Fraud Detection Using Artificial Intelligence

Dr.N. Baggyalakshmi[1*], V. Bharathi[2], Dr.R. Revathi[3]

[1*] Assistant Professor, Department of Computer Science, PSGR Krishnammal College for Women, Coimbatore.
E-mail: baggyanethra@gmail.com
[2] Student, B.sc Computer Science, PSGR Krishnammal College for Women, Coimbatore. E-mail: 2307bharathi@gmail.com
[3] Assistant Professor, Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore.
E-mail: ravathilakshay@gmail.com

***Abstract*---** With the rising prevalence of digital transactions and online services, the risk of fraudulent activities has grown exponentially. Traditional rule-based systems for fraud detection often struggle to keep up with evolving tactics used by malicious actors. This research delves into the application of machine learning modules in detecting fraud across various domains, such as financial transactions, insurance claims, and online platforms. Moreover, the study investigates the combination of these methods in ensemble approaches to achieve higher accuracy and robustness. Additionally, feature engineering and data pre-processing techniques play a crucial role in enhancing model performance. We delve into the identification and selection of relevant features and the impact of different data transformations on the detection process. Furthermore, addressing the challenge of imbalanced datasets, we examine techniques like oversampling, under sampling, and synthetic data generation to mitigate class imbalance and enhance fraud detection efficiency. To evaluate the models, we utilize publicly available benchmark datasets and conduct comparative analyses to highlight their strengths and weaknesses. The results showcase the superiority of certain models for specific fraud scenarios and underscore the importance of a tailored approach for different domains.

***Keywords*---** Fraud Detection, Artificial Intelligence, Financial Transactions.

I. INTRODUCTION

In recent years, the rapid growth of digital transactions and online services has significantly increased the risk of fraudulent activities across various domains, including finance, insurance, e-commerce, and more [1]. Traditional rule-based fraud detection systems have proven insufficient in dealing with the ever-evolving tactics employed by fraudsters, leading to a demand for moresophisticated and adaptive solutions. This has paved the way for the application of machine learning techniques in fraud detection.

Machine learning offers the potential to effectively detect fraudulent behaviour by learning patterns and anomalies from historical data [2]. By leveraging large datasets, machine learning algorithms can uncover hidden insights and createpredictive models that identify potential fraudulent transactions or activities. The primary advantage of using machine learning for fraud detection lies in its ability to adapt to new and emerging fraud patterns without explicit programming [3]. Instead of relying on rigid rule sets, machine learning models can autonomously learn from new data, continuously improving theiraccuracy over time.

This introductory study on fraud detection using machine learning aims to explore various approaches and methodologies employed in detecting fraudulent activities [4]. Furthermore, feature engineering and data preprocessing will be discussed, as these play a vitalrole in optimizing model performance. By selecting relevant features and applying data transformations, we can enhance the effectiveness of fraud detection models [5]. The issue of imbalanced datasets, where legitimate transactions vastly outnumber fraudulent ones, poses a challenge for accurate fraud detection. To explore various techniques like oversampling, under sampling, and synthetic data generation to address this class imbalance problem.

Through comparative analyses using benchmark datasets, we will evaluate the strengths and weaknesses of different machine learning models, shedding light on their suitability for specific fraud scenarios. Moreover, we will discuss the importance of interpretability and scalability when deploying these models in real-time environments. Overall, this study aims to contributeto the advancement of fraud detection techniques using machine learning, providing valuable insights and guidance for designing robust and adaptive systems to combat fraudulent activitieseffectively in today's increasingly digital world.

II. LITERATURE REVIEW

Buyuktepe et al [6] One tactic to lessen these risks is to integrate artificial intelligence (AI) technology into the food supply chain. Precision nutrition, self-driving cars, precision

agriculture, precision medicine, and food safety are just a few of the industries that have seen an increase in the usage of AI. However, because AI is not really explainable, a lot of what AI systems accomplish is opaque. This paper discusses many applications of explainable artificial intelligence (XAI) methods, including LIME, SHAP, and WIT, for predicting food fraud risk. Using these tools, our goal was to interpret machine learning model predictions. Using adulteration/fraud notifications obtained from the economically driven adulteration database and the Rapid Alert System for Food and Feed system, the case study was conducted on a food fraud dataset.

Viswanatha et al [7] The employment documented exchange information framework that was suggested combined a number of elements, including budgetary data, client behavior, and exchanges. The information is first cleaned and converted into an organized understandable format for the planning show using the information control preparation. Subsequently, various machine learning computations, including choice trees, calculated relapse, irregular timberlands, and angle boosting, are employed to construct predictive algorithms capable of identifying fraud. The extension ends with the use of the produced program in a real-world online transaction setting, enabling real-time extortion avoidance and location.

Ileberi et al [8] This study presents a genetic algorithm (GA) based machine learning (ML) based credit card fraud detection engine that selects features. The suggested detection engine employs the following machine learning classifiers after selecting the optimal features: Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), Artificial Neural Network (ANN), and Naive Bayes (NB). The suggested credit card fraud detection engine is assessed using a dataset created from European cardholders in order to verify the performance. The outcome proved that our suggested strategy works better than the current ones.

Aslam et al [9] This study uses 15 predictive models to present a framework for fraud detection in the vehicle insurance market. Using a publicly accessible auto insurance 16 dataset, the feature selection process finds the most significant feature using the Boruta algorithm. To create an 18 effective fraud detection system, three prediction models—naïve Bayes, support vector machines, and logistic regression—are used. The confusion matrix is used to calculate six metrics that are used to evaluate the predictive model's 19 performance. The findings show that the logistic regression obtains the highest f-measure 21 score, but the support vector machine 20 performs better in terms of accuracy.

Btoush et al [10] Cybersecurity is becoming the banking industry's top priority due to the growing prevalence of cyberattacks and criminal activity. Globally, credit card cyber fraud is a significant security problem. Though they are the most time-consuming, resource-intensive, and incorrect methods for spotting cyber fraud, conventional anomaly detection and rule-based algorithms are two of the most often used methodologies. In this discipline, machine learning is one of the methods that is becoming more and more popular. This paper looks at and summarizes earlier research on the identification of credit card cyber fraud. Examining machine learning and deep learning techniques is the main goal of this review.

III. PROPOSED METHODOLOGY

1.1. Fraud Detection with Google Cloud AI Platform

Fraud detection with Google Cloud AI Platform involves leveraging Google Cloud's infrastructure and machine learning capabilities to build, train, and deploy fraud detection models. The general steps to implement fraud detection using Google Cloud AI Platform are:

Data Collection and Preprocessing

Gather relevant data from various sources, such as transaction logs, user behaviour data,and historical records.

Preprocess the data by cleaning, transforming, and normalizing it to prepare it for modeltraining.

Data Storage

Store the pre-processed data in Google Cloud Storage or Big Query, depending on thedata size and complexity.

Data Exploration and Analysis

Use Google Data Studio or Google Big Query for data exploration and visualization togain insights into the data and identify patterns.

Model Development

Choose appropriate machine learning algorithms and techniques for fraud detection. Common algorithms include logistic regression, decision trees, random forests, and deep learning-based models.

Develop the fraud detection model using Python and popular machine learning librarieslike TensorFlow or scikit-learn.

Model Training with AI Platform Training

Use Google Cloud AI Platform Training to train the fraud detection model at scale. AIPlatform Training provides a managed service for distributed training with large datasets.

Utilize GPU or TPU accelerators for faster training, especially for deep learning models.

Hyperparameter Tuning (Optional)

Employ AI Platform Hyperparameter Tuning to automatically search for the besthyperparameters that optimize the model's performance.

Model Evaluation

Evaluate the trained model's performance using evaluation metrics such as accuracy,precision, recall, and F1-score on a separate validation dataset.

Model Deployment with AI Platform Prediction

Deploy the trained fraud detection model using Google

Cloud AI Platform Prediction,which allows for scalable and high-performance model inference.

Expose the model as an API endpoint for real-time scoring of new transactions.

Continuous Monitoring and Feedback Loop

Continuously monitor the model's performance and gather feedback from fraud analyststo improve its accuracy over time.

Periodically retrain the model with new data to ensure it adapts to evolving fraudpatterns.

Alerting and Integration with Business Processes

Set up alerts and notifications to trigger actions when the model identifies suspicious transactions.

Integrate the fraud detection model into the business processes, such as payment processing systems, to take real-time actions based on the model's predictions.

Google Cloud AI Platform provides a scalable and flexible infrastructure to implement fraud detection solutions efficiently. It allows data scientists and developers to focus on building accurate and effective models while taking advantage of Google Cloud's computing power and AI capabilities. Additionally, Google Cloud's security measures help protect sensitive data andensure compliance with data privacy regulations.

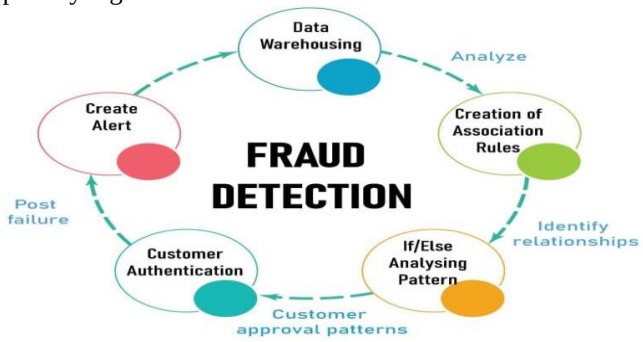

Figure 1 - Fraud Detection

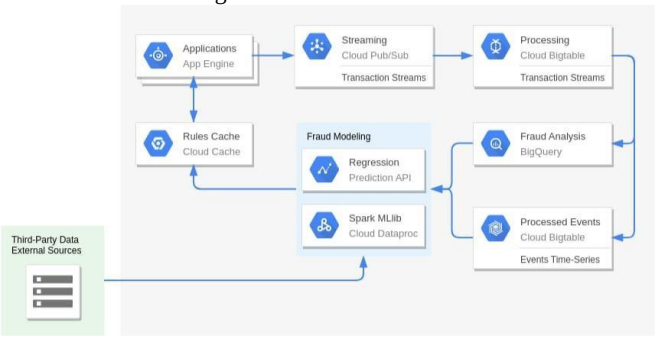

Figure 2 - Google Cloud AI Platform

1.2. Interface

1.2.1. Internal Interface

In fraud detection using machine learning, the internal interface refers to the communication and interaction between different components and modules within the fraud detection system.These components work collaboratively to analyse data, detect anomalies, and identify potentially fraudulent activities. The internal interface facilitates the flow of information and decisions among these modules, ensuring an effective and efficient fraud detection process. This component is responsible for collecting and processing incoming data from various sources, such as transaction logs, user behaviour data, and historical records. It ensures that the data is cleansed, normalized, and pre-processed before being used by other modules. Feature engineering involves transforming raw data into meaningful features that can be used by machine learning models.

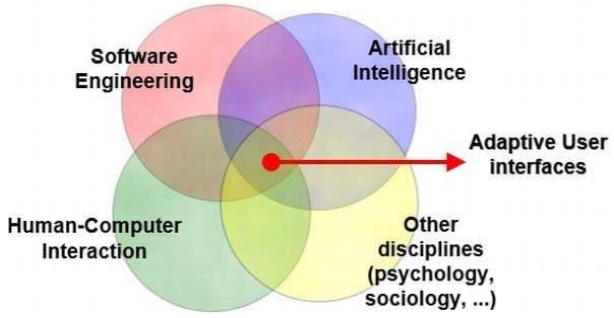

Figure 3 - Internal Interface Using AI Platform

1.2.2. External Interface

The external interface in fraud detection using machine learning refers to the interaction and communication between the fraud detection system and external entities, such as users, applications, and other systems. It enables the fraud detection system to receive input data, provide outputs, and collaborate with external stakeholders. The fraud detection system relies on various external data sources to gather information about transactions, user behaviour, andhistorical records. These sources may include financial institutions, e-commerce platforms, payment gateways, user databases, and third-party data providers. The system must be able to securely access and ingest data from these sources for analysis. To interact with other applications and systems, the fraud detection system often exposes Application Programming Interfaces (APIs). These APIs endless data exchange and real-time communication between the fraud detection system and external applications.

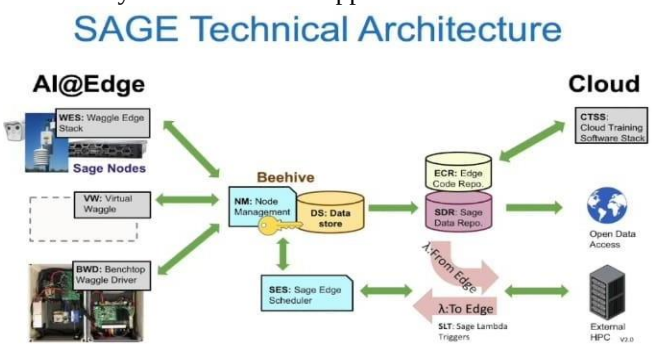

Figure 4 - Sage Technical Architecture

1.2.3. User Interface

The user interface (UI) in fraud detection using machine learning serves as the primary interaction point for fraud analysts, investigators, and other relevant stakeholders. It provides a visual and user-friendly way to access and interact with the fraud detection system, allowingusers to review,

investigate, and act on flagged transactions and potentially fraudulentactivities. The UI typically presents a dashboard that provides a high-level overview of the fraud detection system's performance. This summary may include key metrics such as the totalnumber of transactions processed, the number of flagged transactions, the overall fraud rate, and the accuracy of the machine learning model. The user interface displays a list of recent transactions, including both legitimate and flagged transactions. Each transaction entry typically shows relevant information such as transaction amount, date and time, user details, and any associated risk scores.

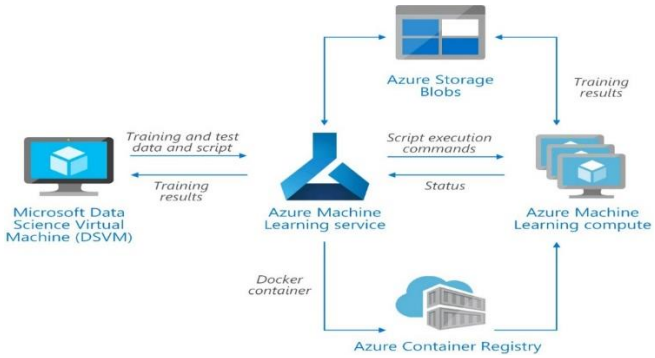

Figure 5 - user Interface Using AI Platform

1.3. Different Files Used Tensor Flow

TensorFlow is primarily a machine learning framework and library, and it is not directly relatedto generating HTML files. TensorFlow is designed for numerical computations, data flow graphs, and training machine learning models, mostly used in Python programming language TensorFlow can be used in web applications as part of the back-end for tasks such as serving machine learning models, data processing, and predictions. In these cases, the HTML files are typically generated by the front-end part of the web application, which may use HTML, CSS,and Java .

To clarify, here are some common HTML files you may find in a web application that uses TensorFlow in the back end:index.html: The main HTML file that serves as the entry point of the web application. It typically includes references to CSS and JavaScript files and may contain the structure of the user interface.result.html: An HTML file used to display the results of machine learning predictions or other processed data from the TensorFlow back end.form.html: If the application requires user input or data submission, a form.html file may be used to collect data from users and send it to the TensorFlow back end for processing.dashboard.html: For applications with a dashboard or visualizations, this HTML file may be used to display various charts and data representations generated by the back-end TensorFlow models.error.html: An error.html file is used to display error messages or handle exceptions that may occur during data processing or model predictions.

TensorFlow itself doesn't generate or manage HTML files directly. It is up to the web application developers to design the front end, including HTML files, to interact with the TensorFlow back end and present the results and functionality to users.

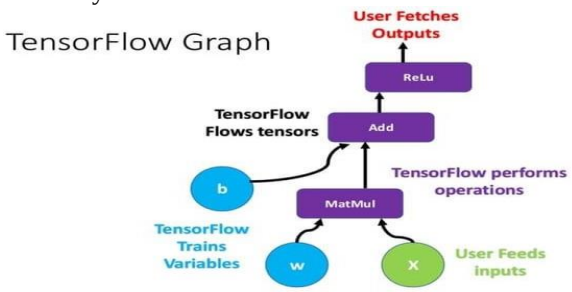

Figure 6 - Tensor Flow Graph

1.4. System Design

1.4.1. Input Design

Input Design in fraud detection system using machine learning involves several steps:

- **Data Collection:** Gather historical data on past transactions, including both legitimate and fraudulent ones, to create a labelled dataset.
- **Data Preprocessing:** Clean and preprocess the data to handle missing values, outliers, andstandardize the features.
- **Feature Engineering:** Extract relevant features from the data that could be indicative of fraud,such as transaction amount, location, device used, time of day, etc.
- **Model Selection:** Choose appropriate machine learning algorithms for fraud detection, such as logistic regression, decision trees, random forests, or neural networks.
- **Model Training:** Split the dataset into training and testing sets, and train the selected modelson the training data.
- **Model Evaluation:** Evaluate the performance of each model using metrics like precision, recall,F1-score, and ROC-AUC to assess their effectiveness in detecting fraud.
- **Hyperparameter Tuning:** Fine-tune the model's hyperparameters to optimize its performance.
- **Ensemble Methods:** Consider using ensemble techniques like stacking or boosting to combinethe strengths of multiple models.
- **Real-time Monitoring:** Implement the model in a real-time environment, continuouslymonitoring transactions for potential fraud.
- **Feedback Loop:** Periodically update the model with new data to adapt to evolving fraudpatterns.

1.4.2. Code Design

Designing the code for fraud detection using machine learning involves several steps such as:

Data Preparation

- Load and preprocess the historical transaction data, handling missing values andoutliers.
- Split the dataset into features (X) and labels (y) for training the model.

Feature Engineering

- Extract relevant features from the data that could be indicative of fraud.
- Perform feature scaling or normalization to ensure consistent ranges for the features.

Model Selection and Training

- Choose the appropriate machine learning algorithm(s) for fraud detection.
- Split the dataset into training and testing sets.
- Train the selected model(s) on the training data.

Model Evaluation

- Evaluate the performance of the trained model(s) using various metrics like precision, recall, F1-score, ROC-AUC, etc.

Hyperparameter Tuning

- Fine-tune the model's hyperparameters using techniques like grid search or randomsearch to optimize its performance.

Real-time Prediction

- Implement a function to process new, real-time transaction data and make predictions using the trained model
- Assign fraud scores or probabilities to each new transaction.

Threshold Setting and Decision Making

- Set appropriate thresholds for the fraud scores to classify transactions as low, medium,or high risk.
- Based on the thresholds, make decisions such as flagging, blocking, or manual review

Alerts and Notifications

- Implement an alerting mechanism to notify relevant stakeholders when high-risk transactions are detected.

Visualization

- Create interactive visualizations to present fraud trends and model performancemetrics.

Feedback Loop and Model Updates

- Implement a feedback loop to continuously update the model with new labelled datafor better perform over time.

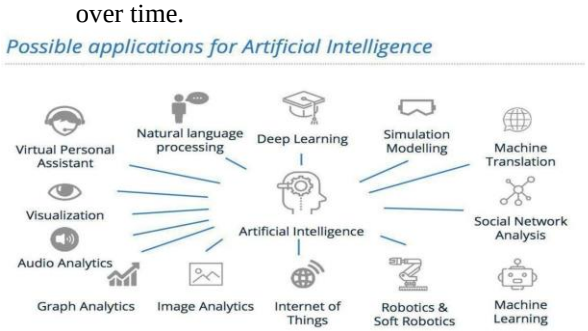

Figure 7 - Possible Application for Artificial Intelligence

1.5. Output Design

The output design for fraud detection using machine learning should focus on providing actionable and reliable insights to facilitate decision-making. The key components of the outputdesign:

- **Fraud Score/Probaility:** For each transaction, the system should generate a fraud score or probability indicating the likelihood of the transaction being fraudulent. This score helps prioritize suspicious transactions for further investigation.
- **Thresholds:** Set appropriate thresholds for the fraud score to classify transactions as low, medium, or high risk. Depending on the organization's risk tolerance, these thresholds can trigger different actions, such as flagging, blocking, or manual review.
- **Alerts and Notifications:** Implement an alerting mechanism to notify relevant stakeholders (security teams, analysts, or even customers) when a transaction exceeds the specified risk threshold, enabling timely action.
- **Transaction Details:** Include relevant transaction details, such as transaction amount, location,time, and user information, in the output to provide context for fraud analysts.
- **Visualization:** Utilize interactive and informative visualizations to present fraud trends, patterns, and statistics, aiding analysts in spotting potential fraud clusters and anomalies.
- **Real-time Monitoring Dashboard:** Develop a real-time monitoring dashboard to display live fraud detection metrics, allowing quick assessments of the system's performance.
- **Model Performance Metrics:** Display evaluation metrics (e.g., precision, recall, F1-score, ROC-AUC) to assess the model's performance over time and identify areas for enhancement.
- **Audit Trail:** Maintain a comprehensive audit trail that logs all model-related activities, ensuring transparency and compliance with data regulations.
- **Compliance Reports:** Generate regular compliance reports for internal and external stakeholders, detailing the effectiveness of the fraud detection system.

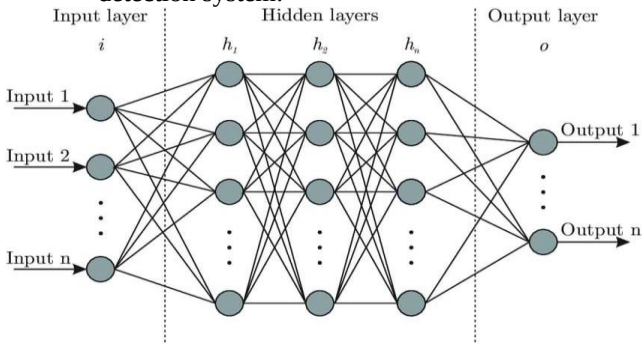

Figure 8 - Output Design Using AI Platform

IV. Results and Discussion

1.6. System Specification

Table 1: Hardware Requirement

Processor	Intel i5 or i7
Installed Memory (RAM)	8GB
Hard Disk	512 GB
Operating system	Windows 11

Table 2: Software Requirements

Front End	HTML, CSS
Back End	Python
Tool	Tensor Flow
Database	Microsoft SQL Server

1.7. Software Specifications

Front End

The front-end for fraud detection serves as the user interface that enables users, such as analystsand investigators, to interact with the fraud detection system. It plays a crucial role in presenting the results of the machine learning models and facilitating decision-making processes. Here aresome key aspects and functionalities typically found in the front-end for fraud detection:

Dashboard: The front-end often features a dashboard that provides an overview of recent activities, detected fraud incidents, and key performance metrics of the fraud detection system. This summary view helps users quickly grasp the current state of fraud activities.

Real-time Monitoring: The front-end allows users to monitor transactions and activities in real-time. It may display incoming transactions with color-coded indicators to highlight potential fraud risks.Alerts and Notifications: The system can generate alerts or notifications whenever a suspicious transaction is detected. These alerts can be shown prominently on the front-end todraw immediate attention to potential fraud incidents.

Data Visualization: Effective data visualization tools are essential in presenting complex fraud-related data in an easy-to-understand manner. Graphs, charts, and heatmaps can be utilized to show trends, patterns, and anomalies.

Back End

The back end for fraud detection using machine learning is the core component of the system responsible for processing data, training machine learning models, and making predictions onincoming transactions or activities. It typically involves several key components and functionalities, including:

Data Ingestion and Preprocessing: The back end collects transaction data from various sources, such as databases, APIs, or streaming platforms. It then preprocesses the data to clean and transform it into a suitable format for machine learning algorithms. Data preprocessing involves tasks like feature engineering, handling missing values, and dealing with imbalanceddatasets.

Feature Extraction and Selection: Relevant features are extracted from the pre- processed data to provide meaningful information for the machine learning models. The back end may employ techniques like dimensionality reduction and feature selection to optimize model performance.

Machine Learning Model Training: The back end uses historical transaction data, labelled with fraudulent and legitimate instances, to train machine learning models. Various algorithms, suchas Random Forest, Support Vector Machines, Neural Networks, or ensemble methods, are utilized to learn patterns and relationships between features to detect fraud.Model Evaluation and Tuning: After training the models, the back end evaluates their performance using evaluation metrics like accuracy, precision, recall, and F1 score. Model hyperparameters may be tuned to optimize performance and generalization.Real-time Prediction: The trained machine learning models are deployed in the back end to make real-time predictions on incoming transactions.

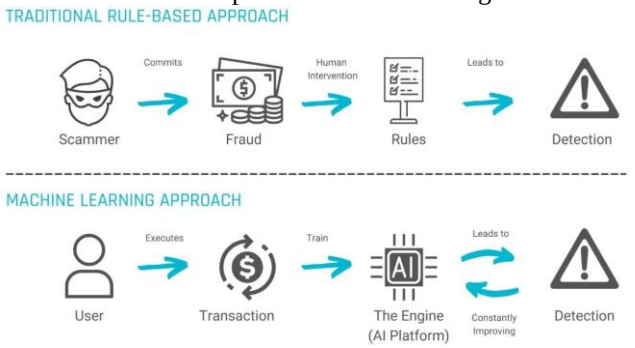

Figure 9 - Traditional Approach VS Machine Learning Approach

Fraud Detection with Google Cloud AI Platform

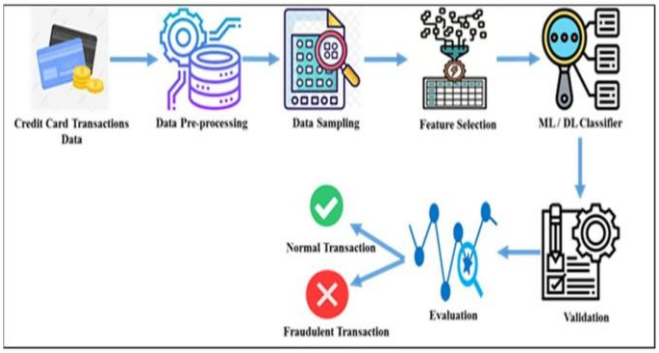

Figure 10 - Fraud Detection with Google AI Platform

V. Conclusion

The application of machine learning in fraud detection has proven to be highly effective and advantageous in modern times. By harnessing the power of advanced algorithms, data analysis,and pattern recognition, machine learning models have demonstrated remarkable success in identifying fraudulent activities across various industries. As fraudsters continually evolve theirtactics, machine learning models can adapt and update their algorithms to stay ahead of new fraudulent schemes. By accurately detecting fraud, machine learning helps build trust with customers, as their financial and personal information is better protected. it's essential to acknowledge that machine learning models are not

foolproof. Challenges such as imbalanced datasets, concept drift, and adversarial attacks can still impact the effectiveness of fraud detection systems. Regular monitoring, model retraining, and a combination of different techniques are necessary to maintain the efficiency of fraud detection systems over time.In summary, the integration of machine learning in fraud detection has brought about significant advancements in safeguarding businesses, financial institutions, and consumers from fraudulent activities. As technology continues to evolve, ongoing research and innovation in this field will further strengthen fraud prevention measures, making transactions and online interactions safer for everyone involved.

REFERENCE

[1] Bao, Y., Hilary, G., & Ke, B. (2022). Artificial intelligence and fraud detection. *Innovative Technology at the Interface of Finance and Operations: Volume I*, 223-247.

[2] Ikhsan, W.M., Ednoer, E.H., Kridantika, W.S., & Firmansyah, A. (2022). Fraud detection automation through data analytics and artificial intelligence. *Riset: Jurnal Aplikasi Ekonomi Akuntansi dan Bisnis*, *4*(2), 103-119.

[3] Prajapati, M.Y., Parasar, M.A., & Khande, R. (2023). An Analysis of Financial Fraud Detection Methods Using Artificial Intelligence. *Vidhyayana-An International Multidisciplinary Peer-Reviewed E-Journal*, *8*(si7), 79-95.

[4] Mytnyk, B., Tkachyk, O., Shakhovska, N., Fedushko, S., & Syerov, Y. (2023). Application of Artificial Intelligence for Fraudulent Banking Operations Recognition. *Big Data and Cognitive Computing*, *7*(2), 1-19.

[5] Vesna, B.A. (2021). Challenges of financial risk management: AI applications. *Management: Journal of Sustainable Business and Management Solutions in Emerging Economies*, *26*(3), 27-34.

[6] Buyuktepe, O., Catal, C., Kar, G., Bouzembrak, Y., Marvin, H., & Gavai, A. (2023). Food fraud detection using explainable artificial intelligence. *Expert Systems*, 1-20.

[7] Viswanatha, V., Ramachandra, A.C., Deeksha, V., & Ranjitha, R. (2023). Online Fraud Detection Using Machine Learning Approach. *International Journal of Engineering and Management Research*, *13*(4), 45-57.

[8] Ileberi, E., Sun, Y., & Wang, Z. (2022). A machine learning based credit card fraud detection using the GA algorithm for feature selection. *Journal of Big Data*, *9*(1), 1-17.

[9] Aslam, F., Hunjra, A.I., Ftiti, Z., Louhichi, W., & Shams, T. (2022). Insurance fraud detection: Evidence from artificial intelligence and machine learning. *Research in International Business and Finance*, *62*.

[10] Btoush, E.A.L.M., Zhou, X., Gururajan, R., Chan, K.C., Genrich, R., & Sankaran, P. (2023). A systematic review of literature on credit card cyber fraud detection using machine and deep learning. *PeerJ Computer Science*, *9*, 1-66.