DETECTION OF ABNORMAL FEATURES IN TRANSACTION DATA FOR MALICIOUS ATTACKS USING HIERARCHICAL NETWORK FEATURE EXTRACTION

DAMODHARAN KUTTIYAPPAN[1], DR RAJASEKAR V[2]

[1] Ph.D. Research Scholar, Computer Science & Engineering, SRMIST, Vadapalani, Chennai, India

[2] Associate Professor, Computer Science & Engineering, SRMIST, Vadapalani, Chennai, India

E-mail: [1] dt3388@srmist.edu.in, [2] rajasekv2@srmist.edu.in

ABSTRACT

The growing number and complexity of financial transactions have made the detection of fraudulent activity and cyberattacks a considerable issue for organizations. Conventional rule-based systems and statistical techniques frequently fail to detect complex assaults that masquerade as ordinary transactions. This work presents a deep learning methodology for detecting anomalous aspects in transaction data to efficiently identify potential threats. Feature extraction technique has been used to find the suitable features for further processing the dataset. Deep learning techniques are used to classify transactions as normal or anomalous. This study introduces a deep learning methodology for detecting anomalous transaction data, using advanced techniques like CNNs, RNNs, and Autoencoders to identify potential threats. The model outperforms conventional detection techniques in precision, recall, and F1-score, providing insights into abnormal behavior, and aiding in attack discovery and mitigation. The system evolves through ongoing learning, enhancing its detection precision for changing assault patterns.

Keywords; *Deep Learning, CNN, RNN, Anomalies, Threats, Feature Extraction, Financial Transaction*

1. INTRODUCTION

The growing dependence on digital transactions and online platforms has rendered the security of transaction data essential. Financial institutions and e-commerce platforms routinely handle extensive data, rendering them attractive targets for hackers. These assaults may present as fraudulent transactions, account hijackings, and diverse forms of financial cybercrime. Standard detection technologies, which insist mostly on rule-based systems, frequently fail to properly identify complex and most complex and dynamic threats [1]. As a result, there is a big and widespread demand for innovative methods to identify anomalous characteristics for secured payment transaction data that could be evidence of an assault. Significance techniques such as deep learning which is a subset of machine learning collected by multilayered neural networks, exhibit outstanding productivity across more sectors. The standard technique is Deep learning, which utilizes its capacity to notice complex concepts, patterns, and relationships in extensive datasets. It is a viable method for improving the detection of fraudulent actions in transaction data [1].

This study examines the utilization of deep learning methodologies for detecting anomalous characteristics in transaction data, emphasizing their capacity to enhance attack detection rates and diminish false positives. This study seeks to demonstrate how deep learning, through the utilization of models such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and autoencoders, may enhance the efficacy of security protocols in financial transactions [2].

Now the significance of feature engineering and pre-processing for the major productivity of transaction data for deep learning techniques is consumed through the developing and mounting of these models. This scheme or method seeks to illuminate the incorporation of deep learning techniques into present security mechanisms and protocols, eventually increasing the response to the progressed cyber risks assaulting the financial models for secured payment transactions in worldwide [3].

Credit cards have primarily been utilized for payment transactions due to their comfort and broad usage which is widely propagated for easy transactions. Their experience presents the significance and importance of the difficulty of encapsulating personal and suitable payment information from fraudulent and unauthorized access. Intractable security and common protocol mechanisms are needed to protect trust and confidence among users. The most important mechanism and goal of financial strategies is the operation of credit cards, triggering consumers and businesses to transmit their payment transactions as cashless in a protected way. Credit card fraud disguises a substantial threat attached to the utilization of credit cards. Credit card fraud must have resulted in significant financial and notoriety consequences for both regular users and IT infrastructure or corporate or large-scale organizations. Single person affected by credit card fraud might be bearing the financial challenges of illicit charges, potentially bruising their credit points and overall credit rating. The study of credit card fraud imposes the investigation of credit card payment transactions to identify and preclude fraudulent possibilities [4]. This impacts considerably, protecting customers and financial institutions from economic losses, securing individual credentials and information to improve confidentiality, and preserving conviction in electronic payment systems for better results. Usually, Credit card fraudulent detection methods involve several measures and steps incorporated and designed to identify for protect against unauthorized or fraudulent payment transactions. These methods mainly utilize enlightened technology, such as machine learning techniques to examine protocol figures, anomalies, and user activities related to credit card usage in a widespread manner. The rapid expansion of digital transactions has made the identification of fraudulent activities a significant challenge for corporations. Traditional approaches, like rule-based systems and statistical models, very often waver in detecting new and complex threats, since cybercriminals immutable alter their schemes for flawless easy payment transactions [5].

1.1 **Deep learning in Malicious attack**

The addition of digital surroundings has given good ideation in an increase in malicious attacks, requiring present detection and safeguarding tools and mechanisms. These vigorous techniques such as Deep learning, have a supremacy subset of machine learning, incorporating multilayered neural networks to diagnose and reproduce complex signs and patterns in large datasets. Malicious attacks exhibit more endangered attacks such as malware, phishing, Denial of Service, intrusion attempts, and Advanced Persistent Threats. These widespread techniques such as Deep learning methodologies emerged Convolutional Neural Networks (CNNs), Recurrent Neural Networks, Long Short-Term Memory Networks, Autoencoders, and Generative Adversarial Networks for protecting and becoming visible for easy and flexible payment transactions [6].

The utility of deep learning techniques bounded improved security over accuracy, automated feature extraction, real-time system detection, and workability to new attack courses. Despite that, hurdles in implementation enlightened data security and enhanced quality and quantity too, model compressibility, and mechanism and availability demands. The utmost quality and quantity of large datasets are needed for training on decomposition, Nonetheless, model compressibility might be discomposed comprehension and hinder critical techniques deployed on smarter applications such as cybersecurity in a tremendous manner [7].

The succeeding part of this work are emphasized as follows: Section 2 exhibited a literature review of current techniques drawn for fraudulent detection and anomaly detection in a transactional larger dataset. Section 3 depicts the proposed concepts of the massive deep learning model techniques and the methodology for larger data processing. Section 4 describes the exploratory configuration and computational criteria. Section 5 defines the resultant, achieved by an exchange of the findings in Section 6. Tremendously, Section 7 completes the task and outlines a potential research path

.

2. Literature survey:

In this fraud detection methodology and anomaly detection, we have increased from conventional rule-based systems to high-level machine learning and deep learning techniques models. Rule-based systems are quite widespread activities to improve and understand, though previous methodologies exhibit hardness and unavailability of the ability to achieve emerging fraudulent techniques. Analytical techniques, which include probability distributions and regression analysis, diagnose previous transaction data to identify signs and patterns that might be insisting on fraudulent activity. Previous techniques have been effective for detecting techniques and anomalies in the large dataset but might have issues in adjusting to become evident for fraud practices. In this Machine learning methodologies, which include supervised and unsupervised learning, have significantly increased fraudulent detection by empowering computers to stand-alone practices learn from previously enhanced transactions over the larger dataset. In this Supervised learning algorithms, which include Decision Trees, Random Forests, Support Vector

Machines (SVM), and Gradient Boosting, can achieve the most accuracy when provided with a considerable and balanced marked dataset. Despite this, they rely significantly on the accessibility of marked datasets and might be enhanced appropriately to previous fraudulent signs or pattern formation. In this Unsupervised learning is superior when marked datasets in a limited manner, However, it can detect exceptions or anomalies unless explicit labels. By using these hybrid methodologies whereas advantages of both supervised and unsupervised learning techniques have been exhibited, employing unsupervised algorithms to capture most anomalies and in due course using supervised classifiers to verify their fraudulent concepts. This model utilized a balanced methodology, improving detection concepts while reducing incorrect positives. By using these Deep learning methodologies previously has been noted to enhance fraudulent detection by discriminating complicated patterns and correlations from larger datasets. This methodology is such that Autoencoders are exhibited for anomaly detections, mainly advantageous for higher-level dimensional datasets. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks are empowered to enhance linear input and identify patterns inside this transaction mechanism and its sequences. By having this Convolutional Neural Networks, mainly imposed for image detection, have been changed for fraudulent detection by possessing dimensional hierarchies in transaction attributes. Despite this, previous patterns might be computationally enhancing and make necessary considerably larger datasets for good comprehension. In this Fraudulent detection is riskier and more challenging such as larger dataset imbalance, improving fraudulent schemes and models, compressibility, and the made more necessary for real-time system detection. Graph-based models and federated learning have been plotted as more advanced methodologies for tainting relationships among attributes in transactional datasets. Future research should focus on improving more flexible, explicable, and scalable fraudulent detection systems as fraudsters progress. As criminals progress, future research must focus keenly on improving many variable, interpretable, and scalable fraudulent detection systems in enhanced mode.

Table 1: *Literature Survey*

Author and Publication	**Techniques Used**	**Dataset**	**Highlights**	**Performance Metrics**	**Remarks**
F. Itoo, M. Meenakshi, and S. Singh(2021) ***Int. J. Inf. Technology***	Random Undersampling for balancing the dataset. logistic regression, Naive Bayes, and K-nearest neighbor	Credit Card Dataset 284,807 transactions with 492 fraud cases	3 different ratios of datasets and an undersampling method to deal with the problem of data imbalance	Ratio A[50:50] LR -91.2% Ratio B[34:66]: LR-92.3% Ratio C[25:75]: LR – 95.9%	Some information could be lost and new resampling methods could be devised to achieve optimal results.
Morteza Rakhshaninejad, Mohammad Fathian, Babak Amiri, Navid Yazdanjue(2022) ***The Computer Journal***	Undersampling for data imbalance	Credit Card Dataset 284,807 transactions with 492 fraud cases	Random Forest classifier for selecting essential features Create ensemble from ML classifiers	Accuracy:99.97% Precision:87.78%, Recall:97.70%, F1-score:92.21% F2-score:95.634%	Computational costs were high

3. METHODOLOGY

It is much more essential to spot erroneous activities such as anomalous characteristics in the larger volume of payment transaction dataset is most needful for finding real-time errors such as fraudulent and money laundering activities. So it is the most essential technique such as Deep learning models perform at improving extensively on larger datasets because of their capabilities to notice complicated graphs or patterns or signs. Methodologists for improving expansive payment transactions over larger data volume used higher level techniques as deep learning methodologies exhibit data delineation and feature engineering, model architecture, in which self-supervised learning and deep reinforcement learning have been intended. Empowered model implementation includes the incorporation of various models, self-supervised learning, and deep reinforcement learning models. Diagnosing unbalanced datasets comprises the development of raw datasets and the mounting of cost-sensitive learning. The implementation for a huge volume of data defining entails data preprocessing, data cleaning, normalization and scaling, batching and sharding, big data infrastructure, data streaming, and again exhibit model training at scale, hyperparameter enhancement, and federated learning.

In these anomaly detection methods, it is essential to entertain supervised learning concepts, unsupervised learning techniques, and semi-supervised learning models. Here it's appraising to enhance the performance constraints and criteria points including scales and distinction, recall, F1 score, evolve the area based on the criteria Curve distinct and real-time system performance, recall, and F1 score points has shown complicated measures for calculating the model's abilities to sharply detect exact abnormalities while reducing false positives. This model is a higher-level measure criterion mapped to traditional accuracy, and reaction time is most needed for online anomaly detection in widespread environments for flawless implementation.

Detecting anomalous strategies along with real-time errors and gathering applicable possession from payment transaction datasets which makes it necessary for the usual scheme that is related to data science techniques with highlighted context intelligence. Eventually, we can check the extensive progress which might be followed for this task.

3.1. Data Acquisition and Preprocessing

In need to diagnose payment transaction data, which always encompasses criteria such as payment transact ID, date of transaction, amount paid, merchant ID, username and user ID, and type of payment that has been done

Preparation of Dataset: Redefine the larger dataset by pointing out missing values, detaching duplicate values accumulated, and normalizing the formats of the dataset. Mahe good Standards of numerical values and encrypt categorical variables such as vendor ID in which provisioning methods such as one-hot encryption.

3.2. Features Extraction

In this feature extraction, the selection of features available might be noted anomalous characteristic.

Features of statistical: Imply basic statistics such as payment or data transaction frequency, the difference in transaction dataset, and primary or varied design along with users data, merchants data, or locations preferably.

User Analysis on Behaviour context: In Contrast with usual transactions with a user's previous transaction dataset activities. Randomly might be an encounter with a basic payment transaction accumulation, typical locations, or payment strategy.

Features of Variances: Using a subset to implement attributes such as the hour of the day, day of the week, or random trends in exact transactions. These fraudulent transactions might occur inside basic intervals or entertain apparent varied patterns.

3.3 Anomaly Detection Characteristics:

To have an identification of anomalies over transaction data or schemes. Functions like z-points or distance-level-based approaches can be allotted in identifying resultant data. To check the status of anomalies in usual transaction patterns for some users or vendors.

Features of Aggregate: We need to generate aggregated concepts such as all transaction data, whole transaction counts per user ID or vendor over specified timestamps, and averages.

Inferred Attributes: Velocity of Transaction: Does Follow the prompts of more payment transactions transpired inside enough time series. Grand velocity might be important for fraudulent activities.

Typical Anomalies: Estimated distances between transaction pages and detected advances larger distances within a specified time series. Anomaly Detection on technical detect an anomalous scheme joint with issues, the following strategies might be encoded.

Rule-Based Compute: encounter average means for transaction characteristics and identify those datasets that added value to the methods.

Unsupervised Learning: In this classification, analogous transactions will be intended to available attributes such as the amount of data, preferred location, and user attempt. Payment transactions that do not comply with any cluster might be defected as potentially excessive.
Isolated Forests: A decision-tree-based method that has detected anomalies in the dataset by recursive categorization. This is potent for higher-level dimensional datasets such as transaction-based datasets.
Autoencoders: A neural network technical design to rebuild basic transactions. Transactions will evolve shortage on recreation by the autoencoder techniques are detected as anomalies.
Supervised Learning: When marked data is utilized, supervised models can be improved to predict anomalous transactions.

By using Random Forests, Gradient Boosting Machines, and XGBoost are incorporated the models which are capable of encountering complicated meeting axes among features and providing prioritization grades for feature selection.
Neural Networks: this deep learning model might specify intricate patterns and be connected in transaction datasets, specifically utilized for suitable attacks such as money laundering.
Surveillance and Identification: Data Streaming: Building real-time system checks up for anomaly detection using streaming rely This utilization of on-demand dataset evaluates, detecting anomalies as transactions occurred. Sliding Techniques Used a sliding window method on payment transaction sequence to detect anomalous graphs or signs or patterns completely both tiny and longer time series.
Post-Processing and Alerting Averages: Encompass the averages dedicated from the trained model resultant dataset to active signals on the detection of an anomalous payment transaction.
Anomalies Consolidation: When enormous anomalies raised inside a clear time series accumulate into one instance to decrease alert errors in real real-time system. Feedback Updating: Decisively redefining models to revert feedback from detected bugs or false positives. Incorporated human end evaluation into a feedback description mechanism to increase added detection precision for real time systems.
Adversarial Detection: Recognise instances where assailants may attempt to circumvent detection by intentionally imitating typical behavior. Methods such as adversarial machine learning can enhance detection models.

3.4 Dataset to detect a malicious attack

The Australian Institute of Cyber Security's Cyber Range Laboratory has analyzed 175,341 malware detection reports in the UNSW-NB 15 database, which includes 49 features and three data types: categorical, binary, and numerical.

Table.2 Number Of Records In Training And Testing Subsets For Each Class

Classes	**Training Subset**	**Testing Subset**
Normal	56,000	37,000
Analysis	2000	677
Backdoor	1746	583
DoS	12,264	4,089
Exploits	33,393	11,132
Fuzzers	18,184	6,062
Generic	40,000	18,871
Reconnaissance	10,491	3,496
Shellcode	1,133	378
Worms	130	44
Total Number of Records	**175,341**	**82,332**

3.5 Requirements:

- ➢ The dataset is required to retrieve the relevant features related to malicious attacks and transaction data.
- ➢ To process the data set, need data preprocessing
    - o Transforms real-world datasets into understandable formats.
    - o Essential for identifying patterns in transmitted data.
    - o Essential for information retrieval efforts, including malicious detection.

- Check for Class Imbalance Problem
- Find suitable data augmentation to solve class imbalance
- Apply feature extraction technique in the processed dataset to extract the relevant features.

3.5 Data pre-processing

This process converts real-world datasets into understandable formats, aids in identifying patterns, aids in information retrieval, and includes preprocessing for accurate findings.

Data Pre-processing on the dataset follows the following steps

3.5.1. Apply clamping

3.5.2. Apply the log function to nearly all numeric, since they are all mostly skewed to the right

3.5.3 Reduce the labels in categorical features

3.5.1 Apply clamping:

Clamping is a technique used in data processing, computer graphics, machine learning, and numerical computations to confine a value within a specified range, ensuring it doesn't exceed or fall below a specified limit.

- **Mathematics & Numerical Computation**: Clamping is a mathematical operation used to maintain a value within a specific boundary. If the value exceeds the upper or lower bound, it is set to the closest boundary value. For instance, if a value is clamped between 0 and 1, the operation would be: y=max(min(x,1),0).
- **Computer Graphics**: Clamping is a common technique in computer graphics for handling colors or coordinates, such as RGB color values between 0 and 255 or 0.0 and 1.0 in normalized systems, and for rendering 3D objects, vertex coordinates or texture mapping to maintain a specific viewport or texture space.
- **Machine Learning**:Clamping is a technique used in machine learning to normalize or bound feature values, particularly in dealing with outliers. It is used in preprocessing to prevent the extreme impact of outliers on model performance and in activation functions like ReLU6 in neural networks to prevent large gradients.

3.6 Use Cases in Data Processing:

Clamping is a technique used to prevent overflow or underflow errors by ensuring values stay within a safe range. It can also be used for noise reduction in signal processing or data preprocessing. In machine learning models, features are often clamped to a reasonable range to avoid skewing predictions, especially if there are outliers.

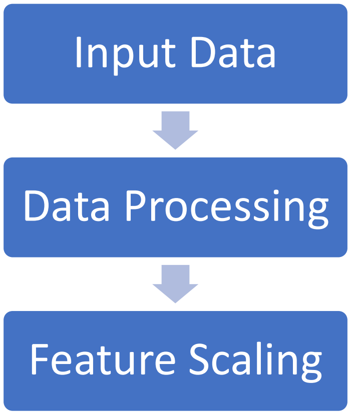


Fig.1 Dataset Distributionapply The Log Function To Nearly All Numeric, Since They Are All Mostly Skewed To The Right

The dataset's right-skewed numerical data indicates a high number of low values and a low number of low values. A logarithmic transformation can normalize the distribution and reduce skewness, improving the data's suitability for machine learning models.Right-skewed data with a long tail of high values can distort statistical analysis and model performance. The log function compresses the data range, reducing extreme values' impact while maintaining relative order. Applying log transformation results in a more symmetrical, bell-shaped distribution, which models like linear regression and logistic regression can better handle.

3.7 How to Apply Log Transformation:

Mathematical Formula:

The logarithm of a positive number is expressed as y=log(x), where log may refer to the natural logarithm (base e) or logarithm base 10. To address zeros and negative values, incorporate a small constant, such as 1, to all values before executing the logarithmic transformation. This guarantees that even null values are converted suitably. Log transformation is appropriate for data exhibiting a positively skewed distribution and encompasses numeric values with a broad range, such as financial data or non-negative continuous variables.

3.8 Reduce the labels in categorical features

To minimize the labels in the categorical features of the UNSW-NB15 dataset, widely known for network intrusion detection, we concentrate on categorical variables within the dataset and apply relevant feature-reduction techniques.

Steps to Minimize Labels in Categorical Features in UNSW-NB15 1. Identify Categorical Features:

The following are the categorical features within the UNSW-NB15 dataset.
1. Protocol (for example, TCP, UDP)

- service: Protocol used at the destination (for example, HTTP, FTP)
- state: Connection condition (e.g., FIN, CON) These features usually have many unique values.
- Methods of Label Reduction: Based on the nature of these categorical feature type, several techniques like rare category aggregation, frequency encoding, and domain-specific categorization are used.

2. Rare Category Aggregation (for example service and state)
Often, the service characteristic encompasses many low-frequency services. We may categorize infrequent categories as one label, "Other," to enhance model performance better.

3. Frequency Encoding for prototype
The proto feature depicts the protocol (TCP, UDP, etc.). Instead of using a one-hot encoding that is of high dimension, we can use frequency encoding to replace protocols by their frequency of appearance.
4. Grouping Based on Domain Knowledge for Services:
Leverage domain knowledge to group services at a high level (eg, HTTP-related services, file transfer services). This is very useful when you have a great understanding of how different services interact.
5. One-Hot Encoding with Limited Labels (Top N Categories):
Apply One-Hot Encoding to proto, service, or state but the encoding is limited to top categories. Less frequent categories will be rolled up into "Other" to avoid a large number of dummy variables.

6.Target Encoding:
The method uses a supervised learning task, say classifying different types of attack, so that target encoding is possible by replacing the categorical labels with the mean target value associated with each category. It is most useful for attributes with many categories, like services.

3.9 *Imbalance Problem in Malicious Attack Dataset*
The imbalance issue in a dataset concerning malicious assaults pertains to the disparity in which the quantity of malicious attack instances (positive class) is markedly inferior to that of normal or benign instances (negative class). This imbalance can significantly impair the performance of machine learning models, particularly in classification tasks, as the model is inclined to favor the majority class and may exhibit suboptimal performance in identifying the minority class (malicious assaults).
Identify the Target Variable (Class)

- Normal as Class 0
- Attack as Class 1

Check for Class Imbalance
For the training set:

- Class 1 has 45,332 instances. Class 0 has 37,000 instances.
- Imbalance Ratio for Training Set: 1.225

For the testing set:

- Class 1 has 119,341 instances. Class 0 has 56,000 instances.
- Imbalance Ratio for Testing Set: 2.131

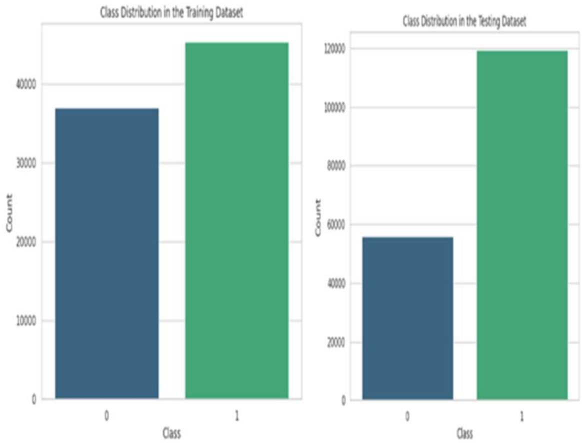


Fig 2. Imbalance Dataset Hierarchical network feature extraction for Malicious attack

The tool converts data into numerical attributes and divides networks into layers for specific purposes, selecting the best attributes for each layer, including datetime, host, src, proto, type, country, locale, postal code, latitude, and

oversampling, undersampling, or class-weighted models can be used.

- Synthetic Minority Over-sampling Technique (SMOTE) to balance

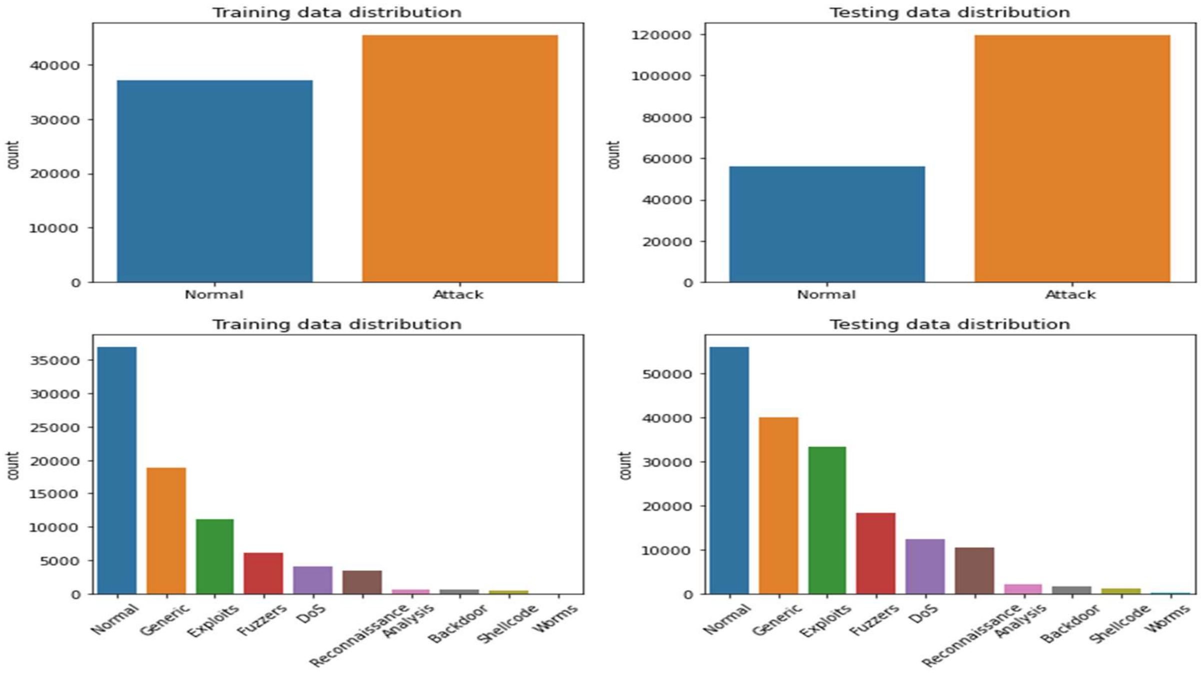


Fig 3. Interpreting the Imbalance Ratios

Ratios indicate class distribution disparity, with higher ratios indicating imbalances. Imbalances can affect model performance and should be considered during training and evaluation using oversampling, undersampling, and specialized algorithms.

Class 1 vs Class 0 Ratio in Dataset

- **Percentage Interpretation:** Class 1 instances are approximately 1.22% of Class 0 instances.
- **Proportion Interpretation**: For every 100 instances of Class 0, there are approximately 122 instances of Class 1.
- **Quantitative Interpretation**: If 100 instances of Class 0, there would be around 122 instances of Class 1.
- **More Prevalent**: Class 1 occurs 1.22 times as frequently as Class 0 within the dataset.

SMOTE Oversampling for Class Imbalance Problem

- To address the class imbalance in the dataset, various techniques like

The dataset Class Distribution after SMOTE

Counter({0: 45332, 1: 45332})

- Dataset trained with deep learning models to find the effectiveness of the augmentation technique.

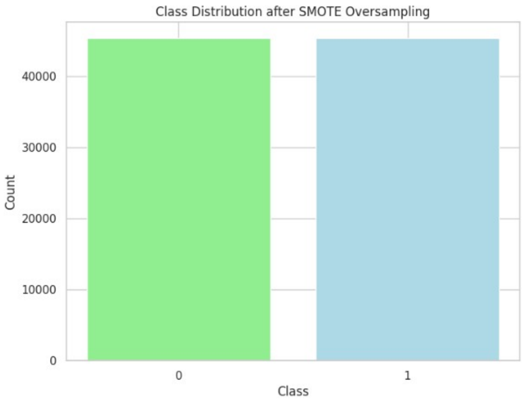


Fig 4. SMOTE Oversampling

Random Undersampling Technique to balance the dataset.

- Class Distribution after Random Undersampling:

Counter({0: 37000, 1: 37000})

- Dataset trained with deep learning models to find the effectiveness of the augmentation technique.

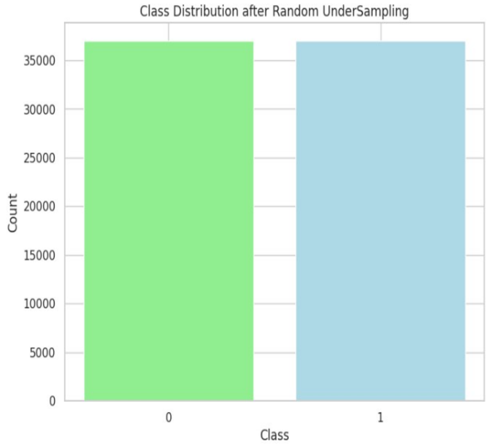


Fig 5. *Random Undersampling*

Table 3: *Smote Sampling*

Dataset	Model	Class 0			Class 1			Accuracy
		Precision	Recall	F1 Score	Precision	Recall	F1 Score	
Random Oversampled UNSW-NB 15	RNN	0.71	0.87	0.78	0.84	0.65	0.73	76
	CNN	0.72	0.88	0.80	0.85	0.67	0.75	80
	ResNet	0.80	0.90	0.84	0.90	0.74	0.82	86

Table 4: *Random Undersampling*

Data set	Model	Class 0			Class 1			Accuracy (%)
		Precision	Recall	F1 Score	Precision	Recall	F1 Score	
Random undersampled UNSW-NB 15	RNN	0.72	0.86	0.78	0.83	0.67	0.74	77
	CNN	0.73	0.89	0.82	0.85	0.69	0.78	82
	ResNet	0.81	0.91	0.86	0.92	0.77	0.85	90

4. Summary Of Augmentation Technique

- ➢ In Undersampling, Both precision and recall for Class 1 have improved, resulting in a better overall F1 score and accuracy compared to oversampling.

- ➢ The model performs well in identifying both Class 0 and Class 1 instances but slightly better for Class

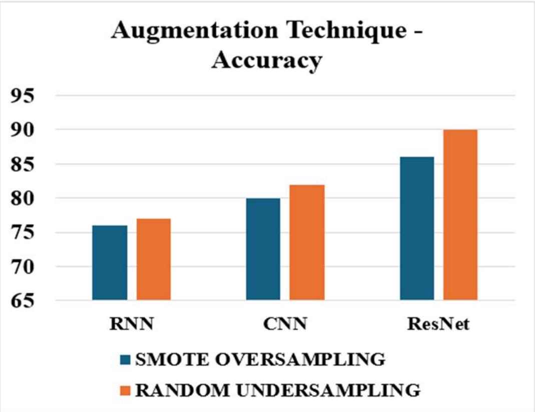
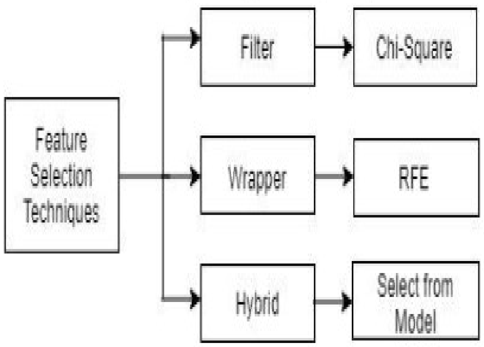


Fig6. *Augmentation Technique Comparison On Accuracy And F1-Score*

4.1 Feature Selection Techniques for Malicious Attack

Wrapper methods are a type of feature selection method using a predictive model in conjunction with the process of selection to provide the best subset possible. The more popular wrapper methods include forward selection, backward elimination, recursive feature elimination (RFE), stepwise selection, and genetic algorithms. The forward selection approach starts from an empty set of features and iteratively includes features while the backward elimination, on the other hand, progressively deletes the least important feature during an iteration cycle. Although computationally intensive, recursive feature elimination works iteratively on removing the least informative features according to their weights. Stepwise selection is comparable to forward and backward selection but still computationally demanding. Genetic algorithms apply optimization techniques developed based on natural selection to find the best subset of features; however, they are highly computationally demanding and require parameter tuning. Wrapper methods are used with models that require large performance levels along with high computational power.



Forward selection begins by starting with an empty feature set and successively adds features, whereas backward elimination always removes the least important feature in each iteration. The recursive feature elimination method involves the removal of the least significant features based on their weights but is computationally expensive. Stepwise selection is a combination of both forward and backward selection methods but is still a computationally expensive method. Genetic algorithms base optimization techniques in finding the best subset of attributes, but such methods are pretty intensive in terms of process resources and parameter tuning. Wrapper approaches are best applied in models of involved learning when they require significant performance and commitment of computational resources.

4.1.1 *SelectKBest Method*

The SelectKBest technique is a method of frequently used feature selection strategies that falls in the filter methods group. Its primary strategy is to determine the most important features based on their significance to the target variable through the evaluation of statistical metrics. This approach can be very beneficial for transactional data analysis, especially in cases where the most important features to predict a target variable should be discovered-for example, in fraud detection, customer attrition, or credit evaluation.

The methodology for the implementation of SelectKBest on transactional data is delineated.

Procedure for Implementing SelectKBest: Prepare transactional information

Structure the transactional data in a feature matrix X with independent variables like amount, location, date, and so on, and then a goal vector y for the dependent variable, that is, fraud or non-fraud classification. Encode all categorical features appropriately using techniques like one-hot encoding. Determine a Scoring Function: One of the steps in SelectKBest is to select an appropriate scoring function. The nature of the scoring function required will depend on the specific character of the attributes of the task under consideration.

4.1.2 *Chi-squared (Chi2) test*

The Chi-squared (Chi2) test is a statistical technique employed to assess the independence of two variables, frequently utilized in feature selection for classification tasks. The Chi2 approach aids in selecting the most pertinent categorical categories for predicting a target variable, such as fraud detection, inside transaction data.

The Functionality of the Chi-squared Method:

Transaction data, wherein the independent variables (features) are categorical or can be discretized (e.g., transaction amount ranges, location, time intervals).

Output: A ranking of the features according to their Chi-squared statistic, reflecting the strength of their association with the target variable (e.g., fraudulent versus non-fraudulent transactions).It calculates the dependence of each feature on the target variable. The better the Chi2, the stronger the relationship between the feature and the target, meaning it can improve the class prediction by using this feature.

Assumptions Using Chi-Squared Qualitative data: Characteristics must either be categorical or continuous variables-for example, transaction amounts are to be measured and classified into discrete categories or intervals.

Non-negative integers:

The Chi-squared test is sensitive to negative results; so ensure your data is prepared appropriately. The process to carry out the Chi-squared Algorithm on Transaction Data:
Prepare your data: Ensure the data you have for transactions contains categorical features or discretized continuous variables and also a target variable that tags the class, such as fraud or non-fraud.
Discretization (if needed): For continuous variables such as transaction amount and duration, you will discretize them into intervals or classes; like low, medium, and high transaction amounts, of course.
Use the SelectKBest class with the chi2 scoring algorithm to select the most important features.

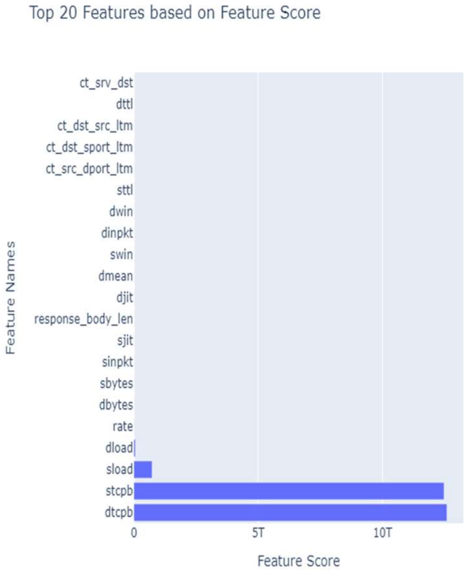


Fig 7. *Feature Selection*

4.2 Hierarchical network feature extraction

Feature extraction method to represent transactional interactions between entities like a consumer, a merchant or an account. Hierarchical network structure is composed of three steps: first, the construction of a transaction network; second, the forming of a hierarchy of a network; third, extracting features at various hierarchical levels in the network. The features are node-level attributes such as the individual transactions, community-level characteristics of the groups of nodes, global network-level metrics for instance, network density, average path length and so on, clustering coefficient, modularity, and hierarchical embeddings.

Temporal hierarchical feature extraction detects changes over time, including some of the temporal centrality measures, community evolution, and time-window characteristics. These characteristics can be incorporated for fraudulent detection, client categorization, and risk measurement. Node-level anomalies, group-level patterns, or global network variation might be shown as fraudulent. Model building includes noticed hierarchical characteristics to be used in machine learning models and feature engineering applied for a combination of different levels of its characteristics. This could be a way to further expand insights into transactional interactions while improving predictability as it can be used for detection purposes, such as fraud or risk classification and customer segmentation. The process may also be applied to anomaly detection, client categorization, and centrality-based, pattern-based, or position-based risk assessment. The strategy transforms raw data into numerical attributes, organizes networks into layers, and determines the best features for each layer. It also promotes identification by constraining features and combines non-negativity constraints with deep learning frameworks. Data is saved in Dropbox for file sharing, while sparsity constraints are a very popular approach to feature extraction.

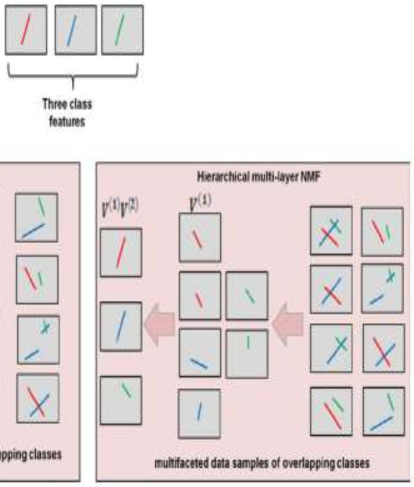

Three class features
Single-layer NMF
V
multifaceted data samples of overlapping classes
Hierarchical multi-layer NMF
$V^{(1)}V^{(2)}$
$V^{(1)}$
multifaceted data samples of overlapping classes

Fig 8. *Multi-Layer Hierarchical Network*

The strategy transforms raw data into numerical attributes, organizes networks into layers, and identifies the optimal attributes for each layer. It enhances identification with constrained features and integrates non-negativity constraints with deep learning frameworks. Data is saved in Dropbox for file sharing, and sparsity constraints are a prevalent technique for feature extraction.The cost function is seen in the equation

$$C = \frac{1}{2}\|Q - VG\|^2 = \frac{1}{2}\sum_{m=1}^{M}\sum_{n=1}^{N}\left(Q_{mn} - \sum_{r=1}^{R} V_{mr}G_{rn}\right)^2.$$

A multiplicative update rule is used to adjust V and G inside the rounds, as in the below equation

$$V_{mr} \leftarrow V_{mr}\frac{(QG^T)_{mr}}{(VGG^T)_{mr}}, G_{mr} \leftarrow G_{mr}\frac{(V^TQ)_{rn}}{(V^TVG)_{rn}}$$

- a sparsity matrix R is placed as a restriction on a typical single-layer NMF

$$R = (1-\theta)eye(g) + \frac{\theta}{g}ones(g)$$

g=attributes, θ=sparsity factor ranging between 0 and 1, $eye(g)$=identity matrix with (gxg), $ones(g)$=matrix with each element of 1s

Matrix sparsity loses when multiplied by R.

• Degradation increases near 1.

• Alternative change: divide R by G, G=RG.

• Decreased G sparsity causes V to become sparse.

Every layer's output is converted to $D^{(a)}$ before being used as an input next

$D^{(a)}$ Calculated by

$$D_{rn}^{(a)} = f\left(G_{rn}^{(a)}\right)$$

$G^{(a)}$=output of layer, f(.)=non-linear function, and a=layer's index with a=1, 2,…,

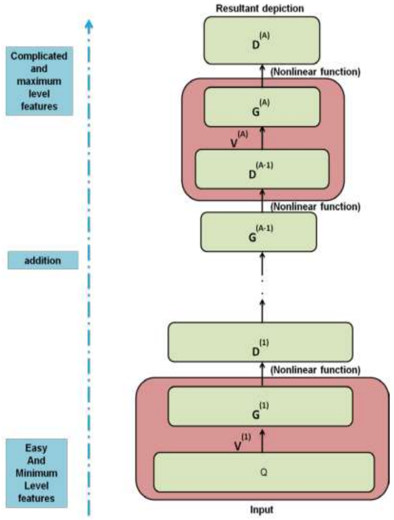


Fig 9. *Hierarchical Network Feature Extraction Flow*

Data Decomposition and Classification

nsNMF is used to dissolve D(a) into V(a+1) and G(a+1), achieving g=G.

• Next-layer decomposition of D(a) reveals how different layers' features combine.

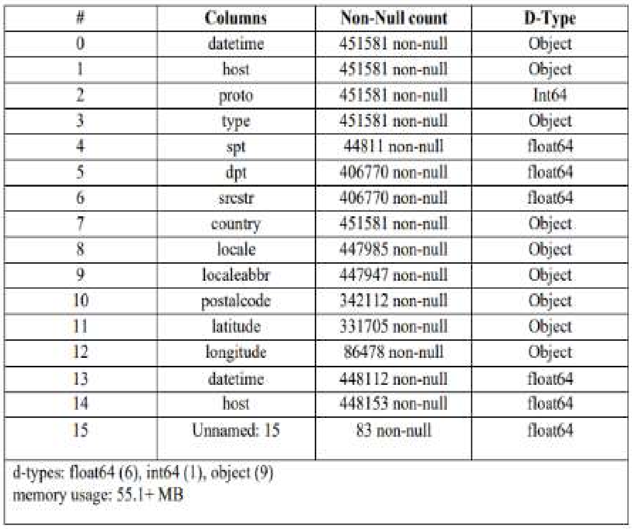
• Complex features are created by identifying which features from one layer are coupled with others in the next.

• Data description D(A) is achieved after training up to the last layer.

• Data is stored in Dropbox for easy file storage and synchronization.

• Dropbox connections allow users to exchange data files without transferring large documents.

• Features are collected from Dropbox to identify malicious attacks in the classification stage.



#	Columns	Non-Null count	D-Type
0	datetime	451581 non-null	Object
1	host	451581 non-null	Object
2	proto	451581 non-null	Int64
3	type	451581 non-null	Object
4	spt	44811 non-null	float64
5	dpt	406770 non-null	float64
6	srcstr	406770 non-null	float64
7	country	451581 non-null	Object
8	locale	447985 non-null	Object
9	localeabbr	447947 non-null	Object
10	postalcode	342112 non-null	Object
11	latitude	331705 non-null	Object
12	longitude	86478 non-null	Object
13	datetime	448112 non-null	float64
14	host	448153 non-null	float64
15	Unnamed: 15	83 non-null	float64

d-types: float64 (6), int64 (1), object (9)
memory usage: 55.1+ MB

Fig 10. *Feature extraction result*

Table 4. Model Performance across Different Attack types

Attack Type	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Detection Rate (%)**	**False Positive Rate (%)**
Phishing	85	88	86	90	7
Malware	88	85	86	89	5
Denial of Service	82	80	81	84	9
Account	87	84	85	88	6
Hijacking	88	91	94	86	9

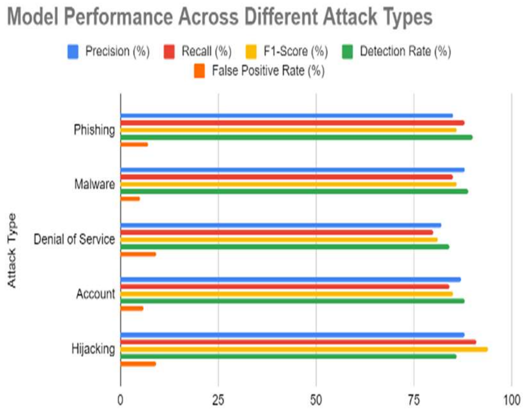


Fig 11. *Model Performance Feature Engineering on Model Accuracy*

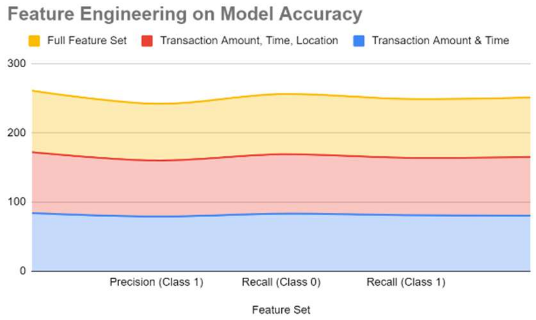


Fig12. Feature Engineering

Table 4: Real-Time Detection Metrics Across Transaction Types

Transaction Type	**Average Anomaly Score**	**Detection Rate (%)**	**False Positives (%)**
Credit Card	76	90	5
Bank Transfer	71	88	7
Mobile Payment	78	85	8

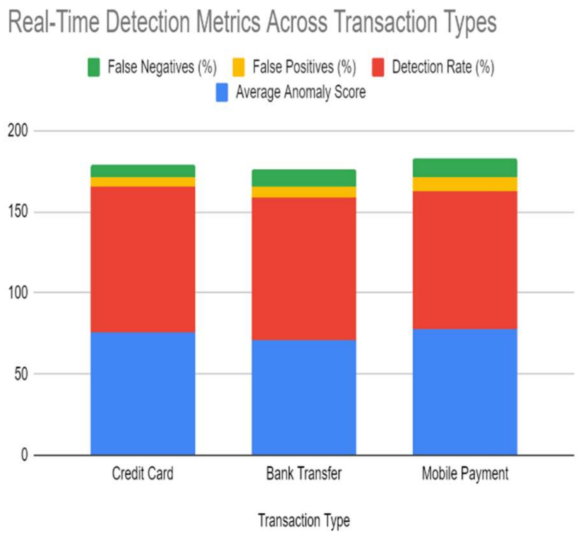


Fig13 .Real –Time detection Metrics

5. CONCLUSION AND FUTURE WORK:

Even though the network environments are highly fluid and complicated, the use of hierarchical networks in extraction of features for abnormal incidences detection within transaction data offers a workable solution and especially assists in detecting real attacks when other methods fail. This method uses hierarchical network features focusing on the interaction of entities both locally and globally and this theory aids the detection of even the least disturbing anomalies.

6. RESULTS AND CONTRIBUTIONS OF THIS WORK CAN BE SUMMED UP IN THE FOLLOWING:

Hierarchical Network Feature Extraction: Feature extraction technique is possible relating transaction data features in various levels. This brings to light concealed networks in transaction networks and hence increases the prospects of detection of some rotten ideologies.

Enhanced Abnormality Detection: This augurs well with abnormality detection as it incorporates structural features and transactional features providing a complete image of the network improving the separation of normal and abnormal actions.

Scalability and Flexibility: The hierarchical strategy presents scalability and flexibility across various transaction networks thus allowing the model to be able to generalise well across different datasets and domains.

Optimized Application of Machine Learning Algorithms: The enhanced detection ability resulting in reduced false alarm rate limiting the occurrence of innocent parties and minimizing losses due to missed detection of the malicious transactions is achieved with senior feature extraction approach and machine learning, especially the ensemble techniques. Prospective Endeavours The hierarchical network extraction features method for spotting discrepancies in transaction data is highly revealing other than offering the outlook for further explorations and enhancements.

REFERENCES

[1] Arroyabe, M. F., Arranz, C. F., De Arroyabe, I. F., & De Arroyabe, J. C. F. (2024). Revealing the Realities of Cybercrime in Small and Medium Enterprises: Understanding Fear and Taxonomic Perspectives. *Computers & Security*, *141*, 103826. https://doi.org/10.1016/j.cose.2024.103826
[2] Landauer, M., Onder, S., Skopik, F., & Wurzenberger, M. (2023). Deep learning for anomaly detection in log data: A survey. *Machine Learning With Applications*, *12*, 100470. https://doi.org/10.1016/j.mlwa.2023.100470

[3] Mumuni, A., & Mumuni, F. (2024). Automated data processing and feature engineering for deep learning and big data applications: a survey. *Journal of Information and Intelligence*. https://doi.org/10.1016/j.jiixd.2024.01.002

[4] Bloomenthal, A. (2024, September 30). *Credit Card: What It Is, How It Works, and How to Get One*. Investopedia. https://www.investopedia.com/terms/c/creditcard.asp

[5] Fraudcom International. (2024, March 4). *What is fraud detection and why is it needed?* Fraud.com. https://www.fraud.com/post/fraud-detection

[6] Salam, A., Ullah, F., Amin, F., & Abrar, M. (2023). Deep Learning Techniques for Web-Based Attack Detection in Industry 5.0: A Novel Approach. *Technologies*, *11*(4), 107. https://doi.org/10.3390/technologies11040107

[7] Salam, A., Ullah, F., Amin, F., & Abrar, M. (2023). Deep Learning Techniques for Web-Based Attack Detection in Industry 5.0: A Novel Approach. *Technologies*, *11*(4), 107. https://doi.org/10.3390/technologies11040107