Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 10 Issue 38 (March 2025) PP. 181-201
DOI: 10.35631/JISTM.1038012

JOURNAL OF INFORMATION SYSTEM AND TECHNOLOGY MANAGEMENT (JISTM)
www.jistm.com

FRAUDULENT CREDIT CARD TRANSACTION DETECTION USING LOGISTIC REGRESSION

Syahir Aiman Shahrul Nadzman[1], Gloria Jennis Tan[2*], Tan Chi Wee[3], Ung Ling Ling[4], Norziana Yahya[5]

[1] College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Kuala Terengganu, Malaysia
Email: syahir.aiman009@gmail.com
[2] College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Kuala Terengganu, Malaysia
Email: gloria@uitm.edu.my
[3] Department of Computer Science and Embedded System, Tunku Abdul Rahman University of Management and Technology, Kuala Lumpur, Malaysia
Email: chiwee@tarc.edu.my
[4] College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Kota Kinabalu, Malaysia
Email: ungli720@uitm.edu.my
[5] College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Arau, Malaysia
Email: norzianayahya@uitm.edu.my
[*] Corresponding Author

Article Info:

Article history:
Received date: 14.01.2025
Revised date: 23.01.2025
Accepted date: 27.02.2025
Published date: 20.03.2025

To cite this document:

Nadzman, S. A. S., Tan, G. J., Tan. C. W., Ung, L. L., & Yahya, N. (2025). Fraudulent Credit Card Transaction Detection Using Logistic Regression. *Journal of Information System and Technology Management, 10* (38), 181-201.

DOI: 10.35631/JISTM.1038012

Abstract:

Credit card fraud poses a significant threat to financial institutions and individuals, leading to substantial losses and undermining trust in digital payments. This study aimed to identify fraudulent transactions using a logistic regression-based machine learning model, develop a fraud detection prototype, and evaluate its accuracy using Precision-Recall Area Under the Curve (PR AUC). The methodology included three phases: Preliminary, Design, and Evaluation. In the Preliminary Phase, a literature review identified research gaps, and the September 2013 European credit card fraud dataset from Kaggle was preprocessed using robust scaling. The Design Phase involved constructing system architecture, creating flowcharts, designing a user interface, and developing logistic regression pseudocode. During the Evaluation Phase, the study balanced the dataset using undersampling, conducted 5-fold cross-validation, and split the data into training, testing, and validation sets in a 70:30 ratio. The logistic regression model was trained and evaluated using precision, recall, F1-score, and PR-AUC. The model achieved a PR-AUC score of 99.57% via the 10% validation set consisting of 52 fraud and 48 normal transactions, demonstrating high discriminatory power and reliability. The developed prototype enhances security and trust in digital payment systems. The use of robust scaling to normalise outliers, undersampling to balance the dataset, and comprehensive evaluation metrics

Journal of Information System and Technology Management
EISSN: 0128-1666
JISTM
Volume 10 Issue 38 (March 2025) PP. 181-201
DOI: 10.35631/JISTM.1038012

provide valuable insights for future research and practical applications in fraud detection systems. This study contributes to mitigating credit card fraud and improving financial transaction integrity. Future work should encourage collaboration between financial institutions, regulatory bodies, and researchers to share various types of anonymised transaction data and best practices, which could lead to more robust and generalisable models.

Keywords:

Logistic Regression, Fraud Detection, Transactions Detection, Credit Card

Introduction

The thorough investigation in the literature review has tremendous consequences for comprehending and improving systems to detect fraudulent credit card transactions. The comprehensive analysis of financial fraud detection systems clarifies the intricate terrain, offering insights into the many forms of financial fraud, the legal structure, and numerous detection methods.

The in-depth walkthrough of fraudulent credit card transaction detection using ML outlines a comprehensive process. It entails a step-by-step elucidation covering data collection and preprocessing, feature engineering, the employment of ML models, rule-based systems, model training, anomaly detection, human investigation, and report visualisation. This step suggests that a comprehensive strategy, incorporating both conventional and sophisticated approaches, is crucial for efficient fraud detection.

The scrutiny of conventional fraud detection systems underscores their role and limitations, prompting a reflection on the need for innovation. The exploration of AI and ML implementation signifies a disruptive shift towards adaptive and intelligent systems capable of learning and evolving in response to emerging fraudulent tactics.

The emphasis on specific ML techniques, such as logistic regression, random forest, and SVMs, implies that personalised approaches can substantially contribute to detecting advanced fraud. The simplicity and application of logistic regression are further emphasised by its real-world application in tackling different circumstances.

Finally, reviewing similar works provides a contextualised understanding of the existing literature landscape, highlighting the cumulative nature of research in this domain. The findings from this knowledge acquisition collectively pave the way for the following chapters, guiding the scholarly inquiry towards a nuanced and informed exploration of fraudulent credit card transaction detection systems.

Literature Review

The study conducted by Tressa et al. (2023) explores the application of the random forest algorithm to address credit card fraud. Advanced technologies, particularly machine learning algorithms, have emerged as powerful tools for addressing this issue. The paper emphasises the application of the random forest algorithm and reports a high accuracy of 100% on the test data. This algorithm excels in handling complex datasets, providing robustness and accuracy in fraud detection. However, the literature does not comprehensively compare the performance of different algorithms and methods, leaving room for further exploration. This literature

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 10 Issue 38 (March 2025) PP. 181-201
DOI: 10.35631/JISTM.1038012
review highlights the need for further research that reaches the implementation of various algorithms, including logistic regression, in credit card fraud detection. Future studies could explore ensemble methods that combine the strengths of different algorithms to enhance overall accuracy.

This literature review explores the paper by Aditi et al. (2022), focusing on utilizing logistic regression as a pivotal component in credit card fraud detection. The study employs a comprehensive approach using three advanced machine learning algorithms: decision tree, random forest, and logistic regression. While all three algorithms contribute to the overall efficacy of the fraud detection system, the focus narrows on the significance of logistic regression. With an accuracy rate of 95.55%, logistic regression emerges as a robust tool for identifying fraudulent transactions. However, applying these advanced machines learning techniques, including logistic regression, is not without challenges. The paper also acknowledges the potential privacy concerns of accessing sensitive customer data. As financial institutions grapple with the need for enhanced security, they must navigate the delicate balance between effective fraud detection and safeguarding customer privacy.

Jain et al. (2022) leverages the power of random forest, logistic regression, and AdaBoost algorithms to build a Credit Card Fraud Detection Web Application. The primary advantage highlighted in the literature is the capability of machine learning algorithms to detect real-time fraudulent transactions. The study demonstrates impressive accuracy rates, with random forest achieving 99.92%, logistic regression at 99.91%, and AdaBoost at 99.90%. These results underscore the effectiveness of machine learning in providing a proactive defense against fraudulent activities. While the high accuracy rates are commendable, a notable disadvantage in the literature is the complexity of machine learning algorithms. The paper acknowledges that interpreting these models, especially for complex algorithms like random forest and AdaBoost, can be challenging. Logistic regression, a more interpretable algorithm, is critical in bridging this interpretability gap.

The paper by Devika et al. (2022) has a similar title and contributes to the evolving landscape of fraud detection through the lens of logistic regression. As outlined in the study, the objective of developing a web application signifies a move towards practical implementation. The utilisation of the 2013 European credit card transactions dataset from Kaggle highlights a real-world context for the study. However, the literature reveals a common challenge: the imbalance of datasets. Imbalances, as acknowledged by the authors, can lead to biased models. This issue underscores the ongoing discourse in the literature regarding the significance of addressing data imbalances in credit card fraud detection models. The reported accuracy of 0.9905 for the logistic regression model indicates a high level of precision. Nevertheless, the literature underscores the need for a nuanced evaluation beyond accuracy, especially in imbalanced datasets. Metrics such as precision, recall, and F1-score become crucial for a more comprehensive assessment of model performance.

The study by Varmedja et al. (2019) aligns with the broader trend of leveraging machine learning to enhance fraud detection mechanisms. The study's strength lies in its comprehensive comparative analysis, encompassing a spectrum of machine learning algorithms. The research comprehensively analyses conventional approaches such as logistic regression, Naïve Bayes, and more recent techniques like random forest and multilayer perceptron. However, a notable limitation is the reliance on a single dataset, potentially constraining the generalisability of the

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 10 Issue 38 (March 2025) PP. 181-201
DOI: 10.35631/JISTM.1038012

findings to all credit card fraud detection scenarios. Among the algorithms assessed, logistic regression emerges with an accuracy of 97.46%. This finding is pivotal, considering logistic regression's simplicity and interpretability. The high accuracy underscores the algorithm's effectiveness in discriminating between legitimate and fraudulent transactions within the specific dataset used in the study.

Design and Development

This section presents a detailed overview of the system proposed for detecting fraudulent credit card transactions. It outlines the dataset employed during the preliminary phase of the project, the systematic approach to developing the prototype, including the design and implementation of the algorithm, and a thorough examination of the experiments conducted. Additionally, it highlights the performance of the system before and after key modifications were made, with a focus on applying logistic regression as the core analytical method.

Proposed Methodology

The system architecture, components, and their interactions are described to give a clear understanding of the prototype's design and functionality, as shown in Figure 1.

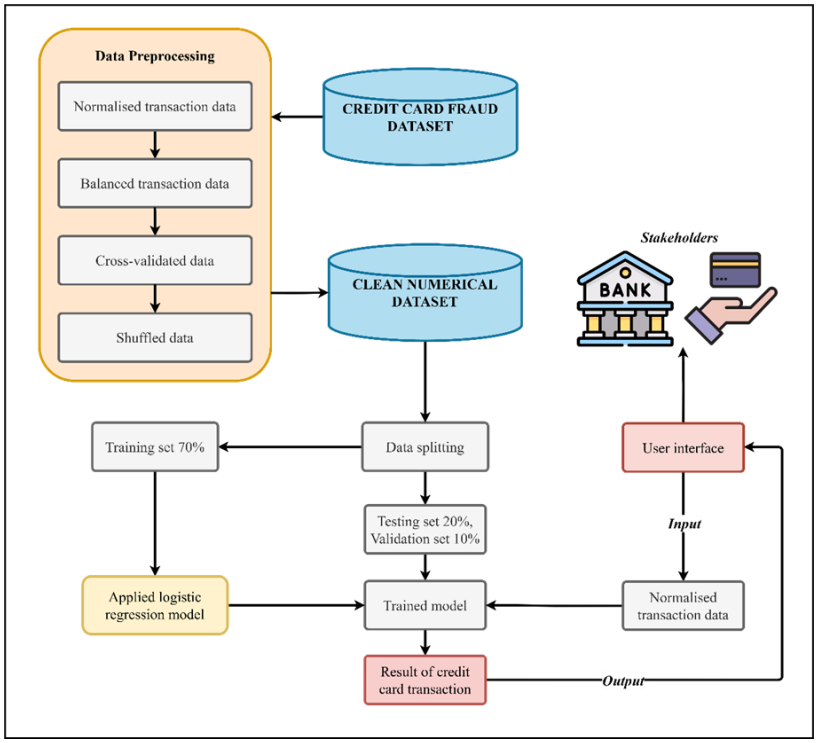


Figure 1: An Overview Of The Proposed Model Framework

Step 1: Preprocessing Data for Analysis

The model training began by uploading a CSV file containing historical credit card transaction data into the prototype system. The system pre-processed the data for better analysis. For instance, the Amount column underwent adjustment to enhance analysis capabilities. The Time column was normalised to a range between 0 and 1 for consistency across transactions.

Journal of Information System and Technology Management
EISSN: 0128-1666
Volume 10 Issue 38 (March 2025) PP. 181-201
DOI: 10.35631/JISTM.1038012

Next, the system created a balanced dataset to ensure fair representation of fraudulent and non-fraudulent transactions. It sampled an equal number of fraudulent and non-fraudulent transactions. Then, the balanced dataset was cross-validated and shuffled to prevent any inherent order bias.

Step 2: Splitting the Dataset

The cleaned and balanced numerical dataset was split into three distinct sets to facilitate model development and evaluation.

i. Training Set (70%): Used to train the logistic regression model.
ii. Testing Set (20%): Employed to assess the model's performance.
iii. Validation Set (10%): Used independently to validate the model's effectiveness.

Step 3: Training the Logistic Regression Model

Using the training set, the system trained a logistic regression model, optimising it to detect fraudulent transactions based on the dataset provided.

Step 4: Evaluating Model Performance

Performance on Testing Set

The model's performance was evaluated using the testing set. A classification report was generated showcasing accuracy and other key metrics relevant to fraud detection. Curves depicting precision versus recall and true positives versus false positives were plotted for interpretation. A confusion matrix illustrated correct and incorrect predictions made by the model.

Performance on Validation Set

The model's performance was further validated using the independent validation set. Similar to the testing set, a classification report highlighted accuracy and other crucial metrics. Curves and a confusion matrix gave insights into the model's consistency and reliability.

Step 5: Fraud Detection Results for Stakeholders

The stakeholders' inputted data will be normalised first to produce better predictions using the trained model. Upon successful evaluation, stakeholders were presented with comprehensive fraud detection results. Detected fraudulent transactions were displayed, providing stakeholders with transparency.

An option was provided to download the fraud data for further review or external reporting. A pie chart visually represents the proportion of fraudulent versus non-fraudulent transactions detected. A gauge showcased the percentage of fraudulent transactions detected, indicating the system's efficacy.

This ensured that stakeholders could confidently interpret the findings and take appropriate actions as necessary.

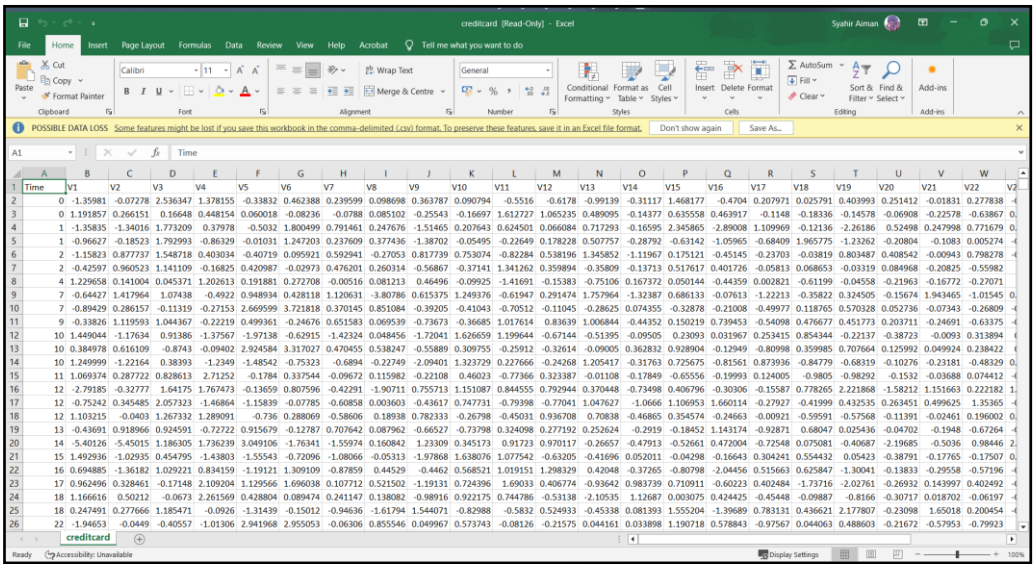
Journal of Information System and Technology Management
EISSN: 0128-1666
JISTM
Volume 10 Issue 38 (March 2025) PP. 181-201
DOI: 10.35631/JISTM.1038012

Preliminary Phase

The prototype utilised the Credit Card Fraud Detection dataset, available for download on Kaggle (2018). The information included is two-day transactions by European cardholders in September 2013.

The dataset has 31 numerical attributes. Due to the presence of financial information in certain input variables, a principal component analysis (PCA) transformation was conducted by the dataset provider to ensure the anonymity of the users and credit cardholders. Three of the provided features were not converted.

The Time feature displays the duration between the initial transaction and each subsequent transaction in the dataset. The dataset details are similar to the previous study in Figure 2.

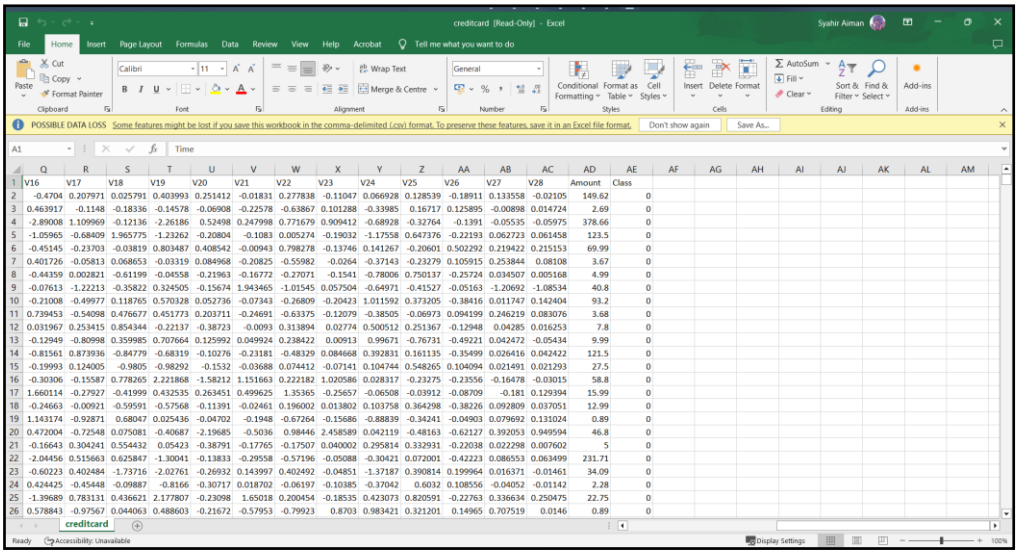


Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22
0	-1.35981	-0.07278	2.536347	1.378155	-0.33832	0.462388	0.239599	0.098698	0.363787	0.090794	-0.5516	-0.6178	-0.99139	-0.31117	1.468177	-0.4704	0.207971	0.025791	0.403993	0.251412	-0.01831	0.277838
0	1.191857	0.266151	0.16648	0.448154	0.060018	-0.08236	-0.0788	0.085102	-0.25543	-0.16697	1.612727	1.065235	0.489095	-0.14377	0.635558	0.463917	-0.1148	-0.18336	-0.14578	-0.06908	-0.22578	-0.63867
1	-1.35835	-1.34016	1.773209	0.37978	-0.5032	1.800499	0.791461	0.247676	-1.51465	0.207643	0.624501	0.066084	0.717293	-0.16595	2.345865	-2.89008	1.109969	-0.12136	-2.26186	0.52498	0.247998	0.771679
1	-0.96627	-0.18523	1.792993	-0.86329	-0.01031	1.247203	0.237609	0.377436	-1.38702	-0.05495	-0.22649	0.178228	0.507757	-0.28792	-0.63142	-1.05965	-0.68409	1.965775	-1.23262	-0.20804	-0.1083	0.005274
2	-1.15823	0.877737	1.548718	0.403034	-0.40719	0.095921	0.592941	-0.27053	0.817739	0.753074	-0.82284	0.538196	1.345852	-1.11967	0.175121	-0.45145	-0.23703	-0.03819	0.803487	0.408542	-0.00943	0.798278
2	-0.42597	0.960523	1.141109	-0.16825	0.420987	-0.02973	0.476201	0.260314	-0.56867	-0.37141	1.341262	0.359894	-0.35809	-0.13713	0.517617	0.401726	-0.05813	0.068653	-0.03319	0.084968	-0.20825	-0.55982
4	1.229658	0.141004	0.045371	1.202613	0.191881	0.272708	-0.00516	0.081213	0.46496	-0.09925	-1.41691	-0.15383	-0.75106	0.167372	0.050144	-0.44359	0.002821	-0.61199	-0.04558	-0.21963	-0.16772	-0.27071
7	-0.64427	1.417964	1.07438	-0.4922	0.948934	0.428118	1.120631	-3.80786	0.615375	1.249376	-0.61947	0.291474	1.757964	-1.32387	0.686133	-0.07613	-1.22213	-0.35822	0.324505	-0.15674	1.943465	-1.01545
7	-0.89429	0.286157	-0.11319	-0.27153	2.669599	3.721818	0.370145	0.851084	-0.39205	-0.41043	-0.70512	-0.11045	-0.28625	0.074355	-0.32878	-0.21008	-0.49977	0.118765	0.570328	0.052736	-0.07343	-0.26809
9	-0.33826	1.119593	1.044367	-0.22219	0.499361	-0.24676	0.651583	0.069539	-0.73673	-0.36685	1.017614	0.83639	1.006844	-0.44352	0.150219	0.739453	-0.54098	0.476677	0.451773	0.203711	-0.24691	-0.63375
10	1.449044	-1.17634	0.91386	-1.37567	-1.97138	-0.62915	-1.42324	0.048456	-1.72041	1.626659	1.199644	-0.67144	-0.51395	-0.09505	0.23093	0.031967	0.253415	0.854344	-0.22137	-0.38723	-0.0093	0.313894
10	0.384978	0.616109	-0.8743	-0.09402	2.924584	3.317027	0.470455	0.538247	-0.55889	0.309755	-0.25912	-0.32614	-0.09005	0.362832	0.928904	-0.12949	-0.80998	0.359985	0.707664	0.125992	0.049924	0.238422
10	1.249999	-1.22164	0.38393	-1.2349	-1.48542	-0.75323	-0.6894	-0.22749	-2.09401	1.323729	0.227666	-0.24268	1.205417	-0.31763	0.725675	-0.81561	0.873936	-0.84779	-0.68319	-0.10276	-0.23181	-0.48329
11	1.069374	0.287722	0.828613	2.71252	-0.1784	0.337544	-0.09672	0.115982	-0.22108	0.46023	-0.77366	0.323387	-0.01108	-0.17849	-0.65556	-0.19993	0.124005	-0.9805	-0.98292	-0.1532	-0.03688	0.074412
12	-2.79185	-0.32777	1.64175	1.767473	-0.13659	0.807596	-0.42291	-1.90711	0.755713	1.151087	0.844555	0.792944	0.370448	-0.73498	0.406796	-0.30306	-0.15587	0.778265	2.221868	-1.58212	1.151663	0.222182
12	-0.75242	0.345485	2.057323	-1.46864	-1.15839	-0.07785	-0.60858	0.003603	-0.43617	0.747731	-0.79398	-0.77041	1.047627	-1.0666	1.106953	1.660114	-0.27927	-0.41999	0.432535	0.263451	0.499625	1.35365
12	1.103215	-0.0403	1.267332	1.289091	-0.736	0.288069	-0.58606	0.18938	0.782333	-0.26798	-0.45031	0.936708	0.70838	-0.46865	0.354574	-0.24663	-0.00921	-0.59591	-0.57568	-0.11391	-0.02461	0.196002
13	-0.43691	0.918966	0.924591	-0.72722	0.915679	-0.12787	0.707642	0.087962	-0.66527	-0.73798	0.324098	0.277192	0.252624	-0.2919	-0.18452	1.143174	-0.92871	0.68047	0.025436	-0.04702	-0.1948	-0.67264
14	-5.40126	-5.45015	1.186305	1.736239	3.049106	-1.76341	-1.55974	0.160842	1.23309	0.345173	0.91723	0.970117	-0.26657	-0.47913	-0.52661	0.472004	-0.72548	0.075081	-0.40687	-2.19685	-0.5036	0.98446
15	1.492936	-1.02935	0.454795	-1.43803	-1.55543	-0.72096	-1.08066	-0.05313	-1.97868	1.638076	1.077542	-0.63205	-0.41696	0.052011	-0.04298	-0.16643	0.304241	0.554432	0.05423	-0.38791	-0.17765	-0.17507
16	0.694885	-1.36182	1.029221	0.834159	-1.19121	1.309109	-0.87859	0.44529	-0.4462	0.568521	1.019151	1.298329	0.42048	-0.37265	-0.80798	-2.04456	0.515663	0.625847	-1.30041	-0.13833	-0.29558	-0.57196
17	0.962496	0.328461	-0.17148	2.109204	1.129566	1.696038	0.107712	0.521502	-1.19131	0.724396	1.69033	0.406774	-0.93642	0.983739	0.710911	-0.60223	0.402484	-1.73716	-2.02761	-0.26932	0.143997	0.402492
18	1.166616	0.50212	-0.0673	2.261569	0.428804	0.089474	0.241147	0.138082	-0.98916	0.922175	0.744786	-0.53138	-2.10535	1.12687	0.003075	0.424425	-0.45448	-0.09887	-0.8166	-0.30717	0.018702	-0.06197
18	0.247491	0.277666	1.185471	-0.0926	-1.31439	-0.15012	-0.94636	-1.61794	1.544071	-0.82988	-0.5832	0.524933	-0.45338	0.081393	1.555204	-1.39689	0.783131	0.436621	2.177807	-0.23098	1.65018	0.200454
22	-1.94653	-0.0449	-0.40557	-1.01306	2.941968	2.955053	-0.06306	0.855546	0.049967	0.573743	-0.08126	-0.21575	0.044161	0.033898	1.190718	0.578843	-0.97567	0.044063	0.488603	-0.21672	-0.57953	-0.79923

Figure 2: Time Feature

The Amount feature represents the total value of purchases with a credit card. The Class feature means the label and has two possible values: 1 for fraudulent transactions and 0 for normal transactions. The dataset details are similar to the previous study in Figure 3.

Journal of Information System and Technology Management
EISSN: 0128-1666
Volume 10 Issue 38 (March 2025) PP. 181-201
DOI: 10.35631/JISTM.1038012



	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
2	-0.4704	0.207971	0.025791	0.403993	0.251412	-0.01831	0.277838	-0.11047	0.066928	0.128539	-0.18911	0.133558	-0.02105	149.62	0
3	0.463917	-0.1148	-0.18336	-0.14578	-0.06908	-0.22578	-0.63867	0.101288	-0.33985	0.16717	0.125895	-0.00898	0.014724	2.69	0
4	-2.89008	1.109969	-0.12136	-2.26186	0.52498	0.247998	0.771679	0.909412	-0.68928	-0.32764	-0.1391	-0.05535	-0.05975	378.66	0
5	-1.05965	-0.68409	1.965775	-1.23262	-0.20804	-0.1083	0.005274	-0.19032	-1.17558	0.647376	-0.22193	0.062723	0.061458	123.5	0
6	-0.45145	-0.23703	-0.03819	0.803487	0.408542	-0.00943	0.798278	-0.13746	0.141267	-0.20601	0.502292	0.219422	0.215153	69.99	0
7	0.401726	-0.05813	0.068653	-0.03319	0.084968	-0.20825	-0.55982	-0.0264	-0.37143	-0.23279	0.105915	0.253844	0.08108	3.67	0
8	-0.44359	0.002821	-0.61199	-0.04558	-0.21963	-0.16772	-0.27071	-0.1541	-0.78006	0.750137	-0.25724	0.034507	0.005168	4.99	0
9	-0.07613	-1.22213	-0.35822	0.324505	-0.15674	1.943465	-1.01545	0.057504	-0.64971	-0.41527	-0.05163	-1.20692	-1.08534	40.8	0
10	-0.21008	-0.49977	0.118765	0.570328	0.052736	-0.07343	-0.26809	-0.20423	1.011592	0.373205	-0.38416	0.011747	0.142404	93.2	0
11	0.739453	-0.54098	0.476677	0.451773	0.203711	-0.24691	-0.63375	-0.12079	-0.38505	-0.06973	0.094199	0.246219	0.083076	3.68	0
12	0.031967	0.253415	0.854344	-0.22137	-0.38723	-0.0093	0.313894	0.02774	0.500512	0.251367	-0.12948	0.04285	0.016253	7.8	0
13	-0.12949	-0.80998	0.359985	0.707664	0.125992	0.049924	0.238422	0.00913	0.99671	-0.76731	-0.49221	0.042472	-0.05434	9.99	0
14	-0.81561	0.873936	-0.84779	-0.68319	-0.10276	-0.23181	-0.48329	0.084668	0.392831	0.161135	-0.35499	0.026416	0.042422	121.5	0
15	-0.19993	0.124005	-0.9805	-0.98292	-0.1532	-0.03688	0.074412	-0.07141	0.104744	0.548265	0.104094	0.021491	0.021293	27.5	0
16	-0.30306	-0.15587	0.778265	2.221868	-1.58212	1.151663	0.222182	1.020586	0.028317	-0.23275	-0.23556	-0.16478	-0.03015	58.8	0
17	1.660114	-0.27927	-0.41999	0.432535	0.263451	0.499625	1.35365	-0.25657	-0.06508	-0.03912	-0.08709	-0.181	0.129394	15.99	0
18	-0.24663	-0.00921	-0.59591	-0.57568	-0.11391	-0.02461	0.196002	0.013802	0.103758	0.364298	-0.38226	0.092809	0.037051	12.99	0
19	1.143174	-0.92871	0.68047	0.025436	-0.04702	-0.1948	-0.67264	-0.15686	-0.88839	-0.34241	-0.04903	0.079692	0.131024	0.89	0
20	0.472004	-0.72548	0.075081	-0.40687	-2.19685	-0.5036	0.98446	2.458589	0.042119	-0.48163	-0.62127	0.392053	0.949594	46.8	0
21	-0.16643	0.304241	0.554432	0.05423	-0.38791	-0.17765	-0.17507	0.040002	0.295814	0.332931	-0.22038	0.022298	0.007602	5	0
22	-2.04456	0.515663	0.625847	-1.30041	-0.13833	-0.29558	-0.57196	-0.05088	-0.30421	0.072001	-0.42223	0.086553	0.063499	231.71	0
23	-0.60223	0.402484	-1.73716	-2.02761	-0.26932	0.143997	0.402492	-0.04851	-1.37187	0.390814	0.199964	0.016371	-0.01461	34.09	0
24	0.424425	-0.45448	-0.09887	-0.8166	-0.30717	0.018702	-0.06197	-0.10385	-0.37042	0.6032	0.108556	-0.04052	-0.01142	2.28	0
25	-1.39689	0.783131	0.436621	2.177807	-0.23098	1.65018	0.200454	-0.18535	0.423073	0.820591	-0.22763	0.336634	0.250475	22.75	0
26	0.578843	-0.97567	0.044063	0.488603	-0.21672	-0.57953	-0.79923	0.8703	0.983421	0.321201	0.14965	0.707519	0.0146	0.89	0

Figure 3: Amount and Class Features

The dataset consists of 284,807 transactions, of which 492 were identified as fraudulent, while the other transactions were classified as legitimate. Based on the numerical statistics, it is evident that this dataset exhibits a significant imbalance, with a mere 0.173% of transactions being classified as fraudulent. Preprocessing of the data is vital because the distribution ratio of classes significantly impacts the accuracy and precision of the model. The model learns from the patterns in the features (V1 to V28, Time, and Amount) to predict the Class label. Accurate predictions of this Class label are crucial for effective fraud detection.

Since the dataset is highly imbalanced, with most transactions being non-fraudulent, special techniques such as undersampling or oversampling are necessary to ensure the model performs well in identifying the minority class (fraudulent transactions).

Next, Figure 4 generates a statistical summary of the numerical columns in the DataFrame df. It provides critical descriptive statistics that summarise the dataset's distribution's central tendency, dispersion, and shape.

	V27	V28	Amount	Class
	2.848070e+05	2.848070e+05	284807.000000	284807.000000
	-3.660091e-16	-1.227390e-16	88.349619	0.001727
	4.036325e-01	3.300833e-01	250.120109	0.041527
	-2.256568e+01	-1.543008e+01	0.000000	0.000000
	-7.083953e-02	-5.295979e-02	5.600000	0.000000
	1.342146e-03	1.124383e-02	22.000000	0.000000
	9.104512e-02	7.827995e-02	77.165000	0.000000
	3.161220e+01	3.384781e+01	25691.160000	1.000000

Figure 4: Dataset Statistical Summary Snapshot For Amount And Class Columns

Journal of Information System and Technology Management
EISSN: 0128-1666
JISTM
Volume 10 Issue 38 (March 2025) PP. 181-201
DOI: 10.35631/JISTM.1038012

In the provided statistical summary, the Amount feature has a maximum value of €25,691.16, significantly higher than the mean and the upper quartile (Q3). This suggests the presence of outliers or extremely high-value transactions in the dataset.

Moreover, the Time feature ranges from 0 to 172,792 seconds (roughly 48 hours). While the values are within a reasonable range for time measurements, the wide span might suggest that scaling or normalisation to a standardised range (0 to 1) could help feature consistency across the dataset in Figure 5.

These characteristics indicate that the dataset may contain outliers or extreme values, particularly in the Amount feature. Applying RobustScaler helps mitigate the impact of these outliers by scaling features based on robust statistics (median and interquartile range), making the scaling process more resilient to outliers and improving the robustness of machine learning models trained on the data.

	Time	V1	V2	V3
count	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05
mean	94813.859575	1.168375e-15	3.416908e-16	-1.379537e-15
std	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00
min	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00

8 rows × 31 columns

Figure 5: Dataset Statistical Summary Snapshot For Time Column

Next, Figure 6 demonstrates the preprocessing steps applied to the DataFrame df using RobustScaler from sklearn.preprocessing. These preprocessing steps prepare the data for subsequent analysis or model training by making the numerical features more suitable and consistent for machine learning algorithms, especially when dealing with different scales and potential outliers in the data.

Journal of Information System and Technology Management
EISSN: 0128-1666
Volume 10 Issue 38 (March 2025) PP. 181-201
DOI: 10.35631/JISTM.1038012

V27	V28	Amount	Class
0.133558	-0.021053	1.783274	0
-0.008983	0.014724	-0.269825	0
-0.055353	-0.059752	4.983721	0
0.062723	0.061458	1.418291	0
0.219422	0.215153	0.670579	0
...	...	...	...
0.943651	0.823731	-0.296653	0
0.068472	-0.053527	0.038986	0
0.004455	-0.026561	0.641096	0
0.108821	0.104533	-0.167680	0
-0.002415	0.013649	2.724796	0

Figure 6: Normalised Dataset

The RobustScaler scales features using statistics that are robust to outliers. Specifically, (1) removes the median and scales the data according to the interquartile range (IQR).

$$\chi_{robust} = \frac{\chi - median(\chi)}{IQR(\chi)}$$

The dataset contains a total of 284,807 transactions. It has 284,315 non-fraudulent transactions (Class 0) and 492 fraudulent transactions (Class 1).

It highlights that fraudulent transactions are relatively rare compared to non-fraudulent ones. This imbalance can impact the performance of machine learning models, as they may tend to predict the majority class (non-fraudulent) more frequently.

This output provides a foundational understanding of the dataset's class distribution, which is fundamental for effective fraud detection model development and evaluation.

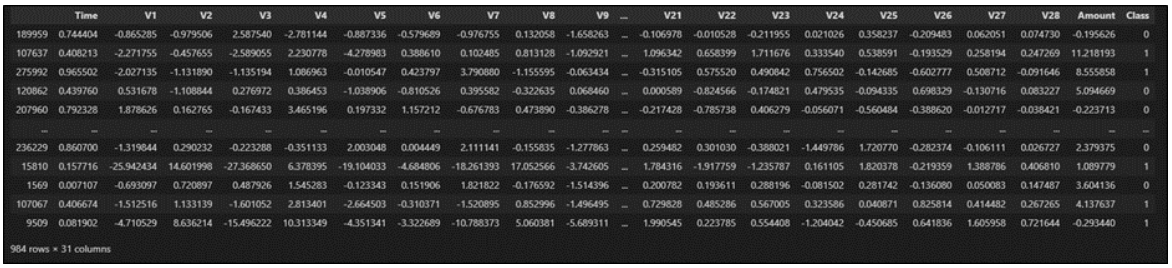
Next, techniques like undersampling (reducing the number of majority class samples) may be necessary to address the class imbalance and improve model accuracy in detecting fraudulent transactions.

After sampling, pd.concat concatenates (combines) the fraudulent transactions (frauds) with the sampled non-fraudulent transactions (not_frauds.sample(...)), thereby creating a new dataframe balanced_df that has an equal number of fraudulent and non-fraudulent transactions. This indicates that the balanced_df dataset now has an equal number of fraudulent and non-fraudulent transactions, ensuring class balance, which is beneficial for training machine learning models, particularly in fraud detection tasks.

Then, implemented a 5-fold cross-validation for the LogisticRegression model using scikit-learn. Before the data split, it trained the LogisticRegression model on each fold, evaluated its performance using the accuracy score, and returned an array of scores, one for each fold.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 10 Issue 38 (March 2025) PP. 181-201
DOI: 10.35631/JISTM.1038012

The output shows that the mean accuracy of the model is 94.41%, with a standard deviation of 2.76%. This suggests that the model performed well on the dataset and has learned the patterns in the data without overfitting, given that the performance is consistent across different folds.
Next, Figure 7 shuffled the balanced_df dataframe and then displayed its contents. Shuffling ensures that the order of the transaction data does not affect the performance or learning of machine learning algorithms. This is particularly important to prevent any unintended patterns or biases that might arise from the original ordering of data.



	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
189959	0.744404	-0.865285	-0.979506	2.587540	-2.781144	-0.887336	-0.579689	-0.976755	0.132058	-1.658263	...	-0.106978	-0.010528	-0.211955	0.021026	0.358237	-0.209483	0.062051	0.074730	-0.195626	0
107637	0.408213	-2.271755	-0.457655	-2.589055	2.230778	-4.278983	0.388610	0.102485	0.813128	-1.092921	...	1.096342	0.658399	1.711676	0.333540	0.538591	-0.193529	0.258194	0.247269	11.218193	1
275992	0.965502	-2.027135	-1.131890	-1.135194	1.086963	-0.010547	0.423797	3.790880	-1.155595	-0.063434	...	-0.315105	0.575520	0.490842	0.756502	-0.142685	-0.602777	0.508712	-0.091646	8.555858	1
120862	0.439760	0.531678	-1.108844	0.276972	0.386453	-1.038906	-0.810526	0.395582	-0.322635	0.068460	...	0.000589	-0.824566	-0.174821	0.479535	-0.094335	0.698329	-0.130716	0.083227	5.094669	0
207960	0.792328	1.878626	0.162765	-0.167433	3.465196	0.197332	1.157212	-0.676783	0.473890	-0.386278	...	-0.217428	-0.785738	0.406279	-0.056071	-0.560484	-0.388620	-0.012717	-0.038421	-0.223713	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
236229	0.860700	-1.319844	0.290232	-0.223288	-0.351133	2.003048	0.004449	2.111141	-0.155835	-1.277863	...	0.259482	0.301030	-0.388021	-1.449786	1.720770	-0.282374	-0.106111	0.026727	2.379375	0
15810	0.157716	-25.942434	14.601998	-27.368650	6.378395	-19.104033	-4.684806	-18.261393	17.052566	-3.742605	...	1.784316	-1.917759	-1.235787	0.161105	1.820378	-0.219359	1.388786	0.406810	1.089779	1
1569	0.007107	-0.693097	0.720897	0.487926	1.545283	-0.123343	0.151906	1.821822	-0.176592	-1.514396	...	0.200782	0.193611	0.288196	-0.081502	0.281742	-0.136080	0.050083	0.147487	3.604136	0
107067	0.406674	-1.512516	1.133139	-1.601052	2.813401	-2.664503	-0.310371	-1.520895	0.852996	-1.496495	...	0.729828	0.485286	0.567005	0.323586	0.040871	0.825814	0.414482	0.267265	4.137637	1
9509	0.081902	-4.710529	8.636214	-15.496222	10.313349	-4.351341	-3.322689	-10.788373	5.060381	-5.689311	...	1.990545	0.223785	0.554408	-1.204042	-0.450685	0.641836	1.605958	0.721644	-0.293440	1

984 rows × 31 columns

Figure 7: Shuffled Balanced Dataset

Shuffling is typically done before splitting the dataset into training, validation, and testing sets. This helps ensure that each subset of data (training, validation, testing) is representative of the overall distribution of the data. By shuffling, the model learns more generalisable patterns rather than being biased by the sequential order of data entries.

Prototype Development

The split of the dataset into different sets to train, test, and validate the model. This ensures the model is evaluated on unseen data, which helps assess its performance and generalisation capability.

A 70% training set was used to train the machine learning model. The model learns the relationships between features and labels from this set.
Then, a 20% testing set was used to evaluate the model after training. This set provides an unbiased evaluation of the model's performance since the model has yet to see this data during training.

Lastly, the remaining 10% validation set was used to help assess the model's performance while fine-tuning the model to avoid overfitting.

Training Set

Features (x_train_b.shape): (688, 30)

This indicates that there are 688 samples, each with 30 features. These features are the inputs that the model will learn from.

Labels (y_train_b.shape): (688,)

This indicates that there are 688 corresponding labels for the training samples. These labels are the target values the model aims to predict.

Journal of Information System and Technology Management
EISSN: 0128-1666
Volume 10 Issue 38 (March 2025) PP. 181-201
DOI: 10.35631/JISTM.1038012

Testing Set

Features (x_test_b.shape): (196, 30)

This indicates that there are 196 samples, each with 30 features. These features are the inputs used to evaluate the model's performance.

Labels (y_test_b.shape): (196,)

This indicates that there are 196 corresponding labels for the testing samples. These labels are the true values used to evaluate the model's predictions.

Validation Set

Features (x_val_b.shape): (100, 30)

This indicates that there are 100 samples, each with 30 features. These features are the inputs used for model selection and tuning.

Labels (y_val_b.shape): (100,)

This indicates that there are 100 corresponding labels for the validation samples. These labels are the true values used to validate the model's performance during tuning.

After splitting the dataset into training, testing, and validation sets, it is essential to verify the class distribution in each set. This ensures that the balance between fraudulent and non-fraudulent cases is maintained across all sets. The output indicates the distribution of classes (fraudulent and non-fraudulent transactions) in each training, testing, and validation set.

Training Set Class Distribution

Class 1 (Fraudulent Transactions) has 345 instances, and Class 0 (Non-Fraudulent Transactions) has 343 instances. The training set maintains a nearly equal balance between fraudulent and non-fraudulent transactions, which is crucial for the model to learn effectively from both classes.

Testing Set Class Distribution

Class 0 (Non-Fraudulent Transactions) has 101 instances, and Class 1 (Fraudulent Transactions) has 95 instances. The testing set also balances the two classes, ensuring an unbiased evaluation of the model's performance on unseen data.

Validation Set Class Distribution

Class 1 (Fraudulent Transactions) has 52 instances, and Class 0 (Non-Fraudulent Transactions) has 48 instances. The validation set retains a balanced distribution, making it reliable for model selection and fine-tuning.

Next, the logistic regression model initialises its parameters (coefficients) to some initial values, usually zero or small random numbers. The training data x_train_b (features) and y_train_b (labels) are prepared. Each instance in x_train_b corresponds to an instance in y_train_b, where x_train_b contains the features, and y_train_b contains the target values (0 for 'Not Fraud' and 1 for 'Fraud'). For each data point in x_train_b, the logistic function calculates the probability that the instance belongs to class 1 (Fraud). The logistic function is defined in (2).

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 10 Issue 38 (March 2025) PP. 181-201
DOI: 10.35631/JISTM.1038012

$$p(X; b, w) = \frac{e^{w \cdot X + b}}{1 + e^{w \cdot X + b}} = \frac{1}{1 + e^{-w \cdot X + b}}$$

Lastly, the trained logistic regression model was saved using the joblib library to facilitate future use without retraining. The validation and test sets, previously split from the balanced dataset, were converted into pandas DataFrames and exported to CSV files. This process ensured that the datasets were properly stored and could be used for further evaluation. Both sets were set to maintain those of the original feature set and specifically exported without the Class column to focus on model predictions during evaluation.

Figure 8 illustrates a responsive Streamlit-based web application for the prototype. The app allows stakeholders to upload a CSV file containing historical credit card transaction data to make predictions and detect fraud and normal transactions using the trained model.

Experiment and Evaluation

To enhance the model's accuracy and robustness, various experiments were conducted to identify potential improvements and optimisations. This section discusses strategies for refining the model, focusing on adjustments to the data splitting ratio, data sampling, and preprocessing techniques.

The use of different data split ratios (70:30 and 80:20) in each SMOTE oversampling and undersampling technique was intended to explore how varying the amount of training data impacts the model's performance. The 70:30 split allows for a more robust evaluation with a larger validation/test set, providing insights into the model's generalization ability. The 80:20 split maximizes training data to potentially improve the learning process. By comparing these approaches, the goal is to understand the trade-offs between training data size and evaluation robustness, ultimately guiding towards the most effective strategy for the fraud detection model.

Journal of Information System and Technology Management
EISSN: 0128-1666
Volume 10 Issue 38 (March 2025) PP. 181-201
DOI: 10.35631/JISTM.1038012

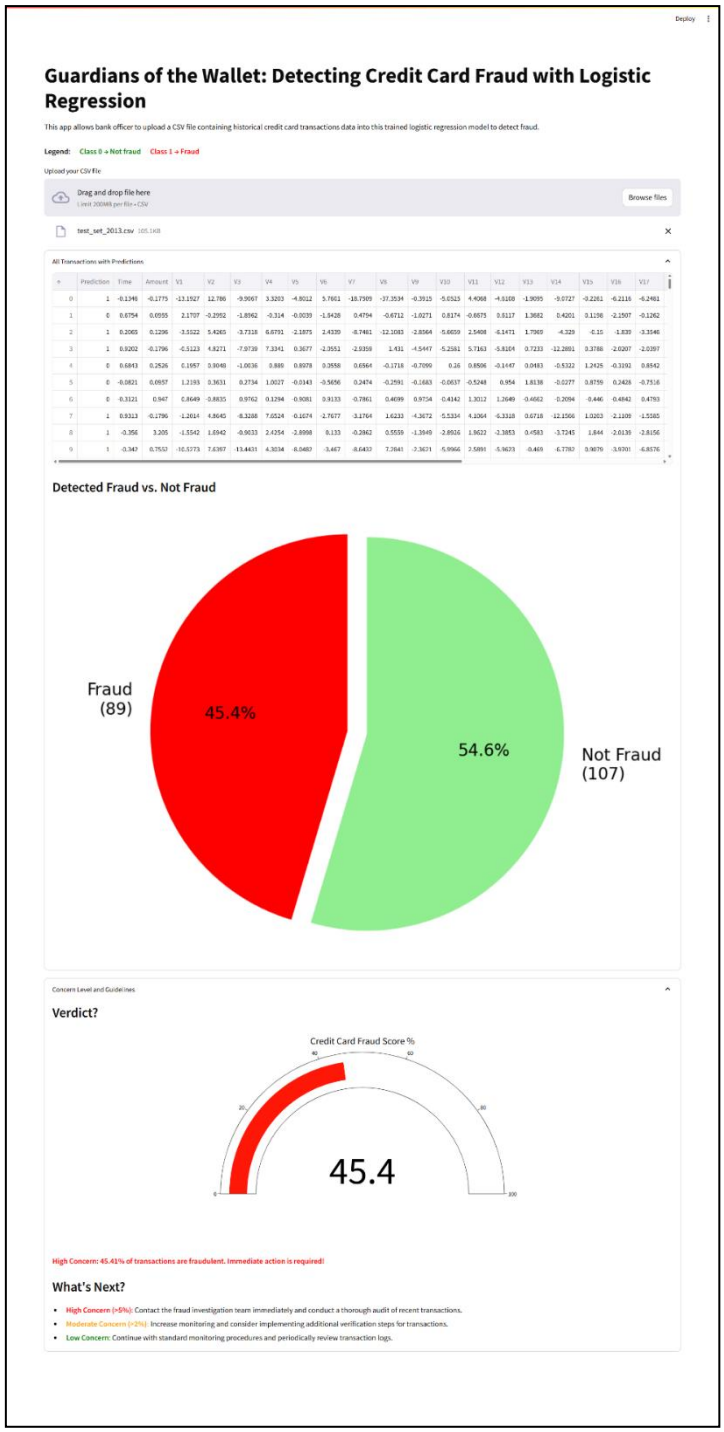


Figure 8: GUI Web Application

The decision to use undersampling instead of SMOTE can be justified based on the observed outcomes and the nature of the dataset. When SMOTE was applied, the results were extremely high, raising concerns about the reliability of these outcomes. Such unusually high results could indicate overfitting, where the model performs exceptionally well on the training data but may not generalise well to unseen data. Overfitting is a significant risk with synthetic data generation methods like SMOTE, especially if the original dataset is not adequately

Journal of Information System and Technology Management
EISSN: 0128-1666
Volume 10 Issue 38 (March 2025) PP. 181-201
DOI: 10.35631/JISTM.1038012

representative. Table 1 summarises the accuracy gained from the series of experiments on the SMOTE technique.

Table 1: Summary of SMOTE Technique Results

	Train set	Test set
	First experiment	
Split ratio	80	20
Accuracy (%)	98	
	Second experiment	
Split ratio	70	30
Accuracy (%)	97	

Using undersampling, the results were high but within a reasonable and expected range, even without normalisation. This suggests that the model is capturing the patterns in the data effectively without overfitting. Undersampling balances the dataset by reducing the number of non-fraudulent cases to match the number of fraudulent cases, which helps the model learn from a balanced dataset. Table 2 summarises the accuracy gained from the series of experiments on the undersampling technique.

Table 2: Summary Of Undersampling Technique Results

	Train set	Test set
	First experiment	
Split ratio	80	20
Accuracy (%)	94	
	Second experiment	
Split ratio	70	30
Accuracy (%)	94	

Now, the dataset was normalised using the robust scaler technique for the finalised model improvement. To form a balanced dataset, an equal number of non-fraud cases were undersampled to match the number of fraud cases and combined. The data was then 5-fold cross-validated and shuffled. It was split into training, test, and validation sets in a 70:30 ratio. Then, the logistic regression model is applied and trained. The k-fold and validation set results are displayed in Figure 9, with the PR-AUC score added.

Journal of Information System and Technology Management
EISSN: 0128-1666
JISTM
Volume 10 Issue 38 (March 2025) PP. 181-201
DOI: 10.35631/JISTM.1038012

```
5-Fold Cross-Validation Results
Mean accuracy: 94.41%
Standard deviation: 2.76%

Classification Report
              precision    recall  f1-score   support

   Not Fraud     94.12%      100%    96.97%        48
       Fraud       100%    94.23%    97.03%        52

    accuracy                         97.00%       100
   macro avg     97.00%    97.00%    97.00%       100
weighted avg     97.00%    97.00%    97.00%       100

PR AUC Score: 99.57

Confusion Matrix
[[48   0]
[ 3  49]]
```

Figure 9: Improved Undersampling 70:30 Results

Besides dataset balancing, cross-validation, appropriate data splitting, and logistic regression, the addition of robust scaling collectively contributed to the model's high performance, as evidenced by various evaluation metrics. This approach ensures a reliable and efficient prototype for detecting fraudulent credit card transactions, making it more resistant to outliers prevalent in the datasets.

Additionally, using a training-validation-test split is a common practice in machine learning projects. The shift to a 70:30 split for the training, testing, and validation sets is justified because it allows for a more reliable and unbiased evaluation of the prototype's performance. It also helps in preventing overfitting, enables better hyperparameter tuning, and ensures adherence to best practices in machine learning.

Comparing the current prototype system's performance with previous research helps to contextualise its effectiveness and identify any advancements or gaps. This comparison offers insights into how well the current model performs relative to established benchmarks in credit card fraud detection. Table 3 illustrates the differences in how the results were achieved despite using the same public dataset.

Benchmarking provides a reference point to measure the improvements and innovations of the current model against previous studies. It helped confirm whether the new model meets or exceeds the performance standards set by earlier research, adding credibility to the results.

Journal of Information System and Technology Management
EISSN: 0128-1666
Volume 10 Issue 38 (March 2025) PP. 181-201
DOI: 10.35631/JISTM.1038012

Table 3: Summary Of Previous Research In Evaluating Fraud Results

No.	Citation	Algorithm	Oversampling (SMOTE)	Undersampling	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
1	Tressa et al. (2023)	Random forest	Data split 70:30 85,296 normal 147 frauds	n/a	95	76	85	100
2	Aditi et al. (2022)	Logistic regression	n/a	Data split 80:20 286 normal 13 frauds	98.94	88.67	96.07	95.65
3	Jain et al. (2022)	Logistic regression	n/a	Data split 80:20 56,864 normal 98 frauds	61	56	58	100
4	Devika et al. (2022)	Logistic regression	Data split ratio n/a 71089 normal 113 frauds	n/a	81.94	52.21	63.78	99.91
5	Varmedja et al. (2019)	Logistic regression	Data split 80:20 56,864 normal 98 frauds	n/a	58.82	91.84	n/a	97.46
6	Our methodology	Logistic regression		Data split 70:30 48 normal 52 frauds	100	94.23	97.03	97
			Data split 80:20 40,051 normal 39,949 frauds		99	97	98	98

Development Challenges

Issues such as overfitting and outliers are commonly encountered with this imbalanced dataset. Overfitting occurs when a model learns the noise and details in the training data to such an extent that it performs poorly on new, unseen data. This is a critical issue, especially with datasets containing many features or imbalanced classes, as in the case of credit card fraud detection.

Discussion

This section presents a concise summary of the research outcomes, acknowledges the study's limitations, and suggests directions for future research.

```
Classification Report
              precision    recall  f1-score   support

     Genuine       0.50      1.00      0.67    284315
  Fraudulent       0.99      0.01      0.03    284315

    accuracy                           0.51    568630
   macro avg       0.75      0.51      0.35    568630
weighted avg       0.75      0.51      0.35    568630
```

Figure 10: SMOTE Overfitting On Test Data

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 10 Issue 38 (March 2025) PP. 181-201
DOI: 10.35631/JISTM.1038012

Signs of overfitting include high accuracy on training data but significantly lower accuracy on test data. The model's performance metrics, such as PR-AUC and F1 score, show a significant disparity between the two classes in Figure 10.

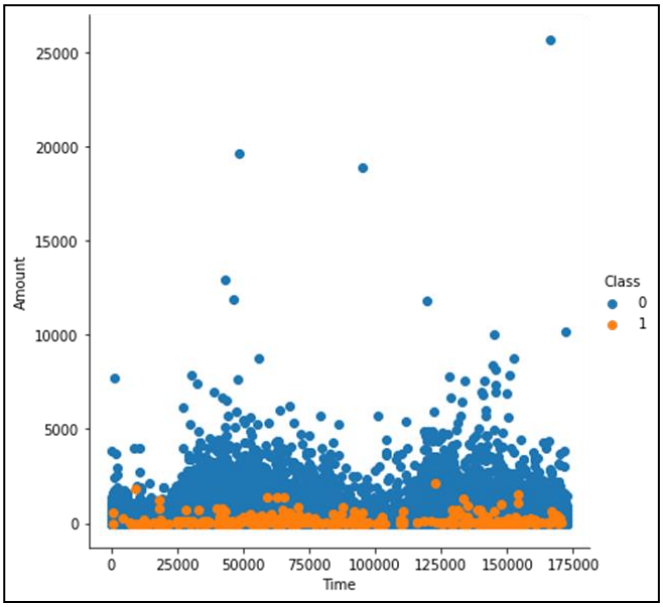


Figure 11: Outliers Challenge

Additionally, outliers are data points that deviate significantly from most of the data. They can skew the model's performance, especially in fraud detection, where fraudulent transactions might exhibit unusual patterns. Figure 11 helps illustrate outliers through the scatter plot of transaction time with the amount of money.

Classification Report

	precision	recall	f1-score	support
Genuine	1.00	0.97	0.98	284315
Fraudulent	0.04	0.86	0.08	492
accuracy			0.97	284807
macro avg	0.52	0.92	0.53	284807
weighted avg	1.00	0.97	0.98	284807

Figure 12: Imbalanced Dataset Challenge

Lastly, the dataset is highly imbalanced, with fraudulent transactions much rarer than legitimate ones. This imbalance can make the model biased towards predicting the majority class (genuine transactions), as shown in Figure 12.

Project Summary

This research project aimed to identify a machine learning-based fraud detection prototype for credit card transactions. The system leveraged preprocessing techniques and the logistic regression algorithm to distinguish between fraudulent and normal transactions. Key

Journal of Information System and Technology Management
EISSN: 0128-1666
Volume 10 Issue 38 (March 2025) PP. 181-201
DOI: 10.35631/JISTM.1038012
components included data preprocessing, model training, evaluation, and developing a prototype application for fraud detection.

Moreover, the project successfully implemented a functional prototype of the logistic regression model trained on historical credit card transaction data. This prototype was designed to detect transactions in fraudulent and normal categories based on features such as transaction amount, time, and class. This prototype served as a proof-of-concept for integrating predictive analytics into transaction monitoring systems, supporting proactive measures against financial fraud.

The accuracy and effectiveness of the prototype were evaluated using the PR-AUC metric. This evaluation provided insights into the model's ability to identify fraudulent transactions correctly and not wrongly flag legitimate transactions

Limitation

Despite its contributions, the project faced several limitations that impacted its scope and applicability. For financial institutions, the non-real-time processing capability of the prototype limits its immediate utility in detecting and responding to fraudulent transactions as they occur. The prototype primarily handles data in the form of CSV files and is not designed for real-time fraud detection. This limitation can affect their ability to mitigate financial losses promptly and protect customer assets in real-time scenarios.

The prototype is also limited in terms of user interaction, making it less accessible for the general public or citizens to upload their monthly credit card statements or similar inputs. Only banks or financial institutions with datasets similar to the one used for training can fully utilise the prototype. Limited user interaction capabilities restrict opportunities for incorporating diverse data sources to improve model accuracy over time.

Regulatory bodies and compliance officers oversee the implementation of fraud detection systems in financial institutions. The prototype's limitations in real-time processing and user interaction could impact compliance with regulatory requirements to ensure timely fraud detection and customer protection.

Recommendation

Based on the findings and limitations identified, several recommendations are proposed. While logistic regression is practical, consider exploring more advanced models such as gradient boosting machines (GBMs) or deep learning approaches like neural networks. These models may capture more complex patterns in the data and potentially improve detection accuracy.

Then, explore possibilities to add and publish other types of datasets on Kaggle, such as monthly credit card statements, if allowed by data providers. This will enable the system to be more versatile and useful to a broader audience, including individual users.

Lastly, investigate methods for real-time data processing and deployment of fraud detection models. Techniques such as stream processing frameworks (e.g., Apache Kafka, Spark Streaming) and containerisation (e.g., Docker, Kubernetes) can facilitate scalable and efficient deployment in production environments.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 10 Issue 38 (March 2025) PP. 181-201
DOI: 10.35631/JISTM.1038012

Conclusion

Stakeholders collectively benefit from the project's outcomes, which include improved fraud detection accuracy, enhanced security standards, and greater trust in financial transactions. The project's findings and recommendations encourage ongoing dialogue and collaboration among stakeholders to further enhance fraud detection systems and mitigate risks effectively.

While effective in its current form for retrospective analysis, future enhancements are recommended to address real-time processing, user interaction, and model adaptation challenges. By embracing these recommendations, financial institutions can strengthen their defences against financial fraud, safeguarding assets and maintaining trust with customers in an increasingly digital financial landscape.

Overall, the project successfully achieved all its objectives by leveraging machine learning to enhance fraud detection capabilities in credit card transactions. The prototype demonstrated promising results in identifying potentially fraudulent activities, laying the groundwork for future enhancements and applications in financial security systems.

Acknowledgments

I am immensely thankful to my supervisors, Dr. Gloria Jennis Tan and Dr. Tan Chi Wee for their invaluable guidance, constructive feedback, and continuous encouragement. Their expertise and insights have greatly contributed to the success of this research. Lastly, I extend my heartfelt thanks to my family and everyone who has supported me in completing this research study. Their patience, understanding, and motivation have been instrumental in ensuring this work's completion.

References

Aditi, A., Dubey, A., Mathur, A., & Garg, P. (2022). Credit card fraud detection using advanced machine learning techniques. 2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT), 1(1). https://doi.org/10.1109/ccict56684.2022.00022

Al-Khater, W. A., Al-Ma'adeed, S., Ahmed, A. A., Sadiq, A. S., & Khan, M. K. (2020). Comprehensive review of cybercrime detection techniques. IEEE Access, 8(1), 1–1. https://doi.org/10.1109/access.2020.3011259

Anwar, S., & Shujauddin, S. (2022). A study on credit card fraud detection based on behaviour and location analysis. International Journal of Advanced Research in Science, Communication and Technology (IJARSCT, 2(9), 2581–9429. https://doi.org/10.48175/568

BNM. (2024). Legislation - Bank Negara Malaysia. Www.bnm.gov.my. https://www.bnm.gov.my/legislation

Btoush, E., Zhou, X., Gururaian, R., Chan, K., & Tao, X. (2021). A survey on credit card fraud detection techniques in banking industry for cyber security. 2021 8th International Conference on Behavioral and Social Computing (BESC), 1(1). https://doi.org/10.1109/besc53957.2021.9635559

Cherif, A., Badhib, A., Ammar, H., Alshehri, S., Kalkatawi, M., & Imine, A. (2022). Credit card fraud detection in the era of disruptive technologies: A systematic review. Journal of King Saud University - Computer and Information Sciences, 35(1). https://doi.org/10.1016/j.jksuci.2022.11.008

Journal of Information System and Technology Management
EISSN: 0128-1666
JISTM
Volume 10 Issue 38 (March 2025) PP. 181-201
DOI: 10.35631/JISTM.1038012
Dayyabu, Y. Y., Arumugam, D., & Balasingam, S. (2023). The application of artificial intelligence techniques in credit card fraud detection: A quantitative study. E3S Web of Conferences, 389(1), 07023. https://doi.org/10.1051/e3sconf/202338907023

Devika, M., Kishan, S. R., Manohar, L. S., & Vijaya, N. (2022). Credit card fraud detection using logistic regression. 2022 Second International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE), 1(1). https://doi.org/10.1109/icatiece56365.2022.10046976

Fraud.net. (2024). Credit card fraud detection. Fraud.net. https://fraud.net/d/credit-card-fraud-detection/

Hanshika, V. G., Preethi, S., Sreemathy, S. P., & Ezhillin Freeda, S. (2023). Flood prediction using logistic regression. 2023 International Conference on Circuit Power and Computing Technologies (ICCPCT), 1(1). https://doi.org/10.1109/iccpct58313.2023.10245832

HLB. (2023). Credit Scores In Malaysia. Hong Leong Bank. https://www.hlb.com.my/en/personal-banking/blog/credit-scores-in-malaysia.html#:~:text=What%20Exactly%20Is%20A%20Credit

Jain, V., Kavitha, H., & Mohana Kumar, S. (2022). Credit card fraud detection web application using streamlit and machine learning. 2022 IEEE International Conference on Data Science and Information System (ICDSIS), 1(1). https://doi.org/10.1109/icdsis55133.2022.9915901

Kaggle. (2018). Credit Card Fraud Detection. Www.kaggle.com. https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

Khodayer, M., Khodayer, M., & Mohammed, O. (2022). Security measures of protection for banking systems. 2022 IEEE 9th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 1(1). https://doi.org/10.1109/picst57299.2022.10238672

Kim Sia Ling, Siti Suhana Jamaian, Mansur, S., & Kwan, A. (2023). Modeling tenant's credit scoring using logistic regression. SAGE Open, 13(3). https://doi.org/10.1177/21582440231189693

Kumar, N., Tomar, K., Sharma, T., Jyala, P., Malik, D., & Dawar, I. (2023). Customer behavior-based fraud detection of credit card using a random forest algorithm. 2023 International Conference on Artificial Intelligence and Applications (ICAIA) Alliance Technology Conference (ATCON-1), 1(1). https://doi.org/10.1109/icaia57370.2023.10169484

Li, W., Zhao, Y., Dai, M., & Li, J. (2023, April 1). Prediction and classification of ancient glass types based on logistic regression models. IEEE Xplore. https://doi.org/10.1109/ICEIB57887.2023.10170035

Mehrban, S., Khan, M. A., Nadeem, M. W., Hussain, M., Ahmed, M. M., Hakeem, O., Saqib, S., Kiah, M. L. M., Abbas, F., & Hassan, M. (2020). Towards secure fintech: A survey, taxonomy, and open research challenges. IEEE Access, 8(1), 23391–23406. https://doi.org/10.1109/access.2020.2970430

Mohammed, D., Asokan, K., & Kavitha Arunasalam. (2023). Anti-fraud measures and corporate policies to combat financial fraud in the financial institutes of Malaysia. E3S Web of Conferences, 389(1), 09028–09028. https://doi.org/10.1051/e3sconf/202338909028

Priyadarshini, A., Mishra, S., Mishra, D. P., Salkuti, S. R., & Mohanty, R. (2021). Fraudulent credit card transaction detection using soft computing techniques. Indonesian Journal

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 10 Issue 38 (March 2025) PP. 181-201
DOI: 10.35631/JISTM.1038012
of Electrical Engineering and Computer Science, 23(3), 1634. https://doi.org/10.11591/ijeecs.v23.i3.pp1634-1642

Ravalika, D., & Pitchai, R. (2023). Prediction of diabetes using binomial logistic regression. Proceedings of the Fourth International Conference on Smart Electronics and Communication (ICOSEC-2023), 1(1). https://doi.org/10.1109/icosec58147.2023.10276068

Ray, S. (2019). A quick review of machine learning algorithms. 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (Com-IT-Con), 1(1). https://doi-org.ezaccess.library.uitm.edu.my/10.1109/COMITCon.2019.8862451

Rohit, Mr., & Chaurasia, A. (2022). Selection of classification and regression algorithms for knowledge discovery –A review. 2022 Seventh International Conference on Parallel, Distributed and Grid Computing (PDGC), 1(1). https://doi.org/10.1109/pdgc56933.2022.10053127

Sekhar, G., Radhika Baskar, & S RimlonShibi. (2023). Prediction of heart disease using random forest in comparison with logistic regression to measure accuracy. 2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), 1(1). https://doi.org/10.1109/accai58221.2023.10199851

Shah, A., & Mehta, A. (2021). Comparative study of machine learning based classification techniques for credit card fraud detection. 2021 International Conference on Data Analytics for Business and Industry (ICDABI), 1(1). https://doi.org/10.1109/icdabi53623.2021.9655848

Singh, Y., Singh, K., & Singh Chauhan, V. (2022). Fraud detection techniques for credit card transactions. 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM), 1(1). https://doi.org/10.1109/iciem54221.2022.9853183

Tibrewal, T. P. (2023, September 15). Support vector machines (SVM): An intuitive explanation. Low Code for Data Science. https://medium.com/low-code-for-advanced-data-science/support-vector-machines-svm-an-intuitive-explanation-b084d6238106

Tressa, N., V Asha, M Govindaraj, Sangamesh Padanoor, Tabassum, R., Desai Vatsal Dharmesh, & Binju Saju. (2023). Credit card fraud detection using machine learning. 2023 3rd Asian Conference on Innovation in Technology (ASIANCON), 1(1). https://doi.org/10.1109/asiancon58793.2023.10270805

Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., & Anderla, A. (2019). Credit card fraud detection - machine learning methods. 2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH), 1(1). https://doi.org/10.1109/infoteh.2019.8717766

Victor, I. O. (2023, July 9). Credit card fraud detection: A machine learning approach to combat financial fraud. DEV Community. https://dev.to/cyber_holics/credit-card-fraud-detection-a-machine-learning-approach-to-combat-financial-fraud-dee

Whitrow, C., Hand, D. J., Juszczak, P., Weston, D., & Adams, N. M. (2008). Transaction aggregation as a strategy for credit card fraud detection. Data Mining and Knowledge Discovery, 18(1), 30–55. https://doi.org/10.1007/s10618-008-0116-z

Zivkovic, S. (2022a, July 25). #004 machine learning - logistic regression models - master data science 25.07.2022. Master Data Science. https://datahacker.rs/004-machine-learning-logistic-regression-model/

Zivkovic, S. (2022b, August 30). #012 machine learning - introduction to random forest - master data science 30.08.2022. Master Data Science. https://datahacker.rs/012-machine-learning-introduction-to-random-forest/