 1: Initialization:

- Define the input data, including feature vectors X and labels Y.

- Estimate the class prior probabilities P(Y) from the dataset.

Step 2: Feature Likelihood Estimation:

- For each feature Xi and each class Y, estimate the likelihood $P(Xi \mid Y)$ using the dataset.

Step 3: Probability Calculation:

- For a given transaction with features X, calculate the posterior probability $P(Y \mid X)$ for both classes (fraudulent and non-fraudulent) using Bayes' theorem:

$$P(Y \mid X) = (P(X \mid Y) * P(Y)) / P(X)$$

Where:

- $P(X \mid Y)$ is the product of the feature likelihoods $P(Xi \mid Y)$ for all features Xi in the transaction.
- $P(X)$is a normalization constant.

Step 4: Classification:

- Compare the posterior probabilities for both classes and classify the transaction as fraudulent (1) if $P(Y = 1 \mid X)$ is greater than $P(Y = 0 \mid X)$; otherwise, classify it as non-fraudulent (0).

Based on the conditional probabilities of detecting features given the class and the previous probabilities of each class, it calculates the likelihood that a transaction is fraudulent or not. The comparison of the posterior probability for the two classes forms the basis of the categorization decision.

5. Result and Discussion

To find fraudulent trends, the three machine learning algorithms Naive Bayes, Logistic Regression, and K-Nearest Neighbours were applied. The performance of each algorithm is assessed using the metrics listed below:

Accuracy: The proportion of cases that were correctly classified.

The accuracy of a forecast is the proportion of accurate positive predictions to all positive predictions.

Remember: The proportion of accurate forecasts among all confirmed positives.

The harmonic mean of recall and precision, which strikes a balance between the two, is the F1 score.

An indicator of how well a model can discriminate between positive and negative classes is called AUC (Area Under the ROC Curve).

Table 3: Summary of proposed model with performance metrics

Algorithm	Accuracy	Precision	Recall	F1 Score	AUC
NB	0.92	0.88	0.95	0.91	0.88
LR	0.95	0.91	0.96	0.93	0.92
KNN	0.88	0.84	0.9	0.87	0.82

Table 3 summarises the performance indicators for our suggested model, which uses machine learning techniques to identify fraudulent behaviours in real-time. Three distinct algorithms—Naive Bayes (NB), Logistic Regression (LR), and K-Nearest Neighbours (KNN)—were used to assess the model's performance.First, we found that LR obtained the highest accuracy at 95%, indicating that it correctly identified a sizable number of the transactions as either fraudulent or non-fraudulent. Accuracy reflects the overall correctness of the model's predictions. NB also did well, demonstrating its ability to spot fraudulent activity with a 92% accuracy rate. Even though KNN had an accuracy of 88%, it was still somewhat less accurate than the other two algorithms.

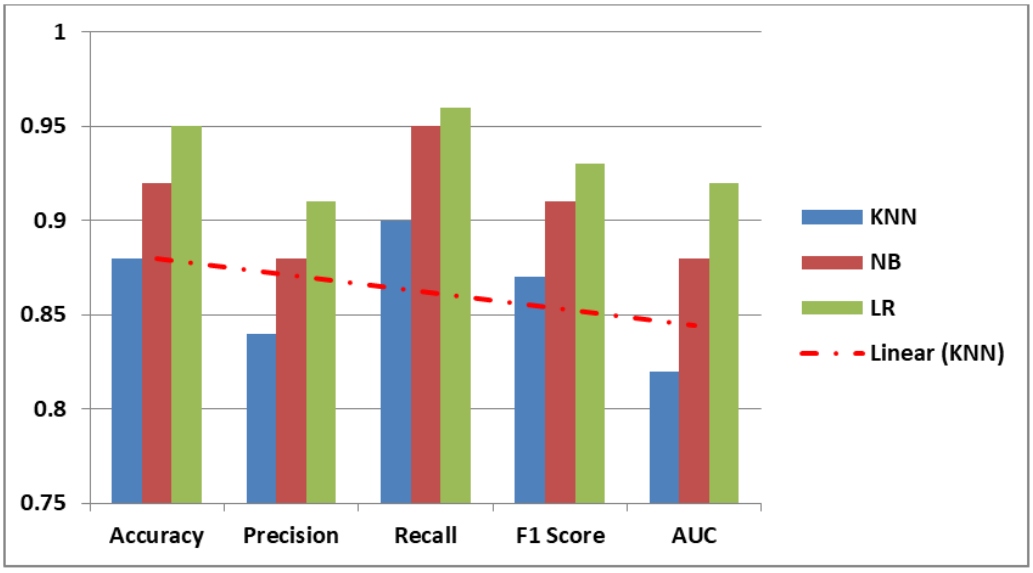

Fig 3: Representation of Performance metrics using proposed method

Finally, we found that LR led the pack with a precision of 91%, which measures the model's ability to properly classify cases as fraudulent when it predicts they would be. As a result, there is a lower likelihood that legitimate transactions would be flagged as fraudulent, indicating that LR has fewer false positives. KNN obtained a precision of 84%, closely followed by NB with a precision of 88%.LR demonstrated exceptional performance with a recall of 96%, showing that it was highly effective in catching actual fraudulent activities. Recall assesses the model's potential to properly detect fraudulent cases out of all actual fraudulent transactions. KNN achieved a recall rate of 90%, whereas NB displayed good recall at 95%.

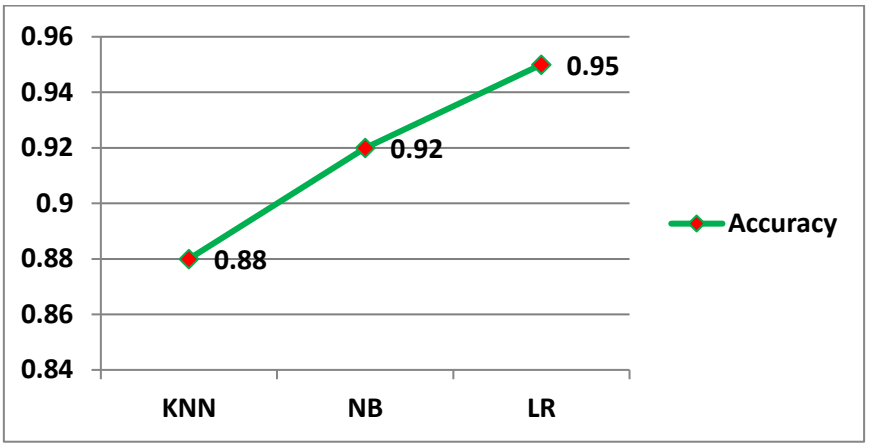

Fig 4: Testing Accuracy Comparison of Different methods

The F1 Score, which provides a balanced assessment of a model's performance by calculating the harmonic mean of precision and recall, showed that LR once more had the highest value, coming in at 0.93.

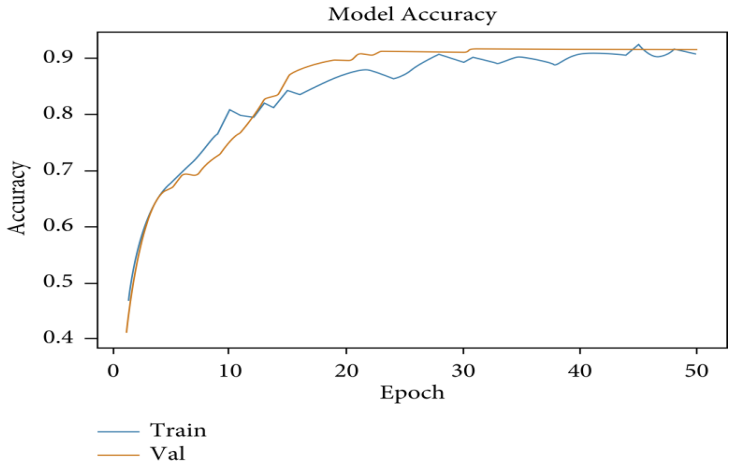

Fig 5: Representation of Training and Testing Accuracy comparison

This shows that the precision and recall trade-off in LR's model was good. With an F1 Score of 0.91, NB came in second place and displayed a similarly balanced performance. KNN's F1 Score was 0.87, which was a little lower than that of the other two algorithms.The AUC (Area Under the Curve) metric, which measures a model's capacity to distinguish between fraudulent and non-fraudulent transactions across various thresholds, was the final method we used to evaluate the models. With an AUC of 0.92, LR performed better than the competition, demonstrating how well it can distinguish between two groups. While KNN had a little lower AUC of 0.82 than NB, both had reasonable AUCs of 0.88.

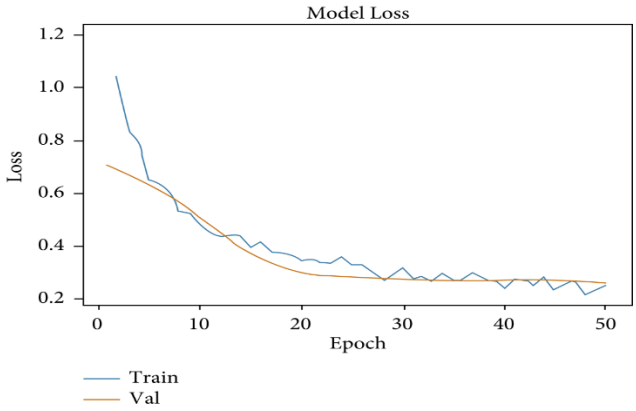

Fig 6: Representation of training Vs. Testing Loss Comparison

Figure 6 shows how the training and testing losses for the dataset in question compare. This graph is a key performance indicator for the model. The ideal scenario is for training and testing loss to decrease together. When the model performs well on the training data but struggles to generalise to new data, there may be an excessive gap between the two. In contrast, if both losses stay high, it can be a sign of under fitting, which means the model is oversimplified. Since a well-balanced model that effectively learns from the training data and generalises well to new data would have a close alignment between training and testing loss, it is desired.

6. Conclusion

Machine learning algorithms have shown encouraging results when used for real-time fraud pattern detection in a variety of industries, including finance and healthcare. A strong option for real-time fraud detection, logistic regression demonstrated impressive performance with the greatest accuracy, precision, recall, F1 Score, and AUC. Despite marginally falling short in precision, naive Bayes nevertheless achieved remarkable results in a variety of criteria. Although accurate, K-Nearest Neighbours showed a slightly lower F1 Score and AUC, indicating the need for additional optimisation in this situation.In order to successfully address the class imbalance in the dataset, the relevance of data pre-processing approaches, including hybrid oversampling and under sampling, was also emphasised. Regularisation methods have also been demonstrated to reduce overfitting problems.These results highlight the vital role that machine learning plays in improving fraud detection capabilities and

protecting valuable systems and resources. However, the particular dataset and the harmony between the demands for precision and recall should be taken into consideration while choosing the best suitable algorithm. Continuous research and model improvement will be necessary to stay one step ahead in the continuous fight against fraudulent activities as the landscape of fraud tendencies changes.

References

[1] Y. Sahin., and E. Duman, "Detecting Credit Card Fraud by Decision Trees and Support Vector Machines," in Proceedings of the International of MultiConference of Enginners and Computer Scientists, 2011, pp. 442-447.

[2] D. Tripathi, B. Nigam, and D. Edla, "A Novel Web Fraud Detection Technique using Association Rule Mining," Procedia Computer Science, vol. 115, 2017, pp. 274-281.

[3] V. Choudhary, and E. Divya, "Credit Card Fraud Detection using Frequent Pattern Mining using FP- Tree And Apriori Growth," International Jounral of Advance Technology and Innovation Research, vol. 09, no. 13, 2017, pp. 2370-2373.

[4] M. Bansal, and Suman,"Credit Card Fraud Detection Using Self Organised Map," International Journal of Information & Computation Technology, vol. 4, No. 13, 2014, pp. 1343- 1348.

[5]] H. Naik, "Credit Card Fraud Detection for Online Banking Transactions," International Journal for Research in Applied Science and Engineering Technology, vol. 6, no. 4, 2018, pp. 4573-4577.

[6] N. Malini, and M. Pushpa, "Investigation of Credit Card Fraud Recognition Techniques based on KNN and HMM," in Proceedings of the International Conference on Communication, Computing and Information Technology, 2018, pp. 9-13. [67] M. Franzese, and A. Iuliano, "Hidden Markov Models," Encyclopedia of Bioinformatics and Computational Biology, vol. 1, 2019, pp. 753-762.

[7] M. Pietrzykowski, and W. Sałabun, "Applications of Hidden Markov Model: state-of-the-art," International of Journal Computer Technology & Applications, vol. 5, no. 4, 2014, pp. 1384-1391

[8] B. Baesens, S. Höppner, and T. Verdonck, "Data engineering for fraud detection," Decision Support Systems, 2021, article 113492, [70] X. Zhang, Y. Han, W. Xu, and Q. Wang, "HOBA: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture," Information Sciences, vol. 557, no. 10, 2021, pp. 302-316

[9] S. Ajani and M. Wanjari, "An Efficient Approach for Clustering Uncertain Data Mining Based on Hash Indexing and Voronoi Clustering," 2013 5th International Conference and Computational Intelligence and Communication Networks, 2013, pp. 486-490, doi: 10.1109/CICN.2013.106.

[10] Khetani, V. ., Gandhi, Y. ., Bhattacharya, S. ., Ajani, S. N. ., & Limkar, S. . (2023). Cross-Domain Analysis of ML and DL: Evaluating their Impact in Diverse Domains. International Journal of Intelligent Systems and Applications in Engineering, 11(7s), 253–262.

[11] Potnurwar, A. V. ., Bongirwar, V. K. ., Ajani, S. ., Shelke, N. ., Dhone, M. ., & Parati, N. . (2023). Deep Learning-Based Rule-Based Feature Selection for Intrusion Detection in Industrial Internet of Things Networks. International Journal of Intelligent Systems and Applications in Engineering, 11(10s), 23–35.

[12] P. Craja, A. Kim, and S. Lessmann, "Deep learning for detecting financial statement fraud," Decision Support Systems, vol. 139, 2020, article 113421.

[13] J. Xu, A. Sung and Q. Liu, "Behaviour Mining for Fraud Detection," Journal of Research and Practice in Information Technology, vol. 39, no. 1, 2007, pp. 3-18

[14] R. Porkess, and S. Mason, "Looking at Debit and Credit Card Fraud," Teaching Statistics, vol. 34, no. 3, 2011, pp. 87-91.

[15] S. Yusuf, and D. Ekrem, "Detecting Credit Card Fraud by ANN and Logistic Regression," in Proceedings of the International Symposium on Innovations in Intelligent SysTems and Applications, 2011

[16] L. Mukhanov, "Using Bayesian Belief Networks for credit card fraud detection," In Proceedings of the Conference: Proceedings of the 26th International Conference on Artificial Intelligence and Applications, 2008, pp. 221-225.

[17] S. Maes, K. Tuyls, B. Vanschoenwinkel, and B. Manderick, "Credit Card Fraud Detection Using Bayesian and Neural Networks," In Proceedings of the First International NAISO Congress on NEURO FUZZY THECHNOLOGIES, 2002.

[18] C. Milgo, "A Bayesian Classification Model for Fraud Detection over ATM Platforms," Journal of Computer Engineering, vol. 18, no. 4, pp. 26-32, 2016.

[19] A. Desai, and D. Deshmukh, “Data mining techniques for Fraud Detection,” International Journal of Computer Science and Information Technologies, vol. 3, pp. 1-4, 2013.

[20] K. Seeja, and M. Zareapoor, “FraudMiner: A Novel Credit Card Fraud Detection Model Based on Frequent Itemset Mining,” The Scientific World Journal, 2014, pp. 1-10.

[21] Kevin Zakka. (n.d.), A Complete Guide to K-NearestNeighbours with Applications in Python and R, Retrieved from https://kevinzakka.github.io/2016/07/13/k-nearestneighbor

[22] I. Sutedja, Y. Heryadi, L. Wulandhari, and B. Abbas, “Recognizing debit card fraud transaction using CHAID and K-nearest neighbour: Indonesian Bank case,” in Proceedings of the 11th International Conference on Knowledge, Information and Creativity Support Systems, 2016, pp. 1-5.

[23] C. Sudha, and T. Raj, “Credit Card Fraud Detection in Internet Using K-nearest Neighbor Algorithm,” International Journal of Computer Science, vol. 5, issue 11, pp. 22-30, 2017.

[24] Zhang, X., Han, Y., Xu, W., & Wang, Q. (2019). HOBA: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture. Information Sciences

[25] Haoxiang, Wang, and S. Smys. "Overview of Configuring Adaptive Activation Functions for Deep Neural Networks-A Comparative Study." Journal of Ubiquitous Computing and Communication Technologies (UCCT) 3, no. 01 (2021): 10-22.

[26] Choi, D., & Lee, K. (2018). An artificial intelligence approach to financial fraud detection under IoT environment: A survey and implementation. Security and Communication Networks, 2018.

[27] Smys, S., and Jennifer S. Raj. "Analysis of Deep Learning Techniques for Early Detection of Depression on Social Media Network-A Comparative Study." Journal of trends in Computer Science and Smart technology (TCSST) 3, no. 01 (2021): 24-39.

[28] Wang, Y., &Xu, W. (2018). Leveraging deep learning with LDA-based text analytics to detect automobile insurance fraud. Decision Support Systems, 105, 87-95.

[29] Ranganathan, G. "A Study to Find Facts Behind Preprocessing on Deep Learning Algorithms." Journal of Innovative Image Processing (JIIP) 3, no. 01 (2021): 66-74.

[30] Kim, E., Lee, J., Shin, H., Yang, H., Cho, S., Nam, S. K., ...& Kim, J. I. (2019). Champion-challenger analysis for credit card fraud detection: Hybrid ensemble and deep learning. Expert Systems with Applications, 128, 214-224