A Comprehensive Machine Learning Framework for Anomaly Detection in Credit Card Transactions

Fathe Jeribi
Department of Computer Science-Faculty of Engineering and Computer Science, Jazan University, Jazan, Saudi Arabia

***Abstract*—Cybercrimes originate in a variety of forms, and the majority of crimes involve credit cards. Despite various steps taken to prevent credit card fraud, it is crucial to alert customers to unusual attempts at fraudulent transactions. The internet has been largely geared to meet this challenge. Many studies have been published over the years to identify anomalies in credit card transactions, and machine learning (ML) has played a significant role in this. Though various anomaly detection techniques are in place, transaction irregularities remain, especially during banking card transactions. The objective of this proposed work is to bring out an efficient machine learning model for identifying abnormal anomalies in credit card-based transactions by considering the limitations of the existing frameworks. The proposed research employs a ML framework comprising data preprocessing, discovering correlations, outlier removal, feature reduction, and classification with a sampling trade-off. The framework uses classifiers such as logistic regression, kNN, support vector machines, and decision trees. The NearMiss and SMOTE approaches are used to address overfitting and underfitting issues through sampling trade-off, which is the defining feature of this research. Significant improvement was noticed when the machine learning models were evaluated using fresh data after a sampling trade-off.**

Keywords—Cybersecurity; anomaly detection; machine learning; optimization; nearmiss; SMOTE

I. Introduction

According to the Nilson Report, December 2020 [1], a leading business magazine that covers the worldwide payment card industry predicted payment card fraud losses globally will approach $32 billion in 2021, with approximately $12 billion in the United States. In 2021, worldwide losses due to fraud actually rose by 14% from the previous year. Over the course of the next decade, the industry is expected to lose $397 billion globally, with the United States contributing $165 billion. Card fraud cost issuers, merchants, and buyers and debit card transactions a total of $28.58 billion in 2020, or $6.8 every $100 in spending. In United States, as reported by the Federal Trade Commission (FTC), consumers lost over $5.8 billion in fraud in 2021, a 70% increase from the previous year [2]. Fig. 1 shows the 10 years trend on worldwide credit card frauds according to Nilson report 2021. "Credit card fraud is a significant issue for the banking industry and consumers today, and there is no fool-proof measure to thwart fraud" said Brian Quarrie, former managing director of First Data in the Middle East. Roughly 93 percent of financial fraud in Saudi Arabia occurred after the pandemic, confirming how cybercrime activity is rapidly increasing [3].

The prevalence of credit card fraud is increasing as technology develops and the creation of the universal super highway is made possible. In light of this, it is desirable to explore existing infrastructure for dealing with identity theft and credit card fraud. There are several concerns about detecting this sort of fraudulent act. This form of fraud detection is primarily reliant on data analysis, and most of this data is restricted by financial institutions due to privacy. Furthermore, due to the volume of transactions that occur each day, the analysis faces challenges in terms of technology deployment and for researchers exploring the data. The complexity of fraud detection techniques advances along with the fraudsters, who will change their strategies from time to time in order to succeed in their mission.

Machine learning has emerged as a vital part of fraud detection. It is a technology that assists in gathering and interpreting as much data on cardholders as possible in order to identify purchasing trends. Alerts, typing speed information, and fresh phone recognition are sent when fraudsters use card information in a new place. Also, if the transaction occurred at an unusual time, the banking system can flag the transaction in question and notify the cardholder. The black box fraud prevention system is a model that utilizes machine learning and contributes to the prevention of credit card fraud [4]. Such systems are becoming increasingly popular since they provide a credit card risk assessment score quickly and specify which features might result in potentially fraudulent transactions. Know Your Customer (KYC), voice-based biometrics, knowledge-based authentication (KBA), address verification services, adaptive authentication, geolocation alerts, and account takeover tools are the various strategies that financial institutions follow to protect their customers from such fraud.

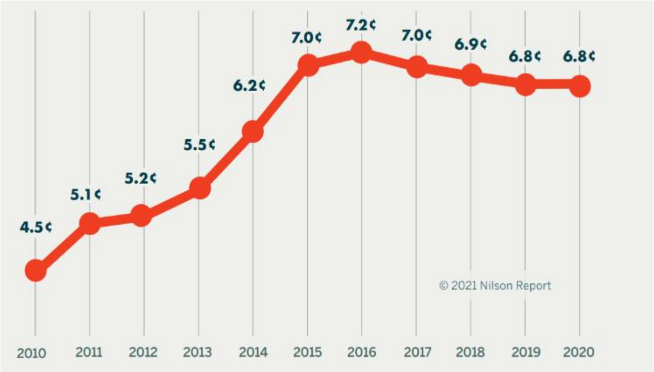


Fig. 1. Card fraud data worldwide.

Proper use of machine learning (ML) algorithms, adoption of systematic approaches for handling the data sets, and efficient use of evaluation methods are important in identifying fraud in credit card based transactions in real time. This is the prime motive behind this research framework. This paper intends to evolve a comprehensive ML-based framework by using several algorithms and a systematic approach. The framework differs from the previous ML-based frameworks in that it handles the data effectively, which is important for bringing enhanced performance in credit card fraud detection. The two main goals of this study are as follows: The first step is to analyze ML and DL-based frameworks for credit card fraud detection, and the second is to create a comprehensive ML-based classification model for fraud detection.

The subsequent sections of the paper are organized as follows: Following the Introduction in Section I, Section II presents the ML-based frameworks that are related to credit card fraud detection, their shortcomings, and the objectives of the proposed work. Section III presents the overall framework and methods of the ML experiments. Section IV presents the results, followed by discussions. The paper ends with a conclusion in Section V.

II. Related Works

As part of the research framework with the objective of identifying the effectiveness of the existing anomaly detection work, several bench-marking studies put forth in recent days are reviewed. Utmost care has been taken in choosing the research literature that reflects the real dilemma in the existing technologies of the financial sector to thwart the security issues of credit card transactions.

Manjeevan Seera et al. [5] address the escalating problem of payment card fraud by employing 13 statistical and machine learning models using both publicly available datasets and real transaction records. The study evaluates these models by comparing results from original transaction features with those derived from aggregated features identified through a genetic algorithm, with statistical tests confirming that aggregated features significantly enhance model performance. The findings highlight the potential of advanced techniques like feature aggregation to improve the accuracy and efficacy of fraud detection models in real-world scenarios. Likewise, a study by Georgios Charizanos et al. [6] introduced a novel real-time fraud detection framework that effectively handles non-stationary changes in fraud patterns and improves model training efficiency with large datasets. The framework employs a robust fuzzy logistic regression model to address class imbalance and separation issues, achieving high specificity and sensitivity, with performance metrics including a Matthew’s correlation coefficient exceeding 0.80 and accuracy over 99%. Comparative analysis shows that this methodology outperforms traditional machine learning and other fraud detection methods, promising reduced financial losses and enhanced customer satisfaction.

Alfaiz, N. S., and Fati, S. M. [7] developed a credit card fraud detection model using several machine learning algorithms in two subsequent phases. Though the authors claimed that their model’s performance was outstanding, it is evident that the performance of the validation test is considered the overall performance of the model. The researchers used the undersampling technique to obtain the overall performance, which is not always helpful for generalization, especially with nonlinear patterns in the dataset.

Alharbi, A. et al. [8] implemented a deep learning-based model by adopting text-to-image conversion and CNN. The converted images are fed into retrained CNN models. Though the authors claim above 98% accuracy, it is understood that when take into consideration the information loss during text-to-image conversion, the implemented model may not perform well for the new datasets. Hence, the performance of the work is questionable.

A hybrid ML-based algorithm is used in a fraud detection framework implemented by Jovanovic, D., and et al. [9]. They used synthetically oversampled data in addition to the original dataset in order to enhance the detection model. The validation trials were run multiple times, including with the actual unbalanced dataset as well as with a synthetic dataset produced using the SMOTE method. In order to lower the significant discrepancy between classes, more synthetic samples were produced using the SMOTE. According to the authors, the results of the simulations show that the proposed model outperforms rivals in the majority of the test cases. Another hybrid ML architecture was put forth by Malik, et al. [10] in which credit card fraud was first detected using cutting-edge machine learning algorithms, and then hybrid techniques were built using the best algorithm from the initial phase. According to their findings, the hybrid model AdaBoost combined with LGBM exhibits improved performance compared to bench marking works.

A credit card anomaly detection system developed by Stojanovi'c, B., et al. [11] was referred to as benchmarking due to its careful usage of multiple datasets, feature extraction techniques, selected algorithms following a thorough review, and distinctive training methodology. According to the authors, the results indicate that the machine learning techniques contribute to fraud detection with success. Similar to this, Mekterovi'c, I. et al. [12] brought out a much focused framework to solve a unique anomaly with the error "card-not-present" transactions, adopting a data mining technique through systematic feature engineering.

A high-performance ensemble staking method was proposed by Aljasim, M. et al. [13] in order to reveal cyberattacks in IoT edge nodes. Three different datasets were used in the experiments, and it is stated that the proposed classifier performed better than each of the base model classifiers. A game-changing methodology originated by Chaquet-Ulldemolins, J. et al. [14] and is applicable to all classification techniques. It enables the breaching of black-box models, the discarding of dependencies, and ultimately the elimination of unwanted biases. This led to a nonlinear analysis of financial data for fraud detection. It is concluded that it is possible to create an efficient, unique, unbiased, and traceable ML strategy that can handle transaction-level queries from clients and authorities in addition to complying with legal regulations.

To solve the unbalanced data issue, Strelcenia, E. et al. [15] researched a number of data augmentation methods and presented a brand-new model, K CGAN, for detecting anomalies in credit card transactions. The effectiveness of the augmentation methods is then assessed using a bunch of classifiers. According to the authors, the findings demonstrated that, when compared to other augmentation techniques, B SMOTE, K CGAN, and SMOTE were achieved the best precision and recall. KCGAN stood out among them with an improved F1 performance to win.

A reliable technique of credit card scam identification using ML and blockchain was proposed by Ashfaq, T. et al. [16]. Transactions are classified and transaction patterns are predicted using the XGboost and random forest (RF) algorithms. According to the authors, the simulation results demonstrate that the proposed method accurately locates transaction fraud.

To find abnormalities in credit card-based financial transactions, Moschini, G. et al. [17] developed a semiparametric-based learning model called ARIMA. To understand the customer's normal spending patterns, the proposed model is initially tuned using the daily average of legal money transactions. Using rolling windows and the fitted model, fraud in the testing set is then predicted. They employed a variety of techniques namely, K-means, the box plot, the local outlier factor, and the isolation forest algorithm, to find anomalies. According to the claim, the proposed model performs better using the box plot technique.

Jiang, J. R., et al. [18] proposed a deep learning-based fraud detection methodology by treating transactions as nonlinear and non-stationary. For detecting anomalies, several approaches, including deep learning, are used, and, on comparison, Tri-CAD exceeds the others in terms of precision, recall, and F1-score. Similar to this, G. Zioviris et al. [19] unveiled a deep learning system with the intention of effectively managing inbound transaction patterns and identifying fraudulent ones. They suggested two auto-encoders to carry out feature selection and learn the hidden patterns of data utilizing a nonlinear optimization model. To detect fraud, the selected features are fed into a deep convolutional neural network.

Mehbodniya, A., et al. [20] used several ML techniques, including CNN, in a fraud detection framework centered on the healthcare industry. In comparison to other algorithms, the KNN algorithm performed better. Similarly, Sanober, S. et al. [21] claim that an improved model that combines Spark with deep learning has materialized. For the purpose of finding abnormalities in the transactions, numerous ML classifiers are also used in addition to DL techniques. The suggested model performed exceptionally well when tested using real-world datasets, according to the authors.

Seeja, K. R., et al. [22] likewise propose a customer-centered matching algorithm to look for anomalies in incoming transactions and make intelligent decisions. According to a performance test of the proposed model using an anonymous and unbalanced dataset, it performs significantly better than other commonly used classifiers.

An ensemble learning-based model for recognizing anomalies in card transactions is proposed by Xie, Y., et al. [23]. The model was developed with the intention of dealing with unbalanced data. The experimental findings show that in recognizing the anomalies in the transactions, the proposed model exhibited the most competent performance.

In order to address credit card transaction anomalies, Karthika, J., et al. [24] brought out a convolutional neural network-based deep learning model that learns both spatial and temporal data. The dilated convolutional layer (DCL), which the author developed, enhances the CNN base model. Three datasets are used in the experiments, which are run with different parameters and compared to the existing CNN model. The proposed model, according to the authors, had a 97.39% accuracy rate.

A framework for ML-based anomaly identification in credit card transactions was created by Matthew, T. E. [25]. A set of six parametric classifiers constitutes the proposed model. Both hard and soft voting were used to combine the ensemble. Individual learning approaches were thought to perform worse than group learning techniques. As per the authors' claim, both were demonstrated steady outcomes and the soft voting classifiers were observed to perform better with typical data without feature selection.

Mienye, I. D., et al. [26] proposed a unique deep-learning model that used three learning machines as base learners: long short-term memory (LSTM) and gated recurrent unit (GRU) neural networks. The meta-learner was a multilayer perceptron (MLP). Meanwhile, to equalize the distribution of classes in the class feature of the dataset, the hybrid synthetic minority oversampling method as well as the edited nearest neighbor (SMOTE-ENN) method are used. According to the authors, the results showed that adopting the offered deep learning model demonstrated superior performance, which is far better compared to the performance of benchmarking ML classifiers.

Several weak points were noticed, especially in handling the datasets, while studying recent literature on utilizing deep learning and machine learning to detect credit card fraud. It is a fact that the performance of the overall framework in machine learning is mostly determined by the dataset. By keeping this in mind, the adopted methodology and used datasets of the articles benchmarked are studied. Following are gaps identified from the more recent researches on credit card fraud detection using ML and DL.

- It was noted that the selection of the algorithms for the framework was made without considering the linearity of the dataset.
- Many of the frameworks never addressed overfitting and underfitting issues, and though a few frameworks used undersampling methods for fitting the model, their suitability was not analyzed for the chosen model.
- The trade-off between different values of the hyperparameters used in the models is unknown.

The above key points were kept in mind while developing the method for this proposed research. The objectives of the proposed research include:

- Study and analyze learning-based credit card fraud detection frameworks proposed in the recent past.

- Identify the major flaws in the selected works and devise the mechanism by proposing systematic machine learning-based model.

III. Materials and Methods

A. The Proposed Classification Model

A unique classification model is devised by keeping in mind the limitations of the earlier frameworks for financial fraud detection using machine learning. Several micro level techniques are used in the framework in order to fill all the gaps identified during the literature survey. Fig. 2 shows the overall framework of a credit card fraud detection using several classifiers.

B. Dataset Description

The dataset used for this research framework was obtained from the Kaggle open-source data repository community [27]. To sense the data, one needs to explore the dataset. All features other than the transaction and amount are scaled, and their names are masked out of respect for privacy.

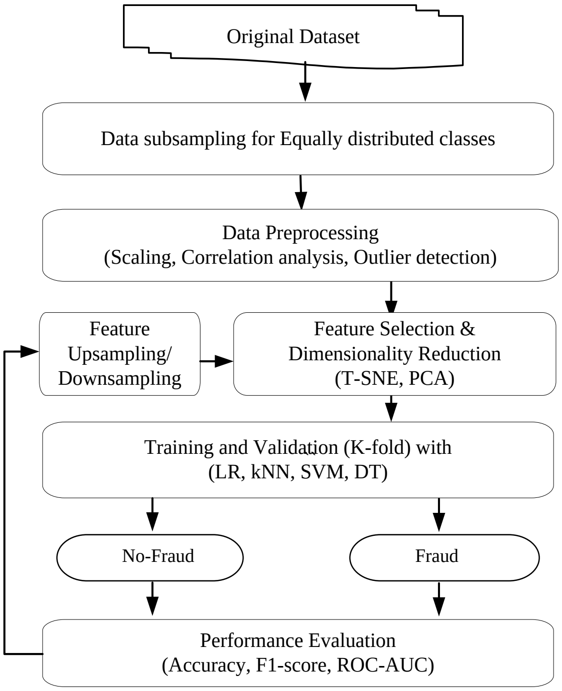

Original Dataset
Data subsampling for Equally distributed classes
Data Preprocessing
(Scaling, Correlation analysis, Outlier detection)
Feature Upsampling/ Downsampling
Feature Selection &
Dimensionality Reduction
(T-SNE, PCA)
Training and Validation (K-fold) with
(LR, kNN, SVM, DT)
No-Fraud
Fraud
Performance Evaluation
(Accuracy, F1-score, ROC-AUC)

Fig. 2. The proposed CCFD framework.

After analyzing the class feature, it was discovered that there was a serious imbalance that needed to be fixed. Only 1% of transactions are fraudulent, while more than 99% of transactions are normal that needed to be fixed. Only 1% of transactions are fraudulent, while more than 99% of transactions are normal. To ensure a balanced distribution of classes, the samples are then evenly distributed by creating a sub-sample of the data frame, which aids the algorithms in better understanding the patterns that define whether a transaction is fraudulent or not. Fig. 3 shows the distribution of class feature before and after subsampling.

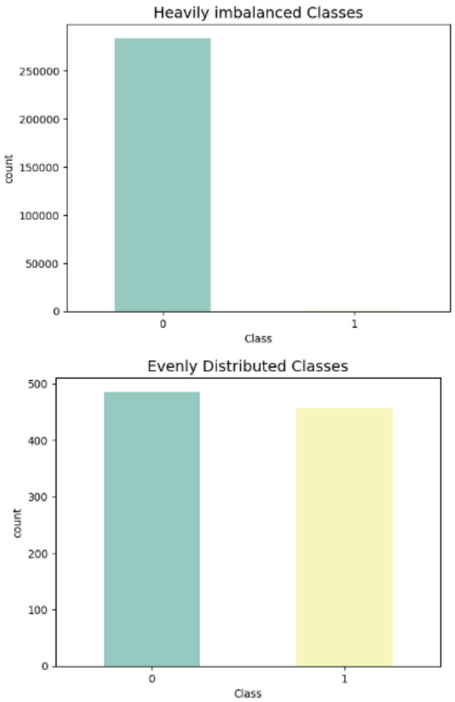


Fig. 3. Class feature before and after subsampling.

C. Data Preprocessing

As a first step in preprocessing, the missing values are filled with the average of the respective columns. The remaining columns, amount and time, should be scaled, as most of the data has already been scaled. The testing set needs to be separated from the original data frame for testing before applying the random undersampling technique. Models must be tested using the original testing set rather than one produced through undersampling or oversampling. To enable pattern detection, models are fitted to under- and over-sampled data sets and then tested on the original testing set.

D. Correlation Analysis

Correlation analysis identifies the most significant features. Negative correlations of the features against the class show that fraud transactions are more likely to occur. Positive correlations of the features against the class show that the likelihood of a fraudulent transaction increases as the feature correlation increases. This is clearly reflected in the heat map which is shown in Fig. 4.

Also, the extreme outliers are removed from the significant features with high correlation, and it is presumed that this will contribute to classification accuracy. A trade-off between different values of threshold in the interquartile range is used to remove outliers. A higher threshold is used to remove only extreme outliers in order to avoid information loss. The histograms shown in Fig. 5 illustrates the density distribution of fraud transactions on selected features with high correlation with class feature namely V10, V12 and V14, before and after outlier removal.

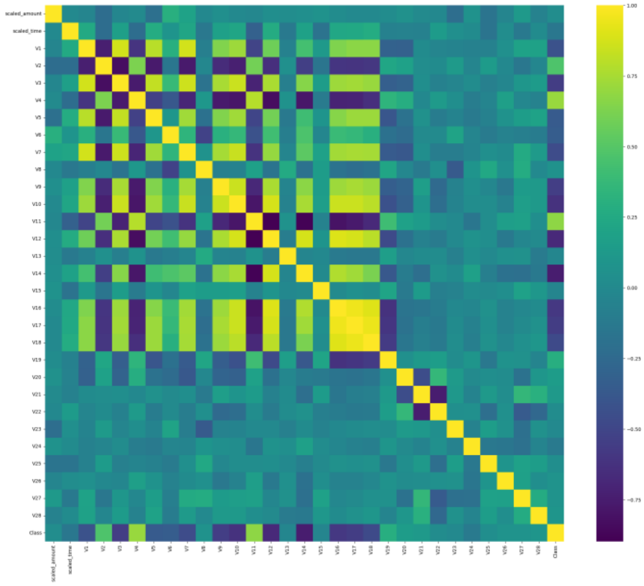
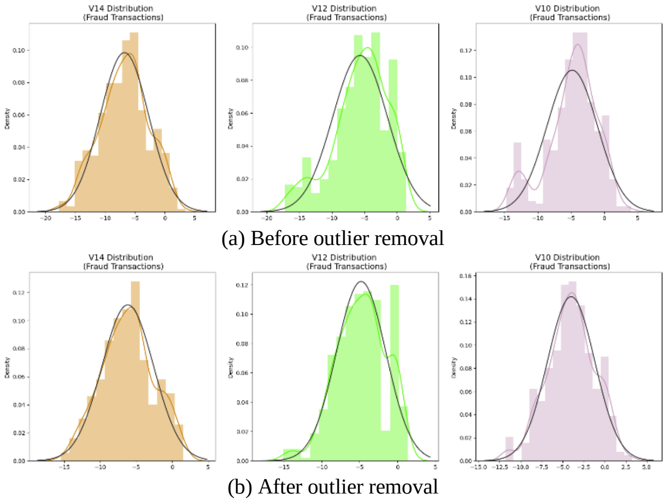


Fig. 4. Heat map showing the correlation matrix of the features.



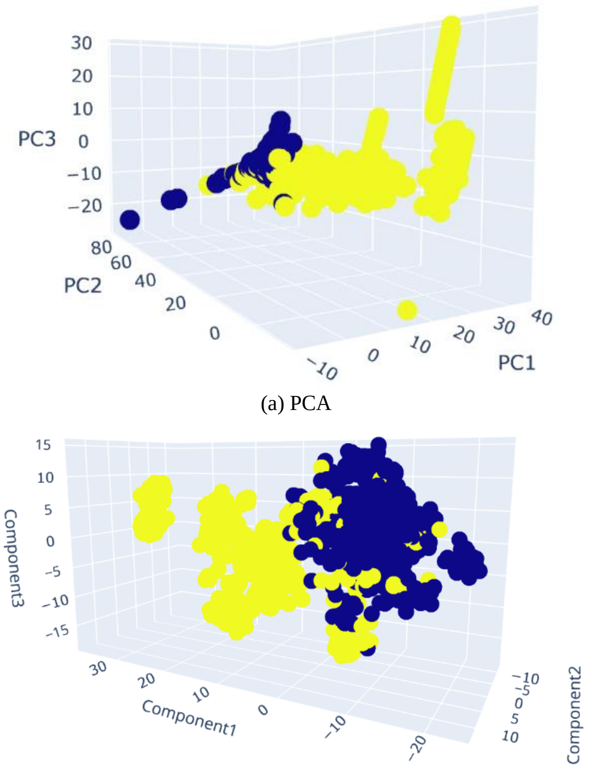
(a) Before outlier removal

(b) After outlier removal

Fig. 5. Density distribution of fraud transactions in V10, V12 and V14.

E. Dimensionality Reduction and Feature Extraction

In machine learning research, dimensionality reduction has a number of benefits, including obtaining a less complex model, a shortened training period, a reduction in space complexity, an enhancement in accuracy, improved visualization, the ability to detect noise, and many more. For feature extraction, two alternative techniques, T-distributed Stochastic Neighbor Embedding (t-SNE) [28] and Principal Component Analysis (PCA) [29], are used. Although both methods are semiparametric, the former is a nonlinear technique, whilst the later is a linear one. If there is a nonlinear relationship in the data, t-SNE will be useful for anomaly detection even though the problem is linear in form and the PCA is sufficient to bring concentrated features as principal components. Both algorithms were employed on the dataset for the proposed problem. PCA's computation time is 0.032 seconds, but t-SNE's computation time is 8.8 seconds for 3 components each. The Fig. 6 exhibits the 3D visualization of the data points after dimension reduction.



(a) PCA

(b) t-SNE

Fig. 6. 3D Visualization of the data points after feature reduction (limited with three components).

F. Classification

Four machine learning classifiers are used to classify the fraud transactions from the normal transactions: logistic regression (LogR), k nearest neighbor (kNN), support vector machine (SVM), and decision tree (DT). Let’s look at the brief note on each of the classifiers.

1) Logistic Regression (LogR): A simple linear statistical model widely employed for classification is known as logistic regression (LogR). The objective of using logistic regression is to identify the model that most accurately captures the implied relationship between the dependent and independent variables. It is ideal for binary categorization. In LR, the sigmoid function acts to determine how likely a label is [30]. The sigmoid function is a mathematical function that is used to convert anticipated outcomes into probabilities. The function may transfer any real value into a value between 0 and 1. In a classification, when variables without relationship or with least relationship to the target variable gets eliminated, logistic regression will perform better. Hence, feature engineering is a crucial component of its performance.

2) K Nearest Neighbour (kNN): kNN is a nonlinear and nonparametric supervised algorithm. The concept underlying Nearest Neighbour classifier is straightforward: data objects are classified by their proximate neighbors. Knowing that it is typically useful to take several neighbors into account, this method is generally referred to as k-Nearest Neighbour (kNN) Classification. The parameter k is denoted the number of labelled points used for classification to identify the classes. It is also known as Memory-based Classification [31] since the training instances are needed at runtime, which means that they must be in memory at runtime. It is known as a Lazy Learning technique since inference gets put off until runtime. It is also known as example-based or a case-based classification since it only uses the training instances to make classification decisions.

3) Support Vector Machine (SVM): When the dataset contains precisely two classifications, then support vector machine is the ideal choice. The support vector machine algorithm (SVM) classifies data by determining the best hyperplane that separates all of the data points in one class from the others. The hyperplane with the biggest margin separating the two classes is the optimum hyperplane for an SVM [32]. The margin is the maximum width of the slab that is perpendicular to the hyperplane but has no internal data points. SVMs use supervised learning method in order to classify unknown data using known classes.

4) Decision Tree (DT): The decision tree is a non-parametric learning technique used in classification and regression applications that is a member of the family of supervised learning algorithms. DT is hierarchical in structure, including a root node, branching nodes, inner nodes, and leaf nodes. It is a rule-based approach to making decisions that is analogous to how people make decisions [33]. An internal node represents a data instances, a branch indicates a decision, and each leaf node shows the outcome in a decision tree, which seems like a flowchart. Decision tree learning employs the divide and conquer strategy by performing a greedy search to discover the optimal partition of data points within a tree. The entire procedure is then repeated from the top down recursively until all or almost all of the items are finally assigned to specific class labels. The complexity of the decision tree influences whether all of the data instances are grouped as homogenous sets.

G. Training, Cross Validation and Sampling Trade-off

Though all the above algorithms are non-parametric, the models are trained using the training set with a trade-off between different sampling methods. Prior to training, data samples are divided into a training set and a test set. Though the training is carried out by fitting the data by importing the predefined classifiers using Python libraries, the training scores using cross validation of the classifiers are recorded at each iteration. By doing this, the best learning parameters are obtained from each classifier and their learning performance is recorded as ‘training score’. Five-fold cross-validation is exercised in the experiments, meaning training data is divided into five segments, one of which will be taken out for validation and the other remaining for training. On completing 5 spells, the average score is recorded as the "cross validation score’ for analysis.

Model optimization is achieved through oversampling and undersampling trade-off methods. The NearMiss [34] and SMOTE (Synthetic Minority Oversampling Technique) [35] are used to address problems caused by class imbalances during undersampling and oversampling. The NearMiss is initially tried to solve the class imbalances caused by undersampling. SMOTE generates synthetic points from the minority class to achieve a level playing field among the minority and majority classes. It selects a distance between the minority class's nearest neighbors and generates artificially created points that span these distances. In contrary to random undersampling, more data is saved as a result of no rows being discarded. Although SMOTE is more likely to be accurately computed than random undersampling, it will take longer to train because no rows are discarded, as previously indicated.

H. Evaluation Metrics

The dataset is divided into multiple training and validation set pairs solely for the model's optimization. The validation sets effectively become part of the data used because they can optimize the model during training, such as determining when to stop learning. After making all of these assessments, if a specific algorithm is chosen and its error is to be reported, this must be done using a separate test set that was not utilized during the final system's training. For the error estimate to be useful, the dataset must not have been used earlier for training or validation and must be substantial. In light of this, a portion of the dataset should first be set aside as the test set, with the remainder utilized for training and validation. The performance of the binary classification is reflected in the confusion matrix as indicated in Table I.

TABLE I. CONFUSION MATRIX

	Predicted Class		
		Positive	**Negative**
Actual Class	Positive	t_p true positive	f_n false negative
	Negative	f_p false positive	t_n true negative

The performance of the models developed using four distinct algorithms is evaluated using multiple types of performance measures. Accuracy is not only sufficient to evaluate the performance of the models especially when using imbalanced datasets. Hence, TPR, FPR, Error rate, F1-Score, AUC-ROC, along with the Accuracy score, were used.

$$Precision = \frac{t_p}{t_p + f_p}$$

$$TPR/Recall = \frac{t_p}{t_p + f_n}$$

$$FPR = \frac{f_p}{f_p + t_n}$$

$$Error\ rate = \frac{f_p + f_n}{N}$$

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{t_p + t_n}}{\boldsymbol{N}}$$

$$Accuracy = \frac{t_p + t_n}{N}$$

$$F1-score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

The projected false events as potential fraud are significant in credit card fraud detection because they are subject to study. The experiments were carried out in the Windows 11 environment using the Python 3.11 (64-bit) programming tool's Scikit Learn on a Jupyter notebook.

IV. Results and Discussions

Fig. 7 depicts the accuracy performance of the four classifiers with different sample sizes, as well as the comparative performance of the 'training score' (classifier training performance with learning parameters on generalization) and the 'cross-validation score' (using 5-fold cross validation of the training set).

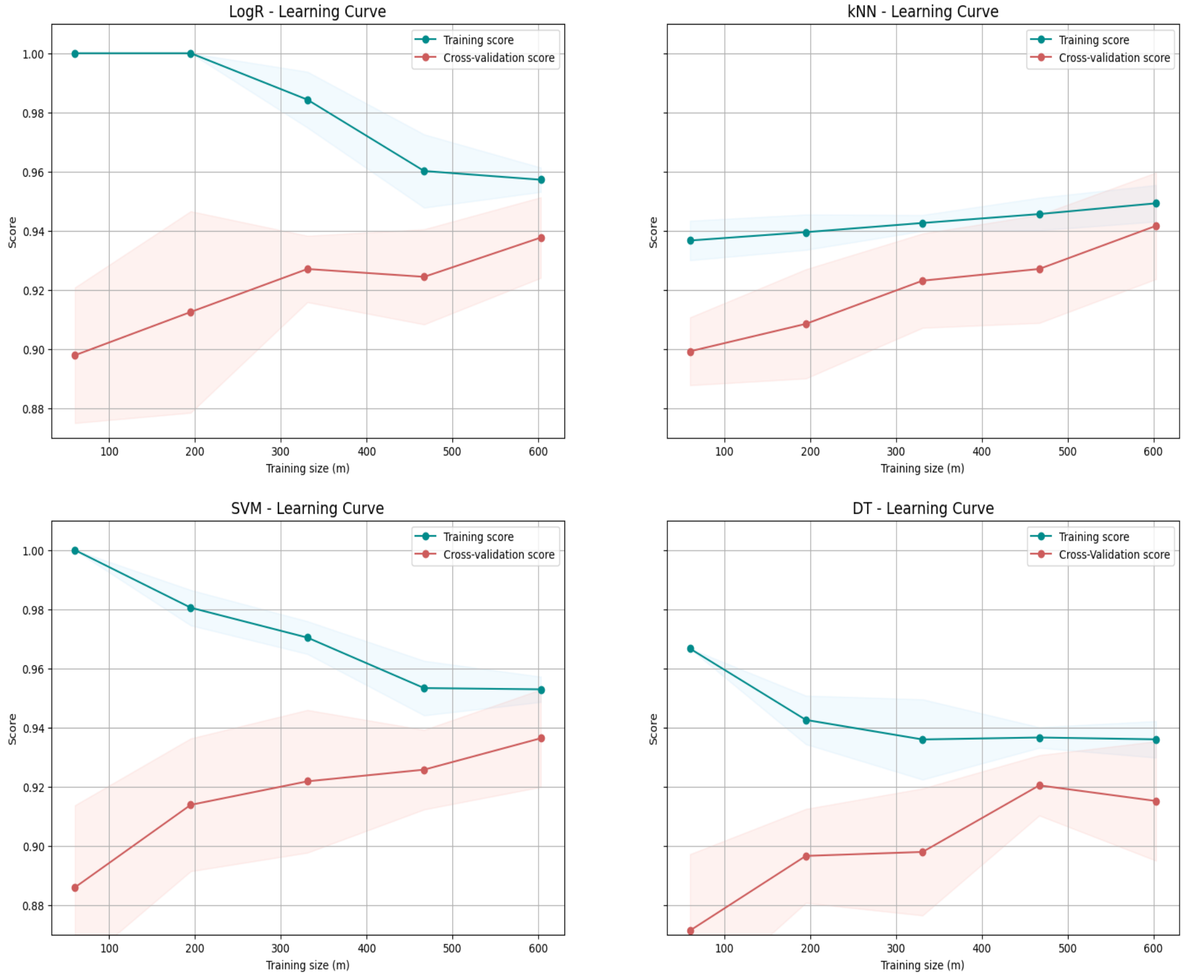


Fig. 7. Learning curves of the classifiers.

TABLE II. Training Performance of Sampling Trade-off before SMOTE

Experiment	Algorithms	Training sample size				
		60	197	333	470	607
Training score (Accuracy)	LogR	0.996666	0.958375	0.951951	0.949787	0.955189
	kNN	0.920000	0.928934	0.948948	0.952765	0.951565
	SVM	1.000000	0.964467	0.965165	0.959148	0.958484
	DT	0.950000	0.938071	0.936336	0.928936	0.930477
5-fold Cross Validation score (Accuracy)	LogR	0.931509	0.940719	0.942044	0.942044	0.944693
	kNN	0.910421	0.919649	0.931483	0.936742	0.928842
	SVM	0.930149	0.907755	0.932816	0.927561	0.939412
	DT	0.851115	0.924904	0.922272	0.920956	0.927535

While experimenting with different sample sizes of the dataset, several interesting observations were made, particularly during training and cross-validation.

The Table II provides insights into the performance of four machine learning algorithms—Logistic Regression (LogR), k-Nearest Neighbors (kNN), Support Vector Machine (SVM), and Decision Tree (DT)—across varying training sample sizes, evaluated through training scores (accuracy) and 5-fold cross-validation scores. From the training score perspective, LogR consistently demonstrated high accuracy across different sample sizes, achieving values ranging from approximately 95% to 99.7%. kNN also performed well, maintaining accuracy levels between 92% and 95%. SVM exhibited near-perfect accuracy (100%) on the smallest training sample and remained consistently high as the sample size increased. DT showed stable performance with accuracy ranging from approximately 93% to 95%.

In terms of 5-fold cross-validation scores, LogR consistently maintained high accuracy, ranging from about 93% to 94.5% across various sample sizes. kNN showed slightly lower but still strong accuracy, ranging from approximately 91% to 93.7%. SVM demonstrated varying accuracy, typically ranging between approximately 90.8% and 93.9%. DT had lower accuracy compared to the other models, with scores ranging from around 85.1% to 92.7%. Overall, the results suggest that SVM and LogR are generally more reliable for this classification task due to their consistently high accuracy across different sample sizes and validation methods. kNN also showed competitive performance but with slight variability, while DT, although effective, exhibited lower accuracy in some cross-validation scenarios. These findings highlight the importance of considering both training and validation scores to assess the robustness and reliability of machine learning models in practical applications.

The performances were studied on various sample sizes in terms of accuracy scores. The average of the AUC-ROC on 5-fold cross-validation is visualized in Fig. 8. LogR and SVM exhibited better performance on cross-validation, irrespective of training sizes.

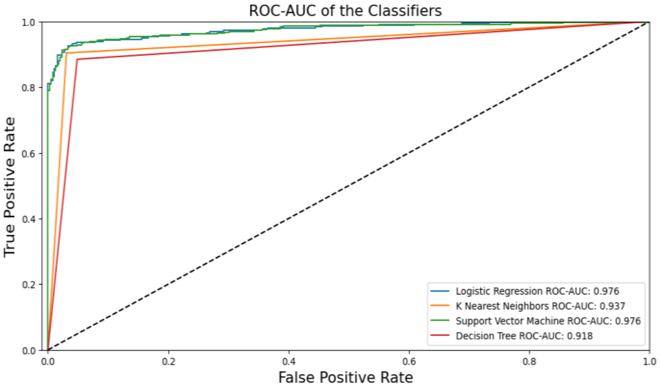
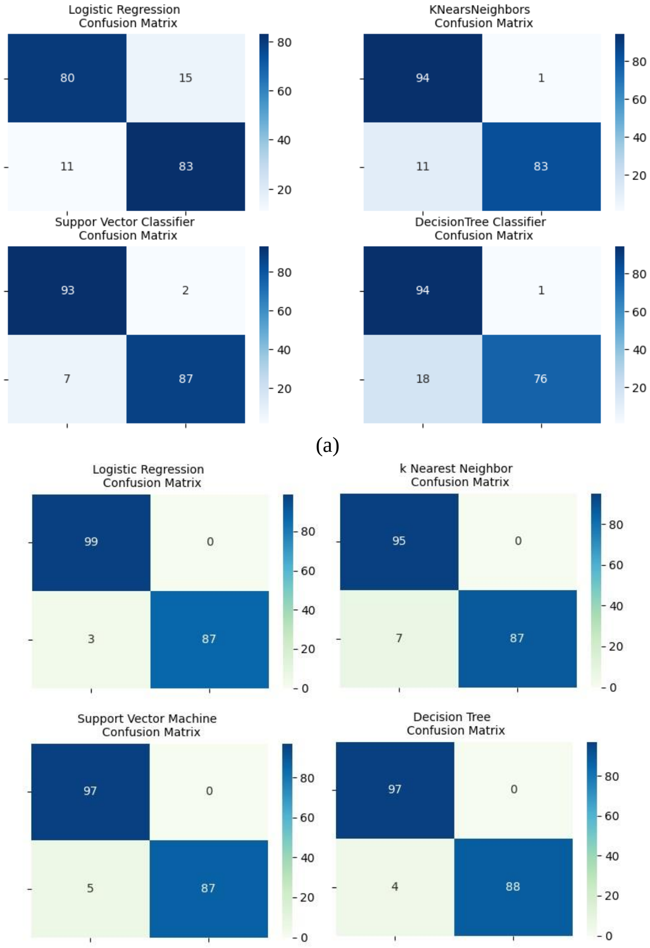


Fig. 8. AUC-ROC of the 5-fold cross validation score before SMOTE.

During this study, underfitting and overfitting issues were observed. Class imbalances are checked and corrected by undersampling. In a few cases, synthetic correction was also experimented with. At each stage of these issues, techniques such as NearMiss and SMOTE are adopted to largely handle them. After upsampling with the SMOTE method, significant improvements were noticed in the testing performance. After this trade-off, the test samples that were never used during training are used to test the performance of the optimized classifiers. The final classification test results in the form of confusion matrix before and after the trade-off are shown in Fig. 9.



(a)

(b)

Fig. 9. (a). Performance before NearMiss/SMOTE trade-off using test set, (b). Performance after NearMiss/SMOTE trade-off using test set.

According to the confusion matrix, the sampling trade-off significantly boosted the algorithms' ability to detect fraudulent credit card transactions. The classification accuracy of the methods in detecting credit card fraud is increased as follows: from 0.862 to 0.984 (LogR), from 0.936 to 0.963 (kNN), from 0.952 to 0.974 (SVM) and 0.899 to 0.979 (DT).

V. Conclusion

Anomalies in credit card and other financial transactions are becoming more common as the user base expands. While financial institutions use a variety of methods to detect these irregularities, the fraud rate has not decreased significantly. Financial institutions are strengthening their fraud detection capabilities with the help of AI and other cutting-edge technologies in order to avoid such fraudulent acts and help users feel comfortable during transactions. Therefore, payment fraud remains a major concern, and taking precautions to safeguard customers and their financial data is critical. Taking advantage of anti-fraud tools, as well as their continuous enhancement and development of new approaches and technologies, is critical to combating payment fraud. The research framework brought out here is one attempt to deal with this issue. In this research, existing ML and DL-based credit card fraud detection methods were reviewed, and a comprehensive ML-based method for detecting credit card fraud was proposed by considering the gaps in the existing literature. Several micro-level approaches were adopted, especially in handling the dataset through sampling trade-offs. While dealing with anomaly detection problems using ML, inefficiencies are usually encountered due to ineffective ways of handling the data. This is smartly addressed in this framework. Significant results were achieved on testing the framework, and it is strongly recommended for the prospective ML of DL-based anomaly detection frameworks.

Despite the promising results achieved by the proposed ML-based credit card fraud detection framework, several limitations remain. One significant limitation is the dependency on the quality and quantity of the dataset. Imbalanced datasets can still pose challenges, potentially leading to biased models that favor the majority class. Additionally, while the framework addresses some inefficiencies in data handling, there remains room for improvement in the preprocessing and feature engineering stages to enhance the detection capabilities further. Future research could explore the integration of more advanced techniques such as ensemble learning and hybrid models that combine both ML and DL approaches to improve detection accuracy. Furthermore, the incorporation of real-time data streams and adaptive learning methods can help in developing more robust and responsive fraud detection systems. Investigating the use of explainable AI (XAI) techniques would also be beneficial to provide transparency and interpretability in fraud detection models, thus increasing trust and adoption by financial institutions.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, author-ship, and/or publication of this article.

References

[1] Card Fraud Losses Worldwide, Nilson report: https://nilsonreport.com/mention/1750/1link/.

[2] The Federal Trade Commission (FTC) USA. https://www.ftc.gov/policy-notices/open-government/data-sets.

[3] Arabian Business. Wed 18 May 2022. https://www.arabianbusiness.com/industries/technology/cybercriminals-are-targeting-financial-institutions-in-the-kingdom-of-saudi-arabia.

[4] Papernot, N., McDaniel, P., & Goodfellow, I. (2016). Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*.

[5] Seera, M., Lim, C. P., Kumar, A., Dhamotharan, L., & Tan, K. H. (2024). An intelligent payment card fraud detection system. *Annals of Operation Research/Annals of Operations Research*. https://doi.org/10.1007/s10479-021-04149-2.

[6] Charizanos, G., Demirhan, H., & İçen, D. (2024). An online fuzzy fraud detection framework for credit card transactions. *Expert Systems With Applications*, 124127. https://doi.org/10.1016/j.eswa.2024.124127.

[7] Alfaiz, N. S., & Fati, S. M. (2022). Enhanced credit card fraud detection model using machine learning. *Electronics*, *11*(4), 662.

[8] Alharbi, A., Alshammari, M., Okon, O. D., Alabrah, A., Rauf, H. T., Alyami, H., & Meraj, T. (2022). A novel text2IMG mechanism of credit card fraud detection: a deep learning approach. *Electronics*, *11*(5), 756.

[9] Jovanovic, D., Antonijevic, M., Stankovic, M., Zivkovic, M., Tanaskovic, M., & Bacanin, N. (2022). Tuning machine learning models using a group search firefly algorithm for credit card fraud detection. *Mathematics*, *10*(13), 2272.

[10] Malik, E. F., Khaw, K. W., Belaton, B., Wong, W. P., & Chew, X. (2022). Credit card fraud detection using a new hybrid machine learning architecture. *Mathematics*, *10*(9), 1480.

[11] Stojanović, B., Božić, J., Hofer-Schmitz, K., Nahrgang, K., Weber, A., Badii, A., ... & Runevic, J. (2021). Follow the trail: Machine learning for fraud detection in Fintech applications. *Sensors*, *21*(5), 1594.

[12] Mekterović, I., Karan, M., Pintar, D., & Brkić, L. (2021). Credit card fraud detection in card-not-present transactions: Where to invest?. *Applied Sciences*, *11*(15), 6766.

[13] Soleymanzadeh, R., Aljasim, M., Qadeer, M. W., & Kashef, R. (2022). Cyberattack and Fraud Detection Using Ensemble Stacking. *AI*, *3*(1), 22-36.

[14] Chaquet-Ulldemolins, J., Gimeno-Blanes, F. J., Moral-Rubio, S., Muñoz-Romero, S., & Rojo-Álvarez, J. L. (2022). On the Black-Box Challenge for Fraud Detection Using Machine Learning (II): Nonlinear Analysis through Interpretable Autoencoders. *Applied Sciences*, *12*(8), 3856.

[15] Strelcenia, E., & Prakoonwit, S. (2023). Improving Classification Performance in Credit Card Fraud Detection by Using New Data Augmentation. *AI*, *4*(1), 172-198.

[16] Ashfaq, T., Khalid, R., Yahaya, A. S., Aslam, S., Azar, A. T., Alsafari, S., & Hameed, I. A. (2022). A Machine Learning and Blockchain Based Efficient Fraud Detection Mechanism. *Sensors*, *22*(19), 7162.

[17] Moschini, G., Houssou, R., Bovay, J., & Robert-Nicoud, S. (2021). Anomaly and fraud detection in credit card transactions using the arima model. *Engineering Proceedings*, *5*(1), 56.

[18] Jiang, J. R., Kao, J. B., & Li, Y. L. (2021). Semi-supervised time series anomaly detection based on statistics and deep learning. *Applied Sciences*, *11*(15), 6698.

[19] Zioviris, G., Kolomvatsos, K., & Stamoulis, G. (2022). Credit card fraud detection using a deep learning multistage model. *The Journal of Supercomputing*, *78*(12), 14571-14596.

[20] Mehbodniya, A., Alam, I., Pande, S., Neware, R., Rane, K. P., Shabaz, M., & Madhavan, M. V. (2021). Financial fraud detection in healthcare using machine learning and deep learning techniques. *Security and Communication Networks*, *2021*, 1-8.

[21] Sanober, S., Alam, I., Pande, S., Arslan, F., Rane, K. P., Singh, B. K., ... & Shabaz, M. (2021). An enhanced secure deep learning algorithm for fraud detection in wireless communication. *Wireless Communications and Mobile Computing*, *2021*, 1-14.

[22] Seeja, K. R., & Zareapoor, M. (2014). Fraudminer: A novel credit card fraud detection model based on frequent itemset mining. *The Scientific World Journal*, *2014*.

[23] Xie, Y., Li, A., Gao, L., & Liu, Z. (2021). A heterogeneous ensemble learning model based on data distribution for credit card fraud detection. *Wireless Communications and Mobile Computing*, *2021*, 1-13.

[24] Karthika, J., & Senthilselvi, A. (2023). Smart credit card fraud detection system based on dilated convolutional neural network with sampling technique. *Multimedia Tools and Applications*, 1-18.

[25] Mathew, D. T. E. (2023). An Ensemble Machine Learning Model for Classification of Credit Card Fraudulent Transactions. *Journal of Theoretical and Applied Information Technology*, *101*(9).

[26] Mienye, I. D., & Sun, Y. (2023). A Deep Learning Ensemble With Data Resampling for Credit Card Fraud Detection. *IEEE Access*, *11*, 30628-30638.

[27] IEEE Computational Intelligence Society. IEEE-CIS Fraud Detection Can You Detect Fraud from Customer Transactions? 2019. Available

online: https://www.kaggle.com/c/ieee-fraud-detection/overview (accessed on 30 May 2023).

[28] Gisbrecht, A., Schulz, A., & Hammer, B. (2015). Parametric nonlinear dimensionality reduction using kernel t-SNE. *Neurocomputing*, *147*, 71-82.

[29] Reddy, G. T., Reddy, M. P. K., Lakshmanna, K., Kaluri, R., Rajput, D. S., Srivastava, G., & Baker, T. (2020). Analysis of dimensionality reduction techniques on big data. *Ieee Access*, *8*, 54776-54788.

[30] Feng, J., Xu, H., Mannor, S., & Yan, S. (2014). Robust logistic regression and classification. *Advances in neural information processing systems*, *27*.

[31] Cunningham, P., & Delany, S. J. (2021, July 13). k-Nearest Neighbour Classifiers - A Tutorial. *ACM Computing Surveys*, *54*(6), 1–25.

[32] Suthaharan, S., & Suthaharan, S. (2016). Support vector machine. Machine learning models and algorithms for big data classification: thinking with examples for effective learning, 207-235.

[33] Gordon, A. D., Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984, September). Classification and Regression Trees. *Biometrics*, *40*(3), 874.

[34] Bao, L., Juan, C., Li, J., & Zhang, Y. (2016). Boosted near-miss under-sampling on SVM ensembles for concept detection in large-scale imbalanced datasets. *Neurocomputing*, *172*, 198-206.

[35] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-3.