I know you're a fraud: Uncovering illicit activity in a Greek bank transactions with unsupervised learning

Karnavou, E.; Cascavilla, G.; Marcelino, G.; Geradts, Ze.

DOI
10.1016/j.eswa.2025.128148
Publication date
2025
Document Version
Final published version
Published in
Expert Systems With Applications

Citation for published version (APA):
Karnavou, E., Cascavilla, G., Marcelino, G., & Geradts, Z. (2025). *I know you're a fraud*: Uncovering illicit activity in a Greek bank transactions with unsupervised learning. *Expert Systems With Applications*, *288*, Article 128148. https://doi.org/10.1016/j.eswa.2025.128148

Expert Systems With Applications 288 (2025) 128148

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

I know you're a fraud: Uncovering illicit activity in a Greek bank transactions with unsupervised learning

Evgenia Karnavou [a], Giuseppe Cascavilla [b,*], Gonçalo Marcelino [c], Zeno Geradts [c]

[a] Jheronimus Academy of Data Science, the Netherlands
[b] Tilburg University, the Netherlands
[c] University of Amsterdam, the Netherlands

ARTICLE INFO

Keywords:
Cybersecurity
Fraud detection
Bank trasactions
Unsupervised learning
Machine learning
Explainable AI

ABSTRACT

In recent years, the banking industry has undergone a significant transformation driven by the digitization of payments, which has led to a rise in financial and credit card fraud. Financial institutions face tremendous losses yearly in trying to mitigate this threat. Machine and deep learning techniques have become the spotlight in this ongoing battle, emerging as a very promising solution that many financial institutions are now actively adopting. Supervised methods (such as SVM, random forest, Markov models, etc.) have been widely studied in credit card fraud detection scenarios. However, their effectiveness is hindered by the need for labeled instances, which are both expensive and difficult to come by, and their lower performance when attempting to detect unknown attack patterns. On the other hand, unsupervised techniques do not require labeled instances and have proven to be effective in discovering new and evolving fraud patterns. This research paper employs a confidential industrial dataset composed of transactions made by cardholders of a Greek bank. We utilize two machine learning models, Isolation Forest, One-class SVM, and a deep neural autoencoder, to detect unsupervised anomalies in credit card data. This work introduces a novel approach to performing anomaly detection since our two machine learning models are trained on a per-cardholder basis, where a separate model is trained for each cardholder in our dataset. Moreover, the autoencoder is employed to identify frauds across the entire dataset, effectively acting as a generalized anomaly detection model. Finally, we examine the use of genetic algorithms for feature selection and additionally incorporate Shapley Additive Explanations (SHAP) to promote interpretability and transparency. Experimental results demonstrate the superiority of the autoencoder in comparison to the machine learning models in this anomaly detection scenario and stress the benefits of applying explainability tools such as SHAP to industrial models.

1. Introduction

The rise of online banking and digital payments has transformed the financial sector, offering convenience but also attracting cybercriminals, leading to increased fraud. Financial fraud, including identity theft and counterfeit cards, undermines trust and causes significant monetary losses, with credit card fraud losses projected to reach $6.8 billion by 2027 (ACI Worldwide, 2024).

The rapid digitization of financial transactions has exposed vulnerabilities, with fraudsters exploiting online systems to illegally obtain sensitive information. Various forms of fraud, including identity theft, counterfeit cards, and 'cardholder not present' fraud, have emerged as major threats (Bhattacharyya, Jha, Tharakunnel, & Westland, 2011; Cascavilla, 2024; Hilal, Gadsden, & Yawney, 2022).

Advancements in artificial intelligence (AI) offer promising solutions by enabling real-time analysis of financial data and detecting complex fraud patterns. Unsupervised learning models, which identify anomalies without labeled data (Pymnts, 2024), are particularly valuable in dynamic environments. As financial institutions invest in AI-driven fraud prevention, these technologies are becoming essential for enhancing security and maintaining trust (Pymnts, 2021, 2024).

This research was conducted in collaboration with a well-established Greek financial institution operating at a European level, anonymized in this study to preserve confidentiality. The dataset contains numerous variables describing the characteristics and transactional behaviors of unique clients. The raw dataset contains 6.915.699 transactions and 30 variables.

* Corresponding author.
E-mail addresses: jennykrnv@gmail.com (E. Karnavou), g.cascavilla@tilburguniversity.edu (G. Cascavilla), g.barretoferreiramarcelino@uva.nl (G. Marcelino), z.j.m.h.geradts@uva.nl (Z. Geradts).

https://doi.org/10.1016/j.eswa.2025.128148
Received 25 November 2024; Received in revised form 14 April 2025; Accepted 10 May 2025
Available online 12 May 2025

This work uses unsupervised learning to develop adaptable, real-time anomaly detection systems for banking. It implements two strategies:

1. First, anomaly detection is performed on credit card data on a per-cardholder basis, where a separate model is trained for each individual cardholder. This personalized approach allows for highly specific anomaly detection tailored to the unique spending patterns of each cardholder.
2. Second, a generalized anomaly detection model is created to identify anomalies across the entire dataset, encompassing all cardholders.

This twofold method represents a more comprehensive approach to anomaly detection than that of the methods previously proposed in literature, as it allows for the detection of a broader range of anomalies in the context of credit card fraud.

Unsupervised models such as Isolation Forest and One-class Support Vector Machine (SVM) have shown effectiveness in fraud detection, particularly in credit card data (Hejazi & Singh, 2013; Hilal et al., 2022; Liu, Ting, & Zhou, 2008; Sahin & Duman, 2011; Schölkopf, Williamson, Smola, Shawe-Taylor, & Platt, 1999). Given their success in identifying rare events in unbalanced datasets, this work seeks to assess their performance through the first research question (RQ), which is defined as follows:

RQ1: How accurately can Isolation Forest and One-class SVM identify fraudulent transactions on a per-cardholder basis?

Beyond traditional machine learning, deep learning models, particularly autoencoders (AEs), have demonstrated significant advantages in anomaly detection, especially with large datasets (Hilal et al., 2022). In this study, we employ an autoencoder architecture for general anomaly detection. This choice is motivated by the fact that autoencoders require a substantial amount of data to achieve high efficiency (Bakumenko & Elragal, 2022). Given that data availability on a per-cardholder basis may be limited, using an autoencoder trained on the entire volume of transactions ensures adequate data for training the model. This leads to the second research question:

RQ2: To what extent do deep learning methods, specifically autoencoders, outperform traditional machine learning methods on generalized anomaly detection?

Feature selection is another critical factor in improving model performance. Genetic algorithms (GAs) have emerged as powerful tools for feature selection due to their ability to navigate complex search spaces efficiently (Brill, Brown, & Martin, 1992; Hilal et al., 2022). The application of GAs in optimizing fraud detection systems prompts the third research question:

RQ3: To what extent can genetic algorithms be used for feature selection to improve the performance of the chosen models?

Model interpretability remains a significant challenge, particularly with complex AI systems. Lack of transparency hinders adoption in the financial sector, where understanding model decisions is crucial (Hilal et al., 2022). Experts stress that the black-box nature of AI systems explains the slow adoption at the business level (Pymnts, 2024). Decision-makers do not always understand why AI systems make certain decisions. 60 % of bank fraud specialists believe that AI technology lacks transparency, and an equal portion view it as complex and time-consuming (Pymnts, 2024). Explainable AI techniques, such as Local Interpretable Model-Agnostic Explanations (LIME) (Ribeiro, Singh, & Guestrin, 2016) and SHAP (Shapley Additive Explanations). SHAP offers a solution by providing local and global explanations for the model's prediction (Lundberg & Lee, 2017). This leads to the fourth and final research question:

RQ4: How can explainable AI be incorporated to interpret the proposed models' predictions?

The contribution of this paper is multifold:

- **Real-world dataset**: We employ a large-scale, real-world dataset from the financial sector, enhancing the relevance and practical applicability of the findings.
- **Dual anomaly detection approach**: We introduce a novel dual approach to credit card fraud detection, combining individual cardholder models with generalized models to capture both personal and global anomalies.
- **Explainable AI**: We address the issue of model interpretability in fraud detection by employing SHAP values, which enhance transparency and contribute to the development of explainable, high-performing AI systems.
- **Feature engineering**: Advanced feature engineering techniques are developed, leveraging domain expertise to improve the detection of subtle fraud patterns, further emphasizing the importance of expert collaboration in anomaly detection tasks.

The rest of this work is organized as follows: Section 2 introduces key concepts and foundational theories in the field of unsupervised anomaly detection. Section 3 reviews the relevant literature on systems and methodologies for fraud detection, with an emphasis on credit card and financial fraud. Section 4 details the data preprocessing pipeline, starting from a dataset description to the intricacies of feature engineering. Section 5 describes the experimental setup and evaluates the chosen approaches for unsupervised anomaly detection. Section 6 discusses the results, providing a thorough assessment of the models' performance. In Sections 8 and 9, we conclude our analysis with the contribution of the current work, the threats to validity, and future steps.

2. Background

This section provides an overview of the key concepts in unsupervised anomaly detection. The goal is to help readers better understand these concepts and the research topic.

2.1. Time series

A time series is a sequence of data points collected at regular intervals over time, capturing changes in a system. These sequences provide insight into the temporal dynamics of various processes, such as server temperature variations or stock price fluctuations. A time series consists of observations indexed by time, offering valuable information about system behavior.

Univariate vs. Multivariate time series. In univariate time series, a single variable is recorded over time. In contrast, multivariate time series involve multiple interdependent variables evolving simultaneously. Formally, a univariate time series is represented as:

$$\mathcal{T} = \{x_1, \dots, x_T\}$$

where x_T denotes an observation at time $\mathcal{T}$. A multivariate time series can be expressed as:

$$\mathcal{T} = \{x_1, \dots, x_T\}, \quad x \in \mathbb{R}^m$$

Where x_T is a vector in m-dimensional real space, representing multiple variables at each time point.

Anomaly detection in time series has broad applications, from fraud detection to network security and medical diagnostics. It involves identifying deviations from normal temporal patterns, often indicating unusual events or system failures (Manish, Jing, Aggarwal Charu, & Jiawei, 2014). Fox's seminal work on univariate time series anomalies distinguishes two types of outliers: Type I, which affects single observations, and Type II, which involves one or more consecutive observations (Fox, 1972).

For example, a credit card transaction far exceeding typical amounts (e.g., a €5,000 purchase instead of usual €100 transactions) may signal fraudulent activity (Type I outlier, unusual but legitimate purchase). If further suspicious transactions follow this transaction (i.e., from Europe to Asia), it suggests a more complex pattern, possibly indicating a compromised account (Type II outlier).

Tsay (1988) later expanded this categorization to multivariate time series, refining the classification of outliers across multiple dimensions. Despite extensive research, consensus on a precise definition of outliers remains elusive. One widely accepted definition by Hawkins describes an outlier as "an observation which deviates so much from other observations as to arouse suspicions that a different mechanism generated it".

Outliers can be noise or critical signals, depending on the context. Data scientists often remove outliers to improve data quality, but identifying the outlier itself is the main objective in cases like fraud detection.

Anomalies in time series are generally classified into three categories: point anomalies, collective anomalies, and contextual anomalies (Braei & Wagner, 2020; Chalapathy & Chawla, 2019; Chandola, Banerjee, & Kumar, 2009).

Point anomalies are individual observations that deviate significantly from the rest of the dataset. These anomalies are akin to Type I outliers and are commonly detected in fraud detection. For instance, a credit card transaction substantially higher than the norm might suggest suspicious activity.

Collective anomalies arise when a group of observations exhibits an unusual pattern, even though individual points may not appear abnormal in isolation. These anomalies typically occur in temporal or spatial datasets (Chandola et al., 2009). For example, a series of credit card transactions in different geographical locations within a short period may collectively suggest fraud, though each transaction alone seems normal.

Contextual anomalies are data points that are considered anomalies within a specific context, often related to time or location. In fraud detection, a transaction occurring at an unusual time for a particular user might be classified as a contextual anomaly. For example, a purchase at 3 AM might be suspicious for one customer but routine for another, depending on individual patterns.

2.2. Fraud detection in practice

Fraud detection methods have evolved significantly due to the adaptive nature of fraudulent activities. In banking, monitoring user behavior through profiling was once widely used but became impractical with the exponential growth of users. Real-time systems are now essential. This study, conducted in collaboration with a fully cloud-based Greek neobank operating at the European level, provides insights into practical credit card fraud detection and presents common anomaly detection techniques based on expert interviews.

Rule-based systems: Real-time and near real-time fraud detection. Credit card fraud detection generally involves two components: *blocking rules*, which prevent fraudulent transactions in real-time, and *checking rules*, which assess suspicious activities post-authorization (Lucas, 2019). Blocking rules must be both fast and precise to avoid interrupting legitimate transactions (real-time), emphasizing precision over recall (Baesens, Van Vlasselaer, & Verbeke, 2015). Checking rules operate in near real-time, flagging transactions for manual review if deemed suspicious.

The Greek bank considered in this study employs a rule-based fraud detection system that monitors transactions for predefined indicators, such as unusual locations or sudden spikes in activity. While this approach is effective for well-known fraud patterns, it often results in high false positives and is less effective against new fraud tactics. Thus, more adaptive models are needed to reduce financial losses from manual investigations and false alarms.

Traditional rule-based systems are static and limited in detecting evolving fraud schemes. In contrast, machine learning (ML) models, including supervised, semi-supervised, and unsupervised techniques, offer more adaptive, data-driven approaches (Roy et al., 2018; Sengupta et al., 2020). This study focuses on unsupervised learning, which detects anomalies without labeled data by identifying deviations from expected patterns. Such methods are especially relevant for fraud detection due to the class imbalance, where fraudulent transactions constitute a small fraction of the overall dataset.

2.3. Models

This section provides an overview of the models applied in this research, specifically Isolation Forest, One-Class Support Vector Machine (SVM), and a deep neural autoencoder.

Isolation Forest: Isolation Forest isolates anomalies by randomly partitioning data, relying on the principle that anomalies are fewer and more isolated than normal points (Liu et al., 2008). By using decision trees, it partitions the data by selecting random features and split values. Anomalies, due to their distinct characteristics, are typically isolated near the root of the tree, while normal data points require more partitions to be isolated. Isolation Forest offers several advantages over other methods: it efficiently handles high-dimensional data, uses subsampling to reduce complexity, and avoids computationally expensive distance or density calculations. With its linear time complexity and minimal memory requirements, it scales well to large datasets, outperforming many distance- and density-based approaches (Wu & Jermaine, 2006).

One Class SVM: One-Class SVM (Schölkopf et al., 1999) is designed for anomaly detection when only one class of data (typically "normal" instances) is available. It projects the data into a higher-dimensional space where a linear classifier can separate normal data from anomalies. By applying a kernel function, such as a linear, polynomial, or radial basis function (RBF) kernel, One-Class SVM makes the data linearly separable in this feature space.

Unlike traditional SVM, which separates different classes, One-Class SVM constructs the smallest possible sphere that encloses normal data. Any data point outside this boundary is considered an anomaly. This method is particularly useful in datasets dominated by normal instances, allowing for outlier detection without requiring explicit examples of anomalies during training.

Autoencoders: Autoencoders are unsupervised deep learning models widely used for anomaly detection due to their ability to handle high-dimensional and noisy data (Pang, Shen, Cao, & Hengel, 2021). They consist of two main components: an encoder, which compresses the data into a compact representation, and a decoder, which reconstructs the original input from this compressed form. By minimizing reconstruction error, autoencoders learn to represent normal patterns in the data. Anomalies, which deviate significantly from normal patterns, exhibit higher reconstruction errors.

Autoencoders address several challenges in anomaly detection, such as low recall rates, high dimensionality, and complex feature relationships (Nonnenmacher & Gómez, 2021). However, they require sufficient training data to be effective (Bakumenko & Elragal, 2022). This makes them particularly powerful in domains with large datasets and complex, interrelated features.

2.4. Evaluation metrics

The confusion matrix is a widely adopted method for evaluating machine learning classifiers, which represents the discrepancies between the ground truth and the model's predictions. An ideal model would perfectly classify all instances, resulting in 100 % true positives (TP) and 0 % false positives (FP). The confusion matrix facilitates the calculation of various metrics that assess and compare model performance (see Table 1).

Precision measures the proportion of correctly predicted positive instances out of all instances predicted as positive:

$$\text{precision} = \frac{TP}{TP + FP}$$

Recall, also known as the true positive rate (TPR), quantifies the proportion of actual positive instances correctly identified by the model:

$$\text{recall} = \frac{TP}{TP + FN}$$

False Positive Rate (FPR) evaluates the proportion of negative instances incorrectly predicted as positive:

$$\text{FPR} = \frac{FP}{FP + TN}$$

Accuracy reflects the overall proportion of correctly classified instances:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

F1 Score represents the harmonic mean of precision and recall, offering a balanced measure for imbalanced datasets:

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP + FP + FN}$$

AUC Score (Area Under the Curve) evaluates the model's discriminative power by calculating the area under the Receiver Operating Characteristic (ROC) curve, which plots TPR against FPR across different thresholds. An AUC score of 1 indicates perfect classification, while a score of 0.5 signifies random guessing:

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR}^{-1}(x))\, dx$$

In the context of fraud detection in banking, such as in highly imbalanced datasets, evaluating solely with precision, recall or accuracy often proves inadequate. Since legitimate transactions vastly outnumber fraudulent ones, a model can achieve high accuracy by predicting the majority class, but this fails to reflect its ability to detect fraud. Instead, the primary challenge lies in balancing the true positive rate (TPR) with the false negative rate (FNR). Maximizing TPR while minimizing FPR is crucial, as a high false positive rate leads to an overwhelming number of false alerts, straining the resources of manual review teams. This balance is essential for effective fraud detection systems, as it directly impacts the capacity to identify fraudulent transactions without overburdening the system with false alarms. Thus, metrics beyond accuracy, particularly the TPR and FPR, are critical in assessing and optimizing the performance of fraud detection models in real-world applications.

2.5. Genetic algorithms

Genetic algorithms (GAs) are a type of evolutionary algorithm (EA) inspired by Darwin's theory of natural selection, emphasizing survival of the fittest (Holland, 1992). GAs offer a powerful approach for exploring large search spaces and identifying high-quality solutions, outperforming traditional optimization techniques in some contexts (Holland, 1992).

The genetic algorithm operates through iterative steps that transform a population of chromosomes (potential solutions) into successive generations using genetic functions such as crossover and mutation. These operations mimic biological processes, facilitating the evolution of the population toward optimal solutions (Babatunde, Armstrong, Leng, & Diepeveen, 2014). GAs are, therefore, considered randomized search algorithms that use principles of reproduction and mutation to solve complex optimization problems.

The fitness (or goodness) of each candidate solution is evaluated by the fitness function, which assigns a numerical score based on the quality of the solution. The design of this fitness function depends on the specific problem being addressed. In the context of feature selection for machine learning models, GAs provide a robust method for selecting relevant features from large datasets. Here, each chromosome represents a subset of features encoded as a binary vector, where each gene indicates the inclusion (True) or exclusion (False) of a feature. The population evolves over generations, approximating an optimal subset of features. The chromosome with the highest fitness score represents the most effective feature combination for the model.

The key terminology of genetic algorithms are:

- **Population**: The set of candidate solutions from which the genetic algorithm begins. GAs iterate over multiple generations, with the initial population generated randomly.
- **Chromosome**: A single candidate solution in the population. Each chromosome is composed of genes, which represent the values for the optimization variables.
- **Fitness Function**: The mechanism that evaluates the performance of individual chromosomes in each generation, determining which solutions are the fittest.

The genetic operators used for feature selection are:

- **Selection**: The process of selecting the fittest solutions from the population to serve as parents for the next generation. The goal is to ensure that stronger solutions are passed on, improving the overall population. Although various selection methods exist (Babatunde et al., 2014), this study applies the Top-k selection method, in which the top k chromosomes with the highest fitness scores are selected for reproduction.
- **Crossover**: Also known as recombination, this operation exchanges genes between two selected parents to generate new offspring. In this work, a one-point crossover is used, where a single crossover point is chosen, and the offspring are created by combining the first segment of one parent with the second segment of the other. This mechanism helps blend both parents' characteristics, potentially generating improved solutions.
- **Mutation**: After selection and crossover, mutation introduces random alterations to chromosomes by flipping genes (e.g., changing a feature from selected to unselected or vice versa). This step ensures genetic diversity within the population, helping the algorithm explore a broader search space and avoid premature convergence (Babatunde et al., 2014).

In machine learning, genetic algorithms have demonstrated significant potential for feature selection. They enable the identification of optimal feature subsets that improve model performance. By iteratively refining candidate solutions and maintaining genetic diversity, GAs efficiently navigate large search spaces, making them well-suited for complex feature selection tasks.

2.6. Explainable AI

As machine learning techniques become more widely adopted, interpreting model decisions has grown increasingly important for promoting user trust, improving models, and understanding underlying processes (Lundberg & Lee, 2017). In fraud detection, the large number of variables in credit applications can enhance model accuracy, but quantifying the contribution of each feature remains challenging. Molnar (Molnar, 2020) highlights several methods that improve the interpretability of black-box models, with this work focusing on SHAP (SHapley Additive exPlanation) values. SHAP (SHapley Additive exPlanations) is a powerful tool for interpreting complex machine learning models. Shapley values' ability to assess the model's output is praised for being better than several existing methods (Lundberg & Lee, 2017). SHAP, based on cooperative game theory, assigns each feature an importance value for a specific prediction by analyzing all possible feature subsets, ensuring a fair allocation of contributions to the model's output (Lundberg & Lee,

Table 1
Confusion matrix.

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

2017). The method guarantees the sum of SHAP values equals the difference between a prediction and the dataset's average prediction. Its advantages include efficiency, fair distribution of feature effects, and contrastive explanations, allowing comparisons with individual data points rather than just dataset averages (Molnar, 2020). However, SHAP's computational demand often requires approximations for large-scale problems, and calculating Shapley values necessitates access to the data, complicating real-time deployment scenarios (Molnar, 2020).

3. Related work

Statistical and machine learning methods have been widely employed to identify fraudulent behavior by uncovering suspicious patterns in transactional data (Khalid et al., 2024; Sudjianto et al., 2010). Cluster-based methods, such as K-means and histogram-based outlier detection, have shown effectiveness in real-world datasets, demonstrating the potential for real-time application due to their speed (Becirovic, Zunic, & Donko, 2020). The authors employed the K-means algorithm independently on each transaction type to improve clustering outcomes, resulting in varying numbers of clusters for different transaction types. The second proposed method involved utilizing histograms and introducing synthetic outliers to represent anomaly data. The results of the experiments appear vague and unclear. Authors claim that 2 % of the entire dataset could be considered an anomaly using as a parameter a score above 2. As well the study in Huang, Zheng, Li, and Che (2024) demonstrated, using an experimental dataset, that machine learning-based K-means clustering can effectively identify fraudulent financial transactions by detecting unusual patterns and behaviors. However, the challenge of determining the optimal number of clusters (K) in K-means clustering can significantly impact the results. Moreover, the principal component analysis (PCA) may not capture all relevant features for fraud detection. Banerjee, Bourla, Chen, Kashyap, and Purohit (2018) evaluated classification models and highlighted the superior performance of SVM for detecting fraud in imbalanced datasets which can lead to lead to poor generalization and difficulties in training stable models. The paper introduced improved metrics for detecting false negatives and evaluated the effectiveness of random sampling to address dataset imbalance. Far from our specific needs, the paper's authors highlight that Random Forest performs better, with a normal-to-fraudulent distribution of the testing and training set being the same. A more recent work from Hilal et al. (2022) reviewed various anomaly detection techniques and their advancements. The paper focuses on applying machine learning, deep learning, and statistical methods to detect different types of financial fraud. From the study, Random Forest (RF) outperforms other models like SVM and Logistic Regression in detecting credit card fraud, while Support Vector Machine (SVM) performs well on imbalanced datasets. Moreover, from the review, the authors highlight how unsupervised methods are especially popular in fraud detection due to their accuracy and expandability (compared to deep learning models). Hidden Markov Models have also proven efficient for identifying fraud by analyzing deviations in transaction patterns (John & Naaz, 2019; Patel & Kale, 2014). This model offers a high coverage of fraudulent transactions with minimal false alarms while handling large transaction volumes, presenting an efficient method for fraud detection with quick results. By analyzing customers' transaction patterns, deviations from the regular patterns are flagged as potential fraud, simplifying the detection process and reducing complexity. Works from Dhulipudi et al. (2024) and Bhattacharyya et al. (2011) compared multiple approaches for detecting credit card fraud. Dhulipudi et al. compared Random Forest (RF) and Isolation Forest (IF) algorithms to evaluate their ability to detect anomalies within high-dimensional credit card transaction datasets. IF and RF performed the same in handling high-dimensional data and spotting subtle anomalies. However, IF excels at fraud detection due to its speed in separating large datasets and identifying outliers, resulting in high detection accuracy with fewer false positives. Authors in Bhattacharya et al. compared SVM and Random Forest, showing how SVM offered competitive performance compared to other approaches. Differently from our work, the proposed approach from Dhulipudi et al. does not provide information on the dataset used, while the authors in Bhattacharya et al. used a labeled dataset dated 2006 from one country for the experiments. Other studies like those in Arjun, Atlas, Sreenarayanan, Janarthanan, and Arvindhan (2022); Maniraj, Saini, Ahmed, and Sarkar (2019); Negi, Kumar, Raj, Sahana, and Jain (2022); Vijayakumar, Divya, Sarojini, and Sonika (2020) favored Isolation Forest over Local Outlier Factor for precision in detecting anomalies. The results demonstrate that Isolation Forest's precision was higher than Local Outlier Factor.

Recent work has explored deep learning approaches like autoencoders and Generative Adversarial Networks (GANs) to enhance fraud detection, with autoencoders (AE) outperforming traditional methods such as gradient-boosted trees and logistic regression (Ding, Kang, Feng, Peng, & Yang, 2023; Rushin, Stancil, Sun, Adams, & Beling, 2017). GANs have been used to generate synthetic examples of minority classes, improving classifier sensitivity (Fiore, De Santis, Perla, Zanetti, & Palmieri, 2019). This approach aims to enhance the performance of classifiers for improved effectiveness in fraud detection.

Saputra et al. (2019) compared the effectiveness of Decision Tree, Naïve Bayes, Random Forest, and neural networks for fraud detection using a Kaggle dataset. Similarly, authors in Chang, Doan, Di Stefano, Sun, and Fortino (2022) tested Logistic Regression, KNN, Decision Tree, Random Forest, and Autoencoders, using SMOTE (Chawla, Bowyer, Hall, & Kegelmeyer, 2002) to address imbalanced data, revealing that neural networks achieved the highest accuracy (96 %), followed by Random Forest (95 %). While SMOTE enhanced performance metrics, the dataset used in these studies did not fully represent e-commerce platforms. Khalid et al. (2024) introduced a Weighted Average Ensemble model combining Logistic Regression, Random Forest, KNN, Adaboost, and Bagging, achieving 99 % accuracy in detecting credit card fraud, surpassing individual models, but reproducibility issues arose due to undocumented feature selection. Prasad, Chowdarv, Bavitha, Mounisha, and Reethika (2023) proposed an ensemble technique using Decision Tree, XGBoost, Logistic Regression, Random Forest, and SVM, with Decision Tree achieving 100 % recall. However, despite improvements in system efficiency, the ensemble model showed lower overall performance.

After conducting a thorough literature review and analyzing the specific requirements outlined by the Greek bank, we selected Isolation Forest, One-Class SVM, and a deep neural autoencoder as the methodologies for our experiments. The Greek bank emphasized the need for a lightweight real-time fraud detection system that achieves high detection accuracy and operates without relying on supervised learning or manual labeling processes. The literature supports our choice, as Isolation Forest, One-Class SVM, and deep neural autoencoders are widely recognized as effective and commonly used approaches for unsupervised anomaly detection. Moreover, differently from most of the papers from the literature, we have no label on fraudulent profiles for our data. Conversely, we only have alerts on possible suspicious activities, making the fraud detection activity a more complex task. These methods align well with the stated objectives, offering a balance between computational efficiency, scalability, and the ability to identify fraudulent transactions with minimal human intervention.

4. Methodology

This section outlines the data preprocessing pipeline, focusing on the key steps of business and data understanding, followed by feature engineering. The methodology follows the Cross-Industry Standard Process for Data Mining (CRISP-DM) as described by Chapman et al. (2000), emphasizing the first three phases: Business Understanding, Data Understanding, and Data Preparation. CRISP-DM offers a cyclical, iterative process essential for effective data handling and assessing potential limitations. An overview of our methodology can be seen in Fig. 1.

Business Understanding involves defining the project's objectives from a business perspective and formulating a plan to achieve them.

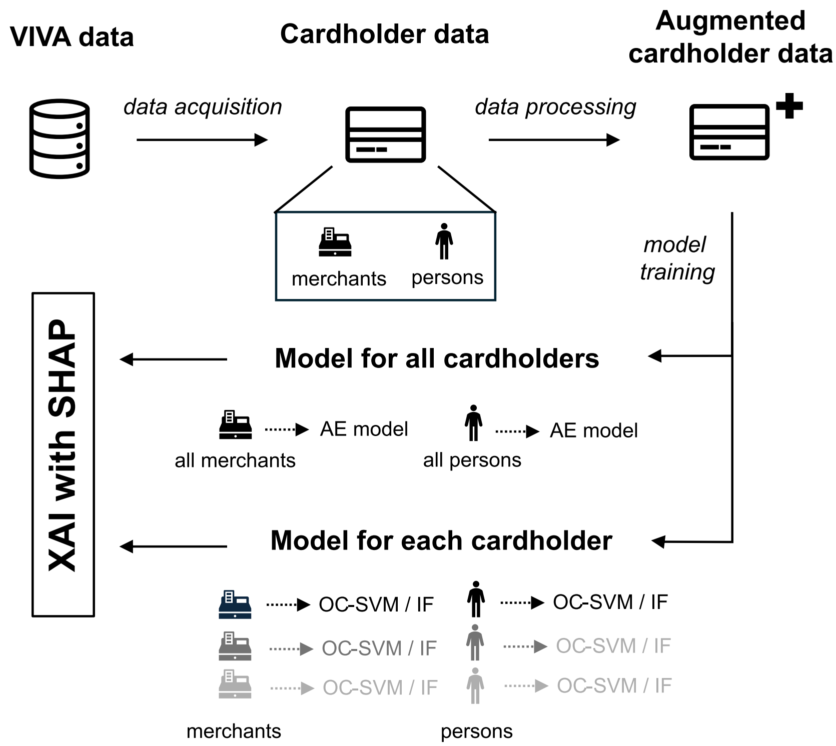

Fig. 1. Methodology employed throughout this paper = .

In partnership with a Greek financial institution operating as both an acquiring and issuing bank, the objective was to develop a fraud detection system based on its issuing data. An acquiring bank processes merchant credit and debit card transactions, while an issuing bank authorizes transactions and manages cardholder accounts. This proposed fraud detection system is needed to analyze cardholder transaction patterns to detect fraudulent activities effectively.

4.1. Data aquisition pipeline

To develop a fraud detection system for the Greek bank's issuing transactions, we required data reflecting cardholder behavior, including transactional patterns and personal characteristics. Due to the large data volume and lack of labeled instances, we focused on cardholders with alerts, as these were the only transactions with available labels. Although the system is unsupervised, these labels were essential for performance evaluation. Since labels were embedded in HTML scripts rather than stored in a dedicated table, we developed a pipeline to extract and structure the data for analysis.

Step 1: **HTML Parsing for Label Extraction**.

Transaction IDs and resolution labels were extracted from HTML-embedded alert descriptions using Python-based regular expressions. This automated parsing resolved label scarcity by associating transactional anomalies with their corresponding alerts, hence constructing a structured dataset.

Step 2: **Cardholder Identification**.

To enable per-cardholder anomaly detection, transaction IDs were mapped to unique cardholder IDs. An SQL query extracted these IDs, ensuring a localized approach to fraud detection. The company emphasized this need, as they recognized that to capture contextual and localized anomalies, a finer-grained system tailored to each cardholder's behavior was needed.

Step 3: **Temporal Data Segmentation.**

We extracted behavioral and transactional data for cardholders with at least one alert, focusing on recent data (January 2021-April 2024) to reflect current spending behavior. Due to data volume constraints, we generated "issuing snapshots" at six-month intervals instead of a single large dataset.

Step 4: **Creating the finalized dataframe.**

Issuing snapshots were concatenated, and resolution types were appended to transactions. Unlabeled transactions were assumed normal. Two datasets were created: one for merchant cardholders (who benefit from lower commission fees) and another for non-merchants. This separation ensured improved detection accuracy, as transaction behavior varies significantly between these groups.

Table 2
Transaction type counts and descriptions.

TransactionTypeId	Description	Count
0	Purchase	6,422,650
1	Withdrawal	379,083
9	Purchase with Cash Back	836
20	Refund	86,465
28	Payment	2074
18	Cash Advance	24,305
12	Recurring Payment	286

4.2. Dataset overview & feature explanation

During the data understanding phase, we familiarize ourselves with the dataset, assess its quality, and conduct exploratory data analysis (EDA) to uncover initial insights and identify potential issues that could affect subsequent modeling (Chapman et al., 2000).

The dataset comprises 6.915.699 million issuing transactions, including both categorical and numerical features such as 'MerchantId', 'PersonId', 'TransactionTypeId', and 'Amount' alongside detailed classification labels for fraud analysis. The dataset encompasses seven types of transactions (e.g., Purchase, Withdrawal, Refund) as outlined by business experts from the Greek bank, as shown in Table 2.

Before preprocessing, the dataset included 30 columns with two key fraud resolution labels: 'LocalizedLabel' (Table 3) from the company's

Table 3
Counts of fraud labels in LocalizedLabel.

LocalizedLabel	Count	Description
No Fraud	25,099	No Fraud
COVERED	131	Initially believed to be a fraud, but after research,the experts decided it was not.
Verified Fraud	102	Expert confirmed that a transaction was fraudulent.
Under Watch	16	Currently under investigation to be classified as a possible fraud.
Suspected Fraud	4	Most likely a fraud, but no definitive decision has been taken yet.

Table 4
Counts of fraud labels in MastercardAcqResolution.

MastercardAcqResolution	Count	Description
No Fraud	6,911,889	No Fraud.
Card Not Present Fraud	2000	No physical card being presented to a merchant (typically for online fraud).
Account Takeover Fraud	1910	Identity theft where fraudsters overtake an online account.
Card Reported Stolen	5	Stolen card.
Card Reported Lost	3	Lost card.

monitoring team and 'MastercardAcqResolution' (Table 4) from Mastercard. The company's label was manually assigned based on alerts from its rule-based fraud detection system, while the Mastercard label was automatically provided as an additional fraud prevention layer. These two labels were combined to form the 'Anomaly' feature, a unified fraud indicator. For each of these two features ('LocalizedLabel' and 'MastercardAcqResolution'), there are multiple resolution types each transaction can have. These resolution types and their count are summarized in Tables 3, 4.

The "LocalizedLabel" table corresponds to the instances the monitoring team manually looked over. Therefore, each of these labels has been prescribed to the data by an expert. In "LocalizedLabel", categories like "No Fraud", "COVERED", and "Under Watch" are marked as non-fraud, while "Verified Fraud" and "Suspected Fraud" are labeled as frauds. Similarly, in "MastercardAcqResolution", only "No Fraud" is non-fraudulent. Two binary columns are created: "Fraud" from "LocalizedLabel" and "MastercardFraud" from "MastercardAcqResolution", with non-fraud set to 0 and fraud to 1. The final "Anomaly" column is derived by setting it to 1 if either source indicates fraud, providing a unified fraud detection indicator. Table 5 gives a comprehensive description of these features.

Merchants and Persons datasets. After exploring the initial combined dataset, it was segmented into two distinct datasets: one for merchants and one for persons. The Merchant dataset contains transactions from 5873 merchants, while the Persons dataset includes 1029 individuals. Both datasets were mostly clean, with minimal missing values. However, the columns "cardacceptorcategorygroupname" and "POSCountryCode" contained some missing data. These rows were removed, as the missing information could not be recovered, and both features were critical for feature engineering.

Anomaly Counts. The number of fraudulent transactions differed between the two datasets, with more fraud instances in the Merchant dataset, which reflects its larger transaction volume. However, both datasets exhibited significant class imbalance, with fraud rates of 0.056 % for merchants and 0.072 % for persons. In Table 6 the summary of the number of fraudulent transactions in each dataset

4.3. Data preparation

The data preparation phase converts raw transaction data into a structured format suitable for machine learning. This process includes data cleaning, categorical encoding, feature selection, and feature engineering to enhance data quality and relevance.

The dataset comprises continuous, categorical, and binary features describing cardholder, transaction, and terminal entities. The cardholder is represented by a unique ID, country code, and currency code. Transactions include date-time, amount, transaction type, and security-sensitive attributes, while terminals are identified by merchant category code (MCC) and country. As machine learning models require numerical inputs, categorical features are transformed using encoding techniques (Kotsiantis, Zaharakis, Pintelas et al., 2007).

Beyond static attributes, fraud detection benefits from dynamic behavioral features that capture transaction patterns over time. To optimize model performance, we implemented a structured feature engineering pipeline leveraging domain knowledge and empirical testing. The key steps applied to both merchant and individual cardholder datasets are detailed below.

1. **Initial Preprocessing:**
   - Converted date columns into the standardized *datetime* format.
   - Calculated the average spending per cardholder and computed the deviation of each transaction amount from this average.
2. **Quartile Information:**
   - Derived statistical measures such as quartiles and interquartile ranges to identify typical transaction hours.
   - Introduced a binary feature indicating whether a transaction occurred within the user's usual activity window.
3. **Transaction Differences:**
   - Engineered temporal features, including time gaps between card issuance and transactions (an unusually short interval between card issuance and a transaction may indicate fraud).
   - Computed the time and amount differences between consecutive transactions (we introduced a binary feature to flag transactions with identical amounts to the previous one, a known fraud pattern).
   - Estimated the time difference between consecutive transactions for each cardholder to identify rapid, back-to-back transactions (pattern indicative of a possible card cloning fraud).
4. **Time Period Metrics:**
   - Aggregated transactions over hourly, daily, and weekly periods to calculate average transaction amounts.
   - Each transaction was then compared to these averages to identify significant deviations.
5. **Country code comparison:**
   - Created a binary feature to flag transactions where the country code of the cardholder differed from the card acceptor (foreign transactions might be potential anomalies).
6. **Usuality Categories:** Assigned a score (0 to 1) to transaction attributes (card acceptors, category codes (MCC), country codes, currencies) based on historical cardholder behavior.
   Score Interpretation:
   - **Closer to 0**: Indicates common transactions aligned with usual behavior.
   - **Closer to 1**: Flags rare or unusual transactions as potential anomalies.

Table 5
Overview of original dataset columns and types.

Feature Name	Type	Description
MerchantId	string	Identifier for the merchant.
MerchantCountryCode	string	Country code of the merchant.
MerchantRegistrationDate	object	Date when the merchant registered.
PersonId	string	Identifier for the person performing the transaction.
PersonCountryCode	string	Country code of the person.
PersonRegistrationDate	datetime	Date when the person registered.
TransactionId	string	Unique identifier for the transaction.
TransactionTypeId	int	Type of transaction.
Amount	float	Transaction amount.
MerchantCurrencyCode	int	Currency code of the merchant.
PointOfServiceEntryMode	string	Mode of entry at the point of service.
CurrencyCodeTransaction	int	Currency code for the transaction.
CardAcceptorBusinessCode	int	Business code of the card acceptor.
TransactionCreationDate	datetime	Date when the transaction was created.
CP	boolean	Card-present indicator.
IsSingleTapProcessingSupported	boolean	Indicator if single tap processing is supported.
PinVerified	boolean	Indicator if the transaction is PIN verified.
IssuedCardId	string	Identifier for the issued card.
CardHolderId	string	Identifier for the cardholder.
MastercardAcqResolution	string	Resolution status (label) offered by Mastercard.
LocalizedLabel	string	resolution type (label) of the transaction as defined by the Greek bank's monitoring team.
IssuedCardCreatedOn	datetime	Date when the card was created.
CurrencyExchangeRate	float	Exchange rate for the currency.
POSCountryCode	string	Country code of the point of service.
mcccode	int	Merchant category code.
merchantcategoryname	string	Merchant category name.
merchantcategorygroupname	string	Merchant category group name.
cardacceptorcategorygroupname	string	Card acceptor category group name.
Anomaly	int	Final label, combination of MastercardAcqResolution and LocalizedLabel.

Table 6
Anomaly counts for merchants and persons datasets.

Anomaly	Merchants Count	Persons Count
0	5,846,886	984,771
1	3264	714
Total	5,850,150	985,485

This scoring system helped capture deviations from a cardholder's established transactional patterns. For example, a transaction occurring in an unfamiliar country or in an unusual merchant category would receive a higher usuality score, signaling a potential anomaly.
Country Code Analysis:

- Domestic Transactions: "Usual_CountryCode" values near 0 confirmed expected spending patterns (see Fig. 2).
- Foreign Transactions: Higher "Usual_CountryCode" values signaled deviations from normal behavior (closer to 1, see Fig. 3).

Tracking these deviations helped detect transactions inconsistent with a cardholder's normal behavior.

7. **Daily Volume Statistics:**
   - Calculated the average number of daily transactions and spending amount per cardholder.
   - Compared each transaction amount against these averages to detect potential fraud cases.
8. **One-hot Encoding:**
   - Transformed categorical variables, such as "TransactionTypeId", into numerical format (one-hot encoding) to ensure compatibility with machine learning models.
9. **Repeated Values Detection:**
   - Developed a feature to identify consecutive repeated values in specific columns such as "PointOfServiceEntryMode" and "CardAcceptorBusinessCode", which can indicate automated or fraudulent behavior.

A summary of all the features created by the feature engineering pipeline is given in Table 7.

The Pearson correlation of all the final features of the Persons and Merchants datasets is shown in Figs. 4 and 5, respectively. In both datasets the features with highest correlation with the target variable were CountCarfdRatioOfAvg, RatioToAverageDailyVolume, Usual_Currency, Usual_CountryCode and Usual_MCC

5. Experiments setup

This section presents the experimental framework for evaluating various unsupervised anomaly detection techniques applied to transactional data. The goal is to assess both model performance and the influence of different feature selection methods on detection accuracy. When applied, feature selection methodologies were executed after the data preparation phase and before model training. All experiments were conducted on a DELL Precision 3560 laptop with 32 GB RAM and an Intel(R) i7 processor.

The first experiment evaluates Isolation Forest and One-class SVM for anomaly detection at the cardholder level, establishing baseline performance metrics for these models using transaction data. The second experiment tests the use of autoencoder to detect fraudulent transactions more generally, investigating whether deep learning can effectively capture complex patterns of fraud. In the third experiment, feature selection techniques, particularly genetic algorithms, are applied to improve model accuracy and computational efficiency. The impact of these techniques is rigorously analyzed. Finally, SHAP (SHapley Additive exPlanations) is employed to enhance model interpretability, with a focus on its potential integration in practical, industrial settings.

5.1. Experiment 1: Machine learning methods for anomaly detection per cardholder basis

We employed two unsupervised machine learning models, Isolation Forest and One-Class SVM, to perform anomaly detection at the cardholder (per each cardholder) level. A unique model for each cardholder was trained on historical data and evaluated on a subset of transactions. Although both models are unsupervised, we utilized labeled data for

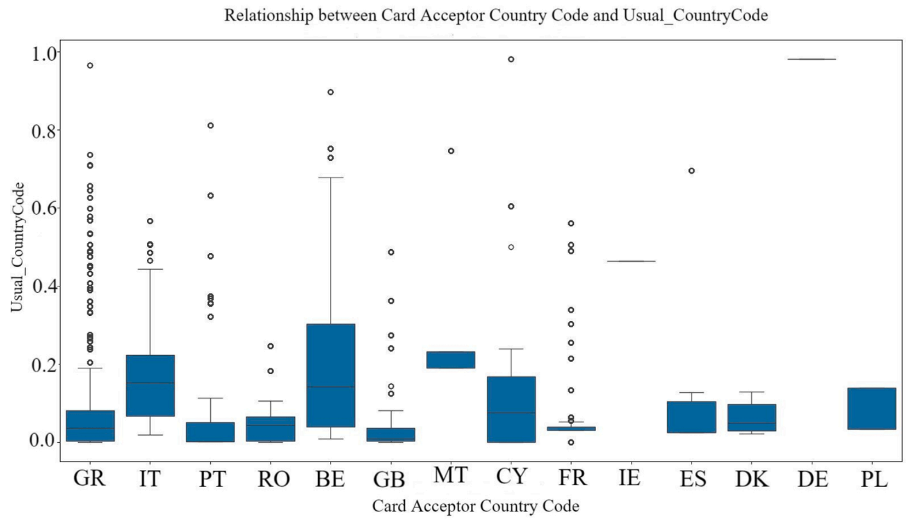

Fig. 2. Usual_CountryCode feature value, when merchants performed transaction **inside their registered country** (so where "cc≠CAcc' equals 0). We can see from the boxplots that most values are closer to 0, with a few outlier exceptions.

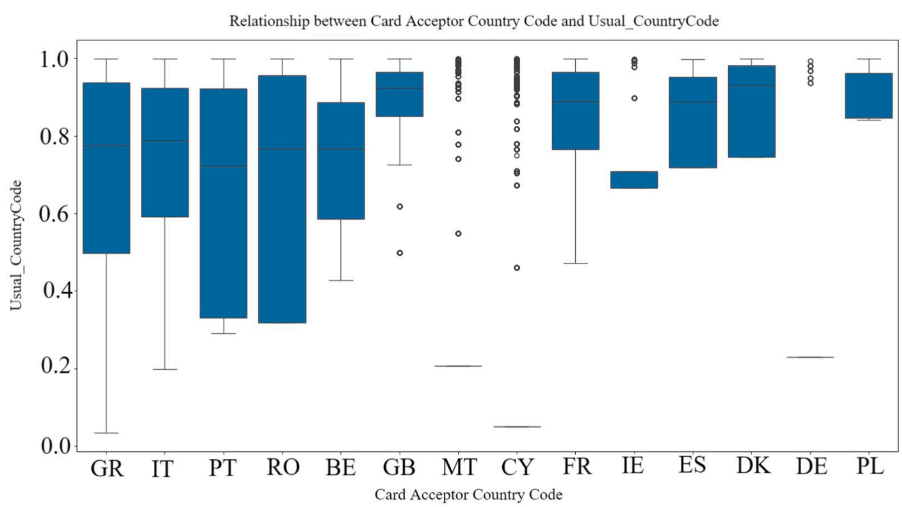

Fig. 3. Usual_CountryCode feature value, when merchants performed transaction **outside their registered country** (so where "cc≠CAcc' equals 1). As we can see, values of the feature are now significantly closer to 1, which indicates an unusual country, as opposed to the above graph.

Table 7
Overview of feature descriptions and types created by feature engineering pipeline.

Feature Name	Type	Description
AmountInEuro	float	Amount of transaction in euro.
AverageTransactionAmount	float	Average transaction amount per cardholder in euro.
DeviationFromAverage	float	Deviation (in euro) of each transaction from the average transaction amount.
PercentageOfAverage	float	Percentage of each transaction amount to the average.
transactionhour	int	Hour the transaction was performed at.
InsideQuartiles	boolean	Was the time of transaction between the usual transaction times of a cardholder.
DaysBetweenTransactionAndCardIssued	int	Difference in days between each transaction and the creation of the issuing card.
time_difference_with_previous_transaction	float	Time difference between consecutive transactions.
AmountDifference	float	Amount difference between consecutive transactions.
IsSameAmount	boolean	Is the amount of the transaction the same as the previous amount.
PercentageOfWeekly	float	The percentage of each transaction amount in regards to the average weekly amount.
cc ≠ Cacc	boolean	Difference between cardholder's and card acceptor's country codes.
Usuall_MCC	float	Range from 0 to 1. Values closer to 1 indicate unusuality in the card acceptor business code of a transaction.
Usual_CountryCode	float	Range from 0 to 1. Values closer to 1 indicate unusuality in the country of a transaction.
Usual_Currency	float	Range from 0 to 1. Values closer to 1 indicate unusuality in the currency of a transaction.
RatioToAverageDailyCount	float	The ratio of a day's volume compared to the average daily volume.
same_CardAcceptorBusinessCode	boolean	Did the previous transaction have the same card acceptor business code.
same_PointOfServiceEntryMode	boolean	Did the previous transaction have the same POS mode.
AmountCardRatioOfAverage	float	The ratio of each transaction amount to that of the average per card.

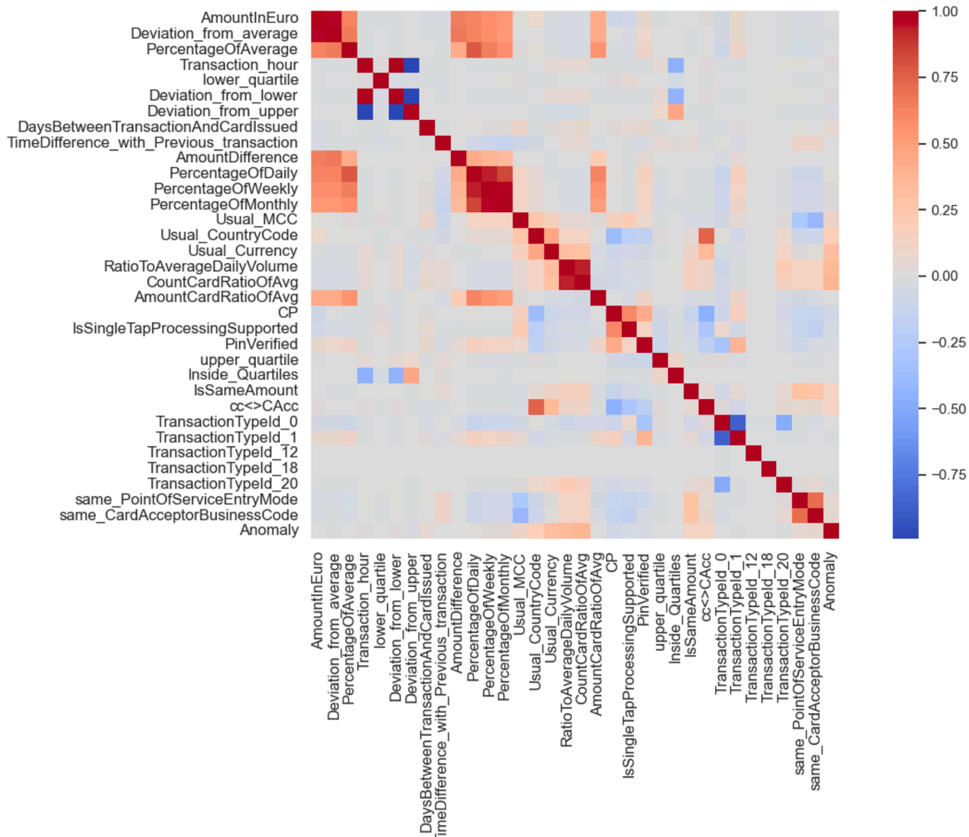

Fig. 4. The correlation heatmap for the Merchant dataset. In this image, the target variable, “Anomaly”, is the last feature.

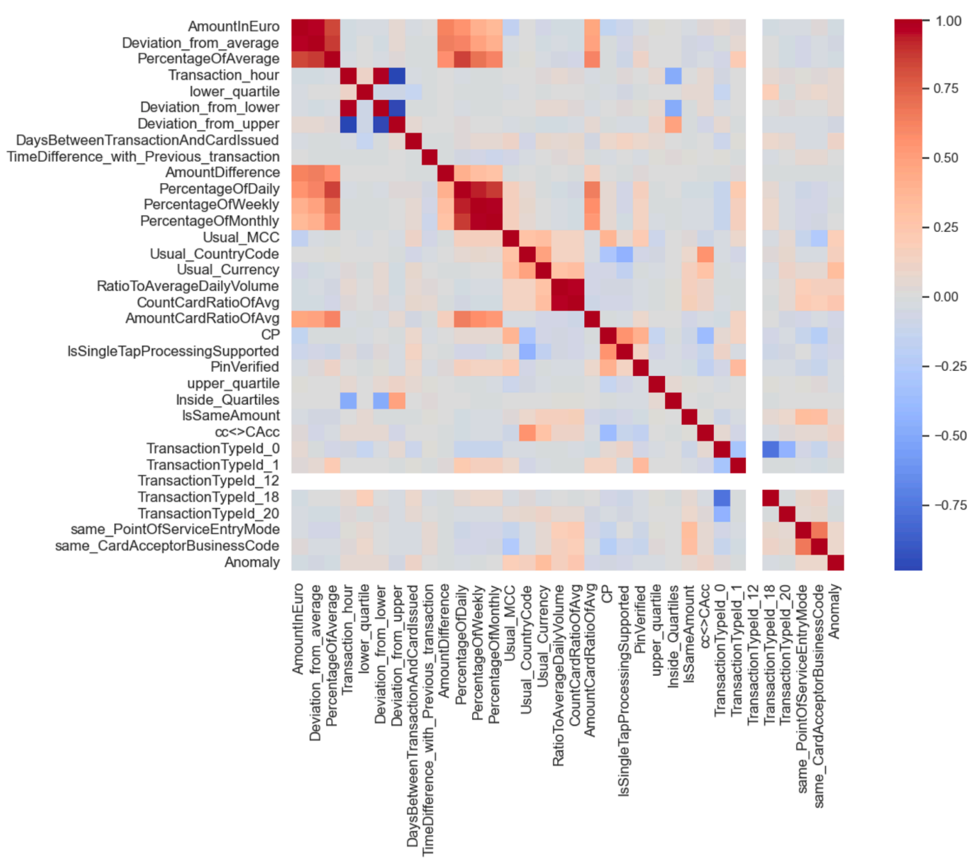

Fig. 5. The correlation heatmap for the Persons dataset. In this image, the target variable, “Anomaly”, is the last feature.

evaluation purposes. Parameter tuning was essential to optimize performance.

5.1.0.1. ***Parameter Tuning.*** For the Isolation Forest, the primary parameters were “n_estimators” (set to 100 following the recommendation of Liu et al. (2008)), which specifies the number of base estimators in the ensemble, typically set to a higher number for better performance, although this increases computational cost. The “max_samples” parameter determines the number of samples to draw from the dataset to train each base estimator. The “contamination” parameter, critical for balancing false positives and true positives, was set based on the proportion of anomalies in the datasets (0.056 for Merchants, 0.072 for Persons). For the One-Class SVM, the “nu” parameter, which bounds the fraction of outliers and support vectors, was similarly aligned with the anomaly proportions. The “gamma” parameter, affecting the decision boundary’s sensitivity, was set to its default value in the scikit-learn library.

5.1.0.2. ***Training Process.*** The training process for both models followed a similar workflow against the Persons and the Merchants datasets separately. We preprocessed the data, selected features using domain expertise, and scaled the inputs uniformly to avoid dominance by features with larger scales. Each model was trained separately on the Persons and Merchants datasets, followed by label predictions for the test sets. The results were then compared against actual labels, and evaluation metrics were computed. Stratified K-fold validation was applied to ensure balanced class representation and provide robust evaluation metrics. The resulting metrics represent averages across the folds, reducing bias and variance.

5.2. Experiment 2: Testing autoencoders for generalized anomaly detection

This experiment uses an autoencoder model to present a deep-learning approach for generalized fraud detection across transactional datasets. Instead of modeling data on a per-cardholder basis, a single model is applied across all cardholders. The chosen architecture features a symmetrical encoder-decoder structure designed to capture and reconstruct input data, facilitating the detection of anomalous transactions.

The encoder consists of five sequential layers, starting with units matching the input features, followed by layers with 16, 8, 4, and finally 2 units. Each layer uses Exponential Linear Unit (ELU) activation function. This compression into a lower-dimensional space, represented by the smallest layer, attempts to capture the most salient features. The decoder mirrors this structure, also employing ELU activations except in the output layer, which uses a sigmoid activation to reconstruct the input.

The autoencoder is trained to minimize the reconstruction error (measured by mean squared error), optimized via stochastic gradient descent (SGD). This process uses stochastic gradient descent (SGD) as the optimization algorithm. Training spans 60 epochs with a batch size of 256, and 20 % of the training data is reserved for validation. This configuration aims to avoid overfitting and optimize model weights. The model’s effectiveness is evaluated by its ability to classify transactions as normal or fraudulent, based on a reconstruction error threshold set at 0.95.

5.3. Experiment 3: Investigating how feature selection can improve our model’ performance

One of the greatest challenges in multivariate analysis is the selection of features that produce the best model performance (Leardi, Boggia, & Terrile, 1992). In our research questions, we wondered if state-of-the-art feature selection methods can further improve the performance of the employed models. Genetic Algorithms have been successfully used as a feature selection method in the past (Babatunde et al., 2014) (Leardi et al., 1992). Therefore, in this experiment, a Genetic Algorithm (GA) for feature selection is implemented to attempt to optimize the performance of our models. Additionally, as a second approach to feature selection, we leverage domain expertise gained through collaboration with experts from the Greek bank to manually curate a feature set. The results of the application of this feature selection methods, as well as the application of no feature selection method, are discussed in Section 6.

5.3.0.1. ***Feature selection process using genetic algorithms:.*** The experiment’s population consists of 60 candidate solutions (chromosomes), with fitness evaluated based on the F1 score from a Decision Tree classifier trained on the selected features. The population evolves over 8 generations through selection, crossover, and mutation. The top 40 chromosomes are selected as parents for the next generation. Crossover combines pairs of parent chromosomes using the One-point cross-over method to create new offsprings. Mutation introduces ’flips’ in random positions in the chromosomes, meaning that features are switched from included to excluded or vice versa. The mutation rate determines the number of positions that will be mutated. In our experiment, this rate is 0.2. This iterative process converges toward an optimal or near-optimal feature subset, maximizing classification performance. The best chromosome from all generations with the best F1 score defines the feature set for the final model, applied to both the Merchants and Persons datasets.

6. Results

Tables 8, 9 and 10 present the performance of the autoencoder, the Isolation Forest and the One-class SVM models on both datasets, leveraging different feature sets. Namely:

- **No FS**: No Feature Selection, all features are used without feature selection.
- **GA FS**: Genetic Algorithm Feature Selection, a near-optimal subset of features is selected via the use of a genetic algorithm.
- **DE FS**: Domain Expertise Feature Selection, features are selected based on domain expertise. This feature selection method was used in Experiments 1 and 2.

6.0.0.1. ***Autoencoder results.*** In the Merchants dataset from Table 8, domain-expertise feature selection (**DE FS**) yields a slightly better True Positive (TP) Rate and Precision compared to the other two conditions, indicating that domain knowledge improves performance marginally. Both **No FS** and **GA FS** show identical results, implying that the genetic algorithm did not significantly enhance the model.

In the Persons dataset, **DE FS** condition achieves the highest TP Rate (85 %), Recall (85 %), and Precision 11 % indicating substantial improvements with domain knowledge. **GA FS**, however, underperforms, with the lowest TP Rate (61 %) and AUC (75 %) implying that the genetic algorithm might not be well-suited for this dataset, possibly due

Table 8
Performance Metrics for **Autoencoder** with Different Feature Selection Methods, on both the Merchant and Persons Datasets. FS denotes No Feature Selection, GA FS denotes feature selection with the genetic algorithm and finally, DE FS denotes feature selection using domain expertise.

Metric	Merchant Dataset			Persons Dataset		
	No FS	GA FS	DE FS	No FS	GA FS	DE FS
TP Rate	77 %	77 %	78 %	83 %	61 %	85 %
FP Rate	9 %	9 %	9 %	9 %	9 %	9 %
Accuracy	90 %	90 %	90 %	90 %	90 %	90 %
F1 Score	10 %	10 %	19 %	20 %	15 %	25 %
AUC	83 %	83 %	85 %	88 %	75 %	87 %
Precision	5 %	5 %	6 %	11 %	8 %	11 %
Recall	77 %	77 %	79 %	83 %	61 %	85 %

Table 9
Performance Metrics for **Isolation Forest** with Different Feature Selection Methods, on both the Merchant and Persons Datasets. FS denotes No Feature Selection, GA FS denotes feature selection with the Genetic algorithm, and DE FS denotes feature selection using domain expertise.

Metric	Merchant Dataset			Persons Dataset		
	No FS	GA FS	DE FS	No FS	GA FS	DE FS
TP Rate	43 %	41 %	59 %	24 %	30 %	48 %
FP Rate	5 %	5 %	5 %	6 %	5 %	6 %
Accuracy	92 %	92 %	93 %	90 %	90 %	90 %
F1 Score	9 %	9 %	15 %	7 %	11 %	20 %
AUC	68 %	68 %	80 %	55 %	60 %	70 %
Precision	7 %	7 %	12 %	6 %	10 %	17 %
Recall	43 %	41 %	59 %	24 %	30 %	48 %

Table 10
Performance Metrics for **One class SVM** with Different Feature Selection Methods, on both the Merchant and Persons Datasets. FS denotes No Feature Selection, GA FS denotes feature selection with the Genetic algorithm and finally, DE FS denotes feature selection using domain expertise.

Metric	Merchant Dataset			Persons Dataset		
	No FS	GA FS	DE FS	No FS	GA FS	DE FS
TP Rate	22 %	29 %	44 %	20 %	16 %	42 %
FP Rate	6 %	7 %	8 %	8 %	6 %	13 %
Accuracy	91 %	91 %	90 %	88 %	89 %	84 %
F1 Score	5.5 %	6 %	8 %	6 %	7.4 %	10 %
AUC	60 %	60 %	68 %	55 %	54 %	63 %
Precision	4 %	5 %	6 %	6 %	7 %	8 %
Recall	22 %	29 %	44 %	20 %	16 %	42 %

to suboptimal feature selection. **No FS** maintains balanced performance, with a high F1 Score (20 %) and AUC (88 %) suggesting that even without feature selection, the model performs well.

Overall, domain expertise (**DE FS**) proves most effective, particularly in the Persons dataset, improving True Positive Rate and Recall while maintaining high Accuracy. The lack of difference between No FS and GA FS in the Merchant dataset indicates that feature selection may not be necessary in all contexts and should be evaluated on a case-by-case basis.

6.0.0.2. ***Isolation Forest results.*** For the Merchants dataset in Table 9, **DE FS** improves TP Rate (59 %) and Recall (59 %), while maintaining the same low FP Rate (5 %), indicating that domain knowledge enhances the model's performance. **GA FS** and **No FS** perform similarly across most metrics, suggesting that the genetic algorithm had little effect. **DE FS** also improves the F1 Score (15 %) and AUC (80 %), highlighting better overall model performance.

In the Persons dataset, **DE FS** again demonstrates superior results with the highest TP Rate (48 %), Recall (48 %), and F1 Score (20 %), indicating that domain expertise significantly improves model performance in this dataset as well. Interestingly, **GA FS** performs better than **No FS**, improving TP Rate (30 % vs. 24 %), F1 Score (11 % vs. 7 %), and AUC (60 % vs. 55 %), indicating some benefit from using a genetic algorithm for feature selection. The No FS condition yields the lowest performance across most metrics, suggesting that feature selection is generally a beneficial practice.

Isolation Forest's superior performance can be attributed to its efficiency in handling high-dimensional data (Liu et al., 2008). It isolates anomalies without constructing a profile of "normal" behavior, making it effective even when anomalies are infrequent but significantly different from normal observations. The isolation forest achieves improved TP performance and an even lower false positive rate than in the case of the Persons datasets. This proves the model's resistance and adaptability to high-dimensionality, and its ability to handle a large volume of information without loss in performance. However, in this case, we also note a decrease in Precision and F1 score. This can be attributed to the fact that while the Isolation Forest (IF) effectively identifies a larger percentage of fraud instances in this dataset with a notably low rate of false positives, the overall increase in the number of fraud instances causes a relative decline in the performance metrics for Precision and F1 score.

6.0.0.3. ***One-class SVM results.*** The Merchants dataset (see Table 10) shows improvement with Domain Expertise Feature Selection (**DE FS**), achieving the highest TP Rate (44 %), Recall (44 %) and F1 Score (8 %) maintaining low FP Rates. While **GA FS** offers a modest increase over **No FS**. This suggests that the features selected by the genetic algorithm might not capture the most relevant information for this dataset, leading to poorer performance.

Table 11
Ensemble Model Evaluation Results for both datasets.

Metric	Persons Dataset	Merchants Dataset
Mean TP Rate	59 %	70 %
Mean FP Rate	15 %	12 %
Mean Precision	10 %	7 %
Mean Recall	59 %	70 %
Mean Accuracy	83 %	87 %
Mean F1	14 %	11 %
Mean AUC	73 %	81 %

In the Persons dataset, **DE FS** consistently yields the best performance across all metrics, while **GA FS** slightly worsens the results compared to **No FS**.

Domain expertise in feature selection (**DE FS**) consistently enhances model performance, particularly in the Persons dataset. While beneficial in some cases, the genetic algorithm does not always outperform the no-feature selection approach. Thus, incorporating domain knowledge proves to be the most effective method for optimizing performance across different datasets. Compared to Isolation Forest, One-Class SVM's reliance on defining a decision boundary around normal data points makes it less effective when normal data is not well-clustered, resulting in higher false positives, especially in the Persons dataset.

6.0.0.4. ***Ensemble.*** An ensemble model combined the outputs of both the Isolation Forest and One-Class SVM models with equal weighting. This approach increased sensitivity by classifying an instance as an anomaly if either model flagged it. However, the trade-off was a higher rate of false positives. The ensemble method performed better on the Merchants dataset, achieving a higher mean true positive rate (70 %) and AUC score (81 %). However, precision remained relatively low in both datasets (see Table 11). This suggests that the ensemble approach prioritizes recall over precision, a strategic trade-off for detecting anomalies.

7. Experiment 4: Explanable AI using SHAP values

Improved interpretability improves the transparency of the model, facilitating broader adoption. Despite the high accuracy of advanced machine learning models, their "black-box" nature limits acceptance. Applying SHAP (SHapley Additive exPlanations) helps us understand these models' internal workings, improving business engagement in evaluating their predictions. Before opting for using SHAP, we also tested Principal Component Analysis (PCA) and LASSO. However, the tests yielded worse performance compared to SHAP. This section presents SHAP plots to interpret fraud detection results with the autoencoder.

Fig. 6 shows a global interpretability graph, where SHAP values indicate each feature's contribution to the target variable. The features

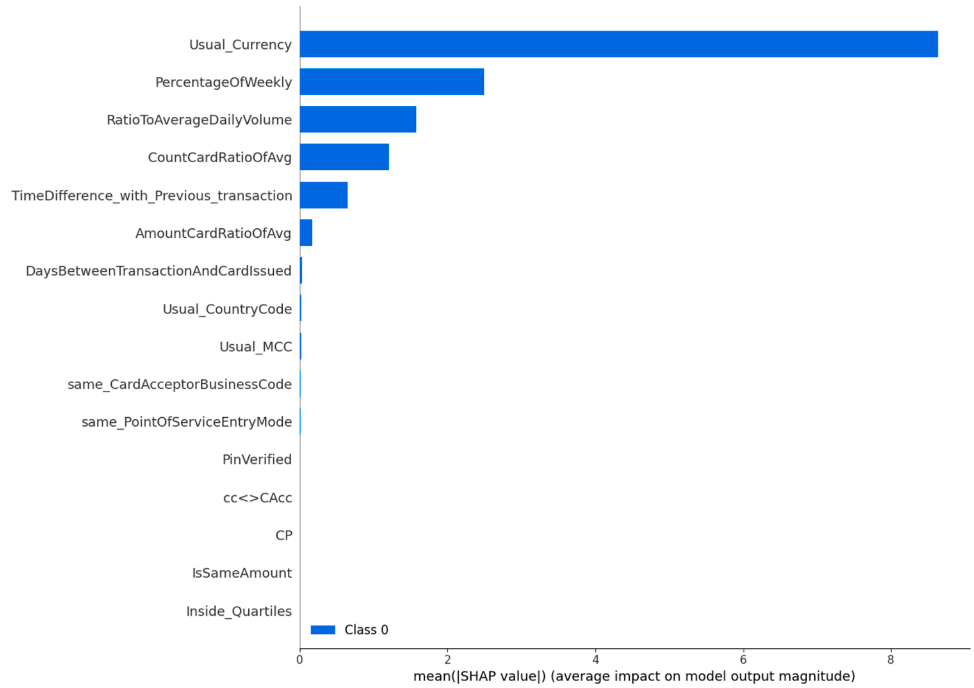

Fig. 6. The SHAP Variable Importance summary plot for the Merchants dataset.

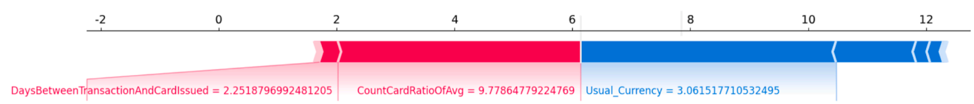

Fig. 7. The SHAP forceplot for a transaction in the test set.

are ranked in descending order of importance, with top-ranked features, such as "Usual_Currency", having the greatest influence on model predictions. Features like (and after) "same_CardAcceptorBusinessCode" have minimal impact.

SHAP also provides local interpretability, assigning each observation specific SHAP values, as shown in Figs. 7 and 8. This local interpretation improves transparency by revealing why the model classifies individual transactions as anomalies and what factors contribute to its score, which is crucial for understanding specific decisions. Local interpretability enables us to pinpoint and compare the effects of various factors on specific observations.

Fig. 7 displays a force plot for an anomaly detected by the autoencoder. Features in red (e.g., DaysBetweenTransactionAndCardIssued, CountCardRatioOfAvg) contribute positively to the model's output, pushing the prediction towards fraud, while those in blue (e.g., Usual_Currency) decrease the likelihood of fraud. Key features influencing this particular instance include:

- Feature "DaysBetweenTransactionAndCardIssued": With a value of 2.2 (higher than the average 0.03) positively impacts the data point being labeled an anomaly. This feature pushes the anomaly score to the left.
- Feature "CountCardRatioOfAvg": This feature also contributes positively on the data point being labeled as anomalous, with a SHAP value of 9.7.
- Feature "Usual_Currency": Negatively correlated, pushing the prediction away from fraud.

Fig. 8 offers a decision plot that visualizes how each feature contributes to the final model output for a specific prediction instance in the Merchants dataset. The x-axis represents the model's mean squared error (MSE)and is centered on the explainer's expected value (the mean of all MSE scores). The y-axis lists the model's features by their descending importance. The importance is calculated based on the observations plotted. Features like "CountCardRatioOfAvg" (with a positive SHAP value of 9.779) and "DaysBetweenTransactionAndCardIssued" (SHAP value of 2.252) significantly push the model toward predicting fraud, while "Usual_Currency" (SHAP value of 3.062) pushes the model away from an anomaly prediction. Other features, such as "PercentageOfWeekly" and "AmountCardRatioOfAvg", contribute minor positive or negative influences.

Several other features, such as "PercentageOfWeekly", "TransactionAndCardIssued", "with_Previous_transaction", and "AmountCardRatioOfAvg", contribute minimally to the model's predictions.

Both the waterfall plot (8) and the force plot (7) demonstrate that the features generated by the feature engineering pipeline are key predictors influencing the model's decisions. This further confirms the significant value these features add to the model and its predictive accuracy.

8. Discussion

This work addresses four key research questions concerning unsupervised anomaly detection in credit card transaction data. The primary objective is to assess model effectiveness in detecting fraudulent transactions and explore how interpretable machine learning can enhance their application in industrial settings.

RQ1: How accurately can Isolation Forest and One-class SVM identify fraudulent transactions on a per-cardholder basis?

The first question examines the performance of Isolation Forest and One-Class SVM to improve the precision of fraud detection. Results from Tables 9 to 10 shows that Isolation Forest outperforms One-Class SVM across multiple metrics, including true positives (TP), F1 score, accuracy, and AUC score. This aligns with existing studies, where Isolation

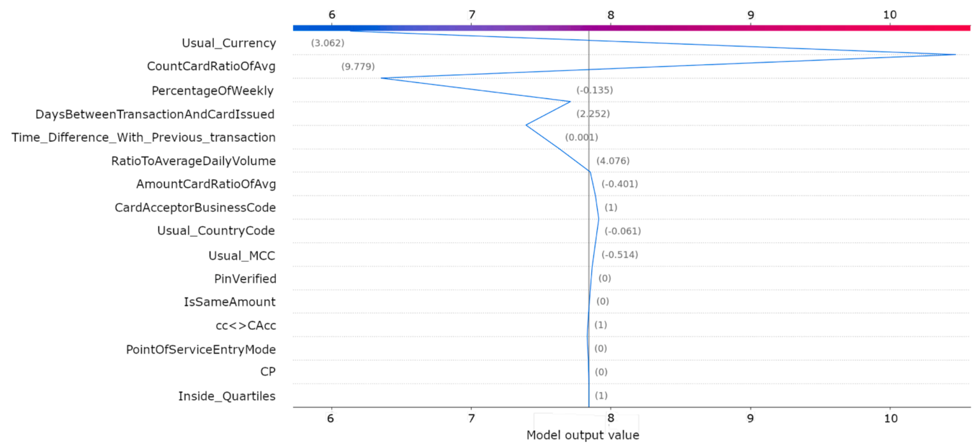

Fig. 8. The SHAP Variable Importance summary plot for the Merchants dataset. This was created for the same point as Fig. 7 was generated.

Forest consistently surpasses methods like Local Outlier Factor and One-Class SVM (Bhattacharyya et al., 2011; Dhulipudi et al., 2024; Maniraj et al., 2019; Ounacer, El Bour, Oubrahim, Ghoumari, & Azzouazi, 2018; Vijayakumar et al., 2020).

Moreover, we investigated an ensemble model approach combining Isolation Forest and One-Class SVM, which improved the true positive rate but increased false positives, indicating a trade-off between recall and precision. This indicates that while ensemble methods can enhance recall, they may also introduce more noise in the form of false positives. Moreover, ensemble models require more computational resources to train (Khalid et al., 2024). We can then state that Isolation Forest may be preferable when minimizing false positives is critical, such as in fraud detection, where excessive false positives result in unnecessary resource expenditure (Khalid et al., 2024).

RQ2: To what extent do deep learning methods, specifically autoencoders, outperform traditional machine learning methods on generalized anomaly detection?

The second question investigates whether deep learning methods outperform traditional machine learning approaches. The results in Table 8 evaluate the effectiveness of a generalized autoencoder model, showing robust performance in detecting fraud across the Persons and Merchants data sets. The autoencoder achieved the highest TP rates (85 % for Persons, 78 % for Merchants) and F1 scores (25 % for Persons, 19 % for Merchants), along with the highest AUC scores (87 % for Persons, 85 % for Merchants). Despite its strong performance, the model had slightly lower precision compared to Isolation Forest, specifically 11 % (Persons) and 6 % (Merchants), compared to the 17 % and 12 % of Isolation Forest, pointing to future work in improving precision without sacrificing recall. Additionally, the autoencoder introduces some false positives. In fact, the autoencoder has an FP rate of 9 % (both for Persons and Merchant), slightly higher than that of the Isolation Forest (5 % for Merchants and 6 % for Persons).

RQ3: To what extent can genetic algorithms be used for feature selection to improve the performance of the chosen models?

Experiment 3 in Section 5.3 and the results from Section 6 explores how feature selection affects model performance using Genetic Algorithms (GA) and domain expertise. The results show that domain expertise significantly improves metrics such as true positives and recall, especially in the Persons dataset. GA yielded mixed results, offering improvements over no feature selection but not surpassing domain expertise. This suggests that while GA is valuable for automating feature selection, integrating domain knowledge with algorithmic methods may produce optimal feature subsets.

RQ4: How can explainable AI be incorporated to interpret the proposed models' predictions?

The final research question examines the role of SHAP in increasing model explainability. Experiment Section 7 demonstrates how SHAP explanations enhance model interpretability by providing global and local insights into feature importance. SHAP's local interpretability enables business experts to understand individual predictions better, increasing trust and transparency. This aligns with previous research, showing SHAP's effectiveness in improving task effectiveness, efficiency, and mental workload in fraud detection (Antwarg, Miller, Shapira, & Rokach, 2019; Weerts, van Ipenburg, & Pechenizkiy, 2019).

Task Effectiveness: SHAP increases users' ability to assess model credibility by highlighting important features, improving task effectiveness in fraud detection. The business experts from the Greek bank acknowledge that the SHAP explanations increase task effectiveness of application processing compared to the model's predictions alone, since they provide transparency and thus, increase trust.

SHAP allows faster decision-making by revealing relevant feature values, aiding experts in processing applications more efficiently by focusing on significant features that correlate with fraud. This is crucial in unsupervised learning scenarios to verify assumptions about outlying behavior and help the domain experts process applications faster.

Mental Efficiency: SHAP helps reduce cognitive load by guiding experts to focus on key features and improving interpretability, particularly in complex cases. SHAP explanations enhanced the assessment of anomalous applications by business experts, improving their understanding of models and speeding up anomaly evaluation. Even though we could not conduct a statistical analysis to confirm differences in task utility metrics between tasks with and without SHAP explanations for confidentiality reasons, we conducted a small number of unstructured interviews with the alert monitoring team that showed the benefits of this explainability tool.

9. Conclusions

Financial institutions (FIs) experience significant losses due to fraud each year, and many have turned to AI and machine learning to detect suspicious activity. As fraud schemes evolve, AI-driven unsupervised techniques offer a valuable solution, capable of analyzing large volumes of data in real time without requiring labeled datasets, making them ideal in situations where labeled data is scarce.

This paper contributes to enhancing the Greek bank's rule-based fraud detection systems by incorporating unsupervised models like Isolation Forest and One-Class SVM for per-cardholder anomaly detection, as well as autoencoders for generalized fraud detection. It also explores the role of feature selection, particularly genetic algorithms, in optimizing model performance, and introduces SHAP values to improve model explainability.

Contributions to the scientific community. This work makes several contributions to the field of fraud detection. It introduces a novel approach that incorporates individual cardholder behavior for a more personalized risk assessment, which, to the best of the authors' knowledge, has not been extensively explored in the literature. This perspective improves the accuracy of fraud detection models.

Additionally, this paper addresses the issue of model interpretability, which is often lacking in complex fraud detection systems. By employing SHAP, the work contributes to the growing discourse on transparent machine learning in finance.

Furthermore, the paper introduces innovative feature engineering techniques inspired by both domain expertise and intuition. These new features improve the ability of the model to detect subtle patterns, further advancing fraud detection capabilities.

Overall, this research contributes to building more effective and interpretable fraud detection systems, offering concepts and methodologies that can benefit both researchers and practitioners in the financial sector. It also strengthens the theoretical foundations of AI and machine learning applications in finance.

Contributions to the Greek bank institution. The primary contribution to the Greek bank involved in this research is the development of a machine learning system for credit card fraud detection, outperforming the company's existing system. The unsupervised models adapt to emerging fraud patterns without requiring retraining on newly labeled data, a crucial capability in fast-evolving transaction environments.

A major advantage of the system is its interpretability. By incorporating SHAP, the model provides clear insights into its decisions, building trust with fraud analysts and enhancing transparency in fraud detection. Moreover, after the experience with the Greek financial institution, we can emphasize that to improve computational scalability, we proposed the company to leverage cloud-based technologies and infrastructure-as-a-service (IaaS) providers that allow cost-efficient resource allocation. A scalable pipeline can be designed where model training and inference are executed within dynamically provisioned job clusters. These clusters allocate only the computational resources necessary for task execution and automatically terminate after completion, minimizing resource waste and optimizing cost efficiency.

10. Threats to validity

This section discusses potential threats to the validity of this work, which may impact the interpretation of the findings and the generalizability of the results.

Parameter setting and model performance. A key limitation in this study lies in the parameter settings of the machine learning and deep learning models. Parameters like the contamination factor in Isolation Forest (set at 0.056) and the 'nu' parameter in One-Class SVM (set at 0.072) were selected based on preliminary experiments to reflect the percentage of anomalies in the datasets. However, optimal parameter settings can vary significantly across different datasets. This variation poses a risk to the models' performance and robustness when applied to other datasets, potentially limiting their generalizability.

Data representativeness. The dataset used in this research, comprising 6,915,699 transactions from the Greek financial institution, represents a subset of the company's full transactional data. While sizable, it may not capture all transaction types or anomalies present in the company's operations. This subset was chosen to manage computational resources and streamline analysis, but it may limit the findings' generalizability to other datasets or scenarios, as it might not fully encompass all patterns or anomalies.

Accuracy of labels. The accuracy of the labels, classifying transactions as normal or fraudulent, poses a validity threat. It is likely that some fraudulent transactions were misclassified by the company's detection systems, leading to imperfect label accuracy. This could bias model training and affect their ability to generalize to real-world scenarios. As a result, the reliability of the evaluation metrics used in this study may be impacted by the quality of the available labels.

Reproducibility challenges. The reproducibility of this research is limited due to the proprietary nature of the dataset from the Greek financial institution, which contains sensitive transactional data, restricting access for external researchers. This hinders the ability to replicate and validate the findings. Additionally, the preprocessing steps, feature engineering, and model training procedures were tailored to this dataset, which may not be directly applicable to other datasets or fraud detection contexts, limiting the broader applicability of the research outcomes.

These validity threats must be considered when interpreting the results. Future studies should address these limitations by using more generalized models, accessing broader datasets, or improving label accuracy with enhanced detection systems. Acknowledging these issues is essential for refining future research in this area.

11. Future work

While the current models deliver promising results, further enhancements are possible. One key limitation is the manual setting of model parameters. Future studies could investigate dynamic parameterization methods, such as the autoencoder architecture proposed by Ko, Na, Oh, Kim, and Youn (2022), which uses denoising and ensemble techniques with a dynamic threshold to reduce false alarms in anomaly detection.

Expanding the application scope beyond credit card fraud to include loan and insurance fraud could also broaden the models' utility. Adapting models to other fraud types, such as fraud in mobile communication networks (Pawling, Chawla, & Madey, 2007) or smart building energy consumption (Ribeiro, Lazzaretti, & Lopes, 2018), would further test their generalizability.

Future work could explore advanced feature engineering techniques, incorporating deep learning for automatic feature extraction and selection. This approach might reveal subtle, complex patterns that traditional methods overlook.

Although SHAP values have improved model interpretability, further research should focus on creating more intuitive explanation interfaces to enhance understanding for non-expert users.

Data availability

The authors do not have permission to share data.

CRediT authorship contribution statement

Evgenia Karnavou: Methodology, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft; **Giuseppe Cascavilla:** Methodology, Conceptualization, Writing – original draft, Writing – review & editing, Supervision; **Gonçalo Marcelino:** Conceptualization, Writing – original draft, Writing – review & editing, Supervision; **Zeno Geradts:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Miss Evgenia Karnavou worked as a master's thesis student at the Greek bank institution from this study for nine months. The bank provided the data for the experiments. Currently, Miss E. Karnavou is no longer affiliated with the bank and has no ongoing relationship with the institution. None of the other members have had or do not have any relationship with the Greek Bank. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

ACI Worldwide (2024). Scamscope report: APP scam trends. Retrieved May 21, 2024. https://www.aciworldwide.com/scamscope-report-app-scam-trends.

Antwarg, L., Miller, R. M., Shapira, B., & Rokach, L. (2019). Explaining anomalies detected by autoencoders using SHAP. arXiv preprint arXiv:1903.02407

Arjun, K. P., Atlas, G., Sreenarayanan, N. M., Janarthanan, S., & Arvindhan, M. (2022). Early prediction of credit card transaction using local outlier factor and isolation forest tree machine learning algorithms. In B. B. V. L. Deepak, D. R. K. Parhi, B. Biswal, & P. C. Jena (Eds.), *Applications of computational methods in manufacturing and product design* (pp. 463–472). Singapore: Springer Nature Singapore.

Babatunde, O. H., Armstrong, L., Leng, J., & Diepeveen, D. (2014). A genetic algorithm-based feature selection.

Baesens, B., Van Vlasselaer, V., & Verbeke, W. (2015). Fraud analytics using descriptive, predictive, and social network techniques: A guide to data science for fraud detection. John Wiley & Sons.

Bakumenko, A., & Elragal, A. (2022). Detecting anomalies in financial data using machine learning algorithms. *Systems*, *10*(5). https://doi.org/10.3390/systems10050130

Banerjee, R., Bourla, G., Chen, S., Kashyap, M., & Purohit, S. (2018). Comparative analysis of machine learning algorithms through credit card fraud detection. In *2018 IEEE MIT Undergraduate research technology conference (URTC)* (pp. 1–4). IEEE.

Becirovic, S., Zunic, E., & Donko, D. (2020). A case study of cluster-based and histogram-based multivariate anomaly detection approach in general ledgers. In *2020 19th International symposium Infoteh-Jahorina (INFOTEH)* (pp. 1–6). IEEE.

Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems*, *50*(3), 602–613.

Braei, M., & Wagner, S. (2020). Anomaly detection in univariate time-series: A survey on the state-of-the-art. arXiv preprint arXiv:2004.00433

Brill, F. Z., Brown, D. E., & Martin, W. N. (1992). Fast generic selection of features for neural network classifiers. *IEEE Transactions on Neural Networks*, *3*(2), 324–328.

Cascavilla, G. (2024). The rise of cybercrime and cyber-threat intelligence: Perspectives and challenges from law enforcement. *IEEE Security & Privacy*, (01), 2–11. https://doi.org/10.1109/MSEC.2024.3407859

Chalapathy, R., & Chawla, S. (2019). Deep learning for anomaly detection: A survey. arXiv preprint arXiv:1901.03407

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, *41*(3), 1–58.

Chang, V., Doan, L. M. T., Di Stefano, A., Sun, Z., & Fortino, G. (2022). Digital payment fraud detection methods in digital ages and industry 4.0. *Computers and Electrical Engineering*, *100*, 107734. https://doi.org/10.1016/j.compeleceng.2022.107734

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R. et al. (2000). CRISP-DM 1.0: Step-by-step data mining guide. *SPSS Inc*, *9*(13), 1–73.

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*, 321–357.

Dhulipudi, S., Siram, S., Pasha, M. J., Yadlapalli, Y., Kadiravan, D. G., & Subramanyam, D. M. M. (2024). Optimizing fraud detection in financial transactions: a comprehensive exploration of the effectiveness of random forest and isolation forest algorithms in detecting anomalies within credit card transactions. *Educational Administration: Theory and Practice*, *30*(4), 9146–9157. https://doi.org/10.53555/kuey.v30i4.3205

Ding, Y., Kang, W., Feng, J., Peng, B., & Yang, A. (2023). Credit card fraud detection based on improved variational autoencoder generative adversarial network. *IEEE Access*, *11*, 83680–83691. https://doi.org/10.1109/ACCESS.2023.3302339

Fiore, U., De Santis, A., Perla, F., Zanetti, P., & Palmieri, F. (2019). Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. *Information Sciences*, *479*, 448–455.

Fox, A. J. (1972). Outliers in time series. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *34*(3), 350–363.

Hejazi, M., & Singh, Y. P. (2013). One-class support vector machines approach to anomaly detection. *Applied Artificial Intelligence*, *27*(5), 351–366.

Hilal, W., Gadsden, S. A., & Yawney, J. (2022). Financial fraud: A review of anomaly detection techniques and recent advances. *Expert Systems with Applications*, *193*, 116429.

Holland, J. H. (1992). Genetic algorithms. *Scientific American*, *267*(1), 66–73.

Huang, Z., Zheng, H., Li, C., & Che, C. (2024). Application of machine learning-based k-means clustering for financial fraud detection. *Academic Journal of Science and Technology*, *10*(1), 33–39. https://doi.org/10.54097/74414c90

John, H., & Naaz, S. (2019). Credit card fraud detection using local outlier factor and isolation forest. *International Journal of Computational Science and Engineering*, *7*(4), 1060–1064.

Khalid, A. R., Owoh, N., Uthmani, O., Ashawa, M., Osamor, J., & Adejoh, J. (2024). Enhancing credit card fraud detection: An ensemble machine learning approach. *Big Data and Cognitive Computing*, *8*(1), 6.

Ko, J. U., Na, K., Oh, J.-S., Kim, J., & Youn, B. D. (2022). A new auto-encoder-based dynamic threshold to reduce false alarm rate for anomaly detection of steam turbines. *Expert Systems with Applications*, *189*, 116094.

Kotsiantis, S. B., Zaharakis, I., Pintelas, P. et al. (2007). Supervised machine learning: A review of classification techniques. *Emerging Artificial Intelligence Applications in Computer Engineering*, *160*(1), 3–24.

Leardi, R., Boggia, R., & Terrile, M. (1992). Genetic algorithms as a strategy for feature selection. *Journal of Chemometrics*, *6*(5), 267–281.

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In *2008 Eighth IEEE international conference on data mining* (pp. 413–422). IEEE.

Lucas, Y. (2019). Credit card fraud detection using machine learning with integration of contextual knowledge. Ph.D. thesis. Université de Lyon; Universität Passau (Allemagne).

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, *30*.

Maniraj, S. P., Saini, A., Ahmed, S., & Sarkar, S. (2019). Credit card fraud detection using machine learning and data science. *International Journal of Engineering Research*, *8*(9), 110–115.

Manish, G., Jing, G., Aggarwal Charu, C., & Jiawei, H. (2014). Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, *25*(1).

Molnar, C. (2020). Interpretable machine learning. Lulu. com.

Negi, K., Kumar, G. P., Raj, G., Sahana, S., & Jain, V. (2022). Degree of accuracy in credit card fraud detection using local outlier factor and isolation forest algorithm. In *2022 12th International conference on cloud computing, data science & engineering (confluence)* (pp. 240–245). https://doi.org/10.1109/Confluence52989.2022.9734123

Nonnenmacher, J., & Gómez, J. M. (2021). Unsupervised anomaly detection for internal auditing: Literature review and research agenda. *International Journal of Digital Accounting Research*, *21*.

Ounacer, S., El Bour, H. A., Oubrahim, Y., Ghoumari, M. Y., & Azzouazi, M. (2018). Using isolation forest in anomaly detection: The case of credit card transactions. *Periodicals of Engineering and Natural Sciences*, *6*(2), 394–400.

Pang, G., Shen, C., Cao, L., & Hengel, A. V. D. (2021). Deep learning for anomaly detection: A review. *ACM Computing Surveys (CSUR)*, *54*(2), 1–38.

Patel, T., & Kale, M. O. (2014). A secured approach to credit card fraud detection using hidden markov model. *International Journal of Advanced Research in Computer Engineering & Technology*, *3*(5), 1576–1583.

Pawling, A., Chawla, N. V., & Madey, G. (2007). Anomaly detection in a mobile communication network. *Computational and Mathematical Organization Theory*, *13*, 407–422.

Prasad, P. Y., Chowdarv, A. S., Bavitha, C., Mounisha, E., & Reethika, C. (2023). A comparison study of fraud detection in usage of credit cards using machine learning. In *2023 7th International conference on trends in electronics and informatics (ICOEI)* (pp. 1204–1209). IEEE.

Pymnts (2021). AI in focus: Waging digital warfare against payments fraud. Retrieved May 21, 2024 https://www.pymnts.com/study/ai-in-focus-digital-payments-fraud-prevention-banking/.

Pymnts (2024). The AI gap. Retrieved May 21, 2024. https://www.pymnts.com/tag/the-ai-gap/.

Ribeiro, M., Lazzaretti, A. E., & Lopes, H. S. (2018). A study of deep convolutional auto-encoders for anomaly detection in videos. *Pattern Recognition Letters*, *105*, 13–22.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135–1144).

Roy, A., Sun, J., Mahoney, R., Alonzi, L., Adams, S., & Beling, P. (2018). Deep learning detecting fraud in credit card transactions. In *2018 Systems and information engineering design symposium (SIEDS)* (pp. 129–134). IEEE.

Rushin, G., Stancil, C., Sun, M., Adams, S., & Beling, P. (2017). Horse race analysis in credit card fraud–deep learning, logistic regression, and gradient boosted tree. In *2017 Systems and information engineering design symposium (SIEDS)* (pp. 117–121). IEEE.

Sahin, Y., & Duman, E. (2011). Detecting credit card fraud by decision trees and support vector machines. In *Proceedings of the international multiconference of engineers and computer scientists* (pp. 1–6). (vol. *1*).

Saputra, A. et al. (2019). Fraud detection using machine learning in e-commerce. *International Journal of Advanced Computer Science and Applications*, *10*(9).

Schölkopf, B., Williamson, R. C., Smola, A., Shawe-Taylor, J., & Platt, J. (1999). Support vector method for novelty detection. *Advances in neural information processing systems*, *12*.

Sengupta, S., Basak, S., Saikia, P., Paul, S., Tsalavoutis, V., Atiah, F., Ravi, V., & Peters, A. (2020). A review of deep learning with special emphasis on architectures, applications and recent trends. *Knowledge-Based Systems*, *194*, 105596.

Sudjianto, A., Nair, S., Yuan, M., Zhang, A., Kern, D., & Cela-Díaz, F. (2010). Statistical methods for fighting financial crimes. *Technometrics : A Journal of Statistics for the Physical, Chemical, and Engineering Sciences*, *52*(1), 5–19.

Tsay, R. S. (1988). Outliers, level shifts, and variance changes in time series. *Journal of Forecasting*, *7*(1), 1–20.

Vijayakumar, V., Divya, N. S., Sarojini, P., & Sonika, K. (2020). Isolation forest and local outlier factor for credit card fraud detection system. *International Journal of Engineering and Advanced Technology (IJEAT)*, *9*, 261–265.

Weerts, H. J. P., van Ipenburg, W., & Pechenizkiy, M. (2019). A human-grounded evaluation of shap for alert processing. arXiv preprint arXiv:1907.03324

Wu, M., & Jermaine, C. (2006). Outlier detection by sampling with accuracy guarantees. In *Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 767–772).