Enhancing Money Laundering Detection with Semi-Supervised Learning

Timo Schouw

June 13, 2024

Master Thesis

Master Applied AI

Applied University of Amsterdam

Contents

Chapter 1

Introduction

Financial institutions worldwide face the complex challenge of identifying and preventing money laundering. Money laundering involves the illegal process through which illegally obtained money is funnelled into legitimate financial systems, undermining the integrity of financial institutions and facilitating a range of criminal activities [1]. Despite significant investments in anti-money laundering (AML) systems, money laundering techniques continue to outpace traditional detection methods [2, 3].

Machine learning has become a critical tool in enhancing AML systems by offering advanced capabilities to detect patterns indicative of fraudulent activities. Conventional approaches, primarily using supervised and unsupervised learning, have provided valuable insights but exhibit significant limitations [4]. Supervised learning relies heavily on labelled datasets, which is problematic given the often unbalanced nature of datasets used for fraud detection. Unsupervised learning struggles with the high dimensionality and complexity inherent in transaction data [5].

To address these challenges, recent research has explored the potential of semi-supervised learning techniques. These methods combine supervised and unsupervised learning elements, leveraging limited labelled data alongside a larger dataset of unlabeled data. This hybrid approach is particularly promising in enhancing the model's ability to detect a broader spectrum of suspicious activities while mitigating the drawbacks of traditional methods[6, 7].

However, despite the theoretical advantages and preliminary successes, there remains a gap in the literature regarding the practical implementation and performance improvements offered by semi-supervised learning in AML detection. Existing studies have highlighted the potential but lack evaluations demonstrating how these techniques perform in certain settings compared to traditional methods. This research aims to fill this gap by researching the effectiveness of semi-supervised learning techniques in AML systems, providing an analysis of their capabilities to improve detection rates and reduce false positives.

1.1 Problem & Research Question

Problem statement

Financial institutions face significant challenges in effectively identifying and preventing money laundering, the illegal process by which illicit money enters legal financial channels [1, 2, 3]. Although machine-learning techniques are used in anti-money laundering (AML) systems to detect fraudulent transactions, traditional methods such as supervised learning and unsupervised learning may have inherent limitations [6]. The methods are mainly limited by unbalanced datasets not representing the diverse spectrum of criminal activity associated with money laundering [4], but also due to the high dimensionality in transaction data [5]. Previous research suggests that combining supervised and unsupervised learning through semi-supervised approaches is promising for improving

fraud detection [6, 7]. However, there is a gap in the literature that researches how semi-supervised learning techniques improve the performance of fraud detection.

Research Questions

A main research question followed from the problem statement, and four sub-questions were used to support answering the main sub-questions. The main and sub-research questions for this research are as follows.

Main Research Question

- How can the performance of AML systems be improved using semi-supervised learning techniques for fraud detection, and what metrics determine the performance quality?

Sub Research Questions

1. What are the challenges of using supervised or unsupervised learning techniques for AML systems?
2. What metrics determine the quality of machine learning architectures in fraud detection?
3. What semi-supervised architectures are available for fraud detection in AML systems?
4. How do semi-supervised architectures perform compared to supervised and unsupervised architectures?

1.2 Literature Review

AML Methods

Financial institutions are required by law to identify any transactions that may be related to money laundering. Each region or country has its own rules. Take the Netherlands, for example. The Netherlands has the Money Laundering and Terrorist Financing (Prevention) Act ('WWFT') [8], which is based on the international standards of the Financial Action Task Force (FATF[1]). The WWFT provides a legal framework for Dutch financial institutions that requires them, among other things, to detect and signal money laundering transactions. However, each institution has its own specific anti-money laundering (AML) approach, according to [9]. For instance, small investment banks require a different approach compared to large banks, as their customers have different types of transactions. Nonetheless, the AML process generally involves two stages. First, a profile is created of the new customer. Second, their transactions are monitored. According to [10], the first stage of client risk profiling assigns a general risk score to clients. The second stage is fraud detection, which should indicate fraudulent activity. This is done using an algorithm that analyzes transactions and raises an alarm if it detects any fraudulent behaviour. The alarm is then sent to a specialised analyst who reviews it to determine if it is a legitimate concern. A review paper on fraud detection [11] provided a survey on machine learning algorithms and methods applied to fraud detection. The authors divided the task into various aspects and discovered a preference among studies for solutions capable of detecting previously identified suspicious transactions. Further, anomaly detection algorithms emerged as the second most favoured approach, although it was noted that anomaly detection identifies transactions departing from the norm, which may not inherently signify money laundering activity. Within [11], an array of machine learning techniques were reviewed, including supervised learning, unsupervised learning, graph analysis, and semi-supervised learning, all widely deployed as algorithms for fraud detection in AML systems.

[1] https://www.fatf-gafi.org./

Supervised learning techniques for fraud detection

Research [12] reviewed supervised methods that are used in other research and showed that Support Vector Machine (SVM), Random Forrest (RF), and Logistic Regression (LR) are used methods. Study [13] represented a solution by creating a system from ML to analyse group behaviour in financial transactions. The authors used RF and SVM as supervised methods. [14] on the other side, studied the interplay of ML and sampling schemes on money laundering detection algorithms. The authors used LG, Decision Tree (DT), RF, SVM and Neural Networks in this research. Their results showed that neural networks may be used as a rare event classification method based on performance, and SVM and RF models showed more positive results in the sampling data. However, they mention the possible bias and need for historical data as limitations in their studies. Both [15, 16] used SVM as a supervised technique to identify suspicious transactions. Their findings suggest that SVM is suitable for the task, with an accuracy of 0.6-0.8, depending on parameter settings. To compare these results [17] proposed an RBF neural network and compared it against an SVM for suspicious transaction detection. The authors showed that their radial basis function (RBF) Neural Network reached an accuracy above 0.80 and a False Positive Rate below 0.1. Improving their SVM with an accuracy of 0.5 and a False Positive Rate of 0.1 on the same dataset. Take in mind that the researchers [15, 16, 17] used different datasets to train and test their models.

Unsupervised learning techniques for fraud detection

Various studies such as [18, 5] have used unsupervised machine learning techniques to identify clusters in unlabelled data that might indicate suspicious transactions. In particular, [18] leveraged the CLOPE algorithm, which is a clustering method that was introduced in [19]. One of the key features of CLOPE is the repulsion parameter, which can be used to adjust how dense clusters must be and thereby control the resulting number of clusters. Compared to K-means clustering the CLOPE outperformed on the dataset of the authors. In the research of A. Barky et. al [5], the authors mentioned that the high dimensionality of the AML data challenges clustering performance. Therefore, they researched if Agglomerative Hierarchical Clustering (AHC) combined with four different dimensionality reduction techniques improves clustering in AML data. From the four techniques used, Kernel Principal Component Analysis (KPCA) performed the highest as a dimensionality reduction method in combination with AHC. Other clustering methods like K-means or DBSCAN are not compared in their research.

Graph Analysis learning techniques for fraud detection

Other research studies the graph analysis technique to detect suspicious behaviour. Both studies [20, 21] introduced their own method to analyse graphs that contain information about the transactions in an AML dataset. At last, study [22] reviewed fraud techniques by researching the effect of graph learning on the classification of money laundering transactions. The authors use a scientific dataset to perform this research. The results show that using their graph deep learning performs well.

Semi-supervised learning techniques for fraud detection

Study [6] proposed a semi-supervised method to detect money laundering transactions in a total of 4889 transactions dataset. They mention that either supervised or unsupervised learning has its limitations with supervised learning not being able to adapt to fraud patterns in an imbalanced dataset, and unsupervised learning not being able to give state-of-the-art results. The authors researched the performance of an Autoencoder (AE) and Variational Autoencoder (VAE), but the used dataset was too imbalanced. Therefore, they generated more data with a Generative Adversarial Network (GAN) to make the dataset more balanced. Adding the generated data improved the performance of the AE and VAE with an accuracy of .93, an AUC of 0.96-0.97 and an FPR of 0.07-0.08. However, the authors mention the slight overfit on the mixed dataset suggesting future work with a more advanced

dataset of transactions. Similarly, [7] researched the employment of semi-supervised graph learning where they classify in graphs which are lower dimensional representations of the transaction data by the use of embedding models. The authors tested different embedding models and classifiers, and the results showed that the Dynamic graph transformer, in combination with XGBoost, provided the highest performance with an Area under the precision-recall curve (AUPR) of 0.833 and an F1-score of 0.832.

Metrics used in literature

Numerous performance measures have been utilized in research studies. For instance [15, 17, 16] have employed both the Detection Rate (DR) and False Positive Rate (FPR) to evaluate performance, while [15] also used Accuracy as a metric. On the other hand, [6, 20, 21, 13, 7] have utilized the Area under the ROC Curve (AUC-ROC) metric, each with a unique application to the graph's axes. [7, 6, 20] have all utilized the F-1 Score as a metric, while [6] also used Accuracy, Precision, and Recall as additional metrics. Lastly, [12], a review paper on fraud detection methods, showed that AUC-ROC and Accuracy were the most commonly used metrics in the literature. Another review paper on machine learning techniques for detecting suspicious behaviour [11] also underscores the significance of accuracy in evaluating an approach. Nevertheless, directly measuring accuracy can prove challenging as it heavily hinges on the type and scale of datasets utilised for assessment. Therefore, it is difficult to compare the results of studies with different datasets, and only conclusions can be drawn from the results of a particular study, ensuring the fidelity of the findings.

Common problems in fraud detection

Review paper [11] mentioned that most datasets used in their review have a small sample size and studies use different features. Studies not being consistent in size and attributes affect the ability to scale effectively to large volumes of transaction data. At last, the authors mention the importance of a balanced dataset since a high imbalance may lead to a misleading performance. Furthermore, According to [4], there is a significant need for realistic and publicly available transaction data. To address this, the authors created a scientific dataset that overcomes the main issues with existing datasets. Their dataset includes the entire money laundering cycle and covers nine sources of criminal activity, including extortion, loan sharking, gambling, prostitution, kidnapping, robbery, embezzlement, drugs, and smuggling. Moreover, the authors established patterns across multiple banks to set a benchmark for state-of-the-art methods capable of identifying money laundering patterns across different banks. This feature is not available in other datasets, making their work a valuable contribution to the field.

Findings wrap-up

The literature review examined Anti-Money Laundering methods, highlighting the supervised, unsupervised, graph analysis and semi-supervised methods together with metrics that are used and common problems found in the literature.

The findings show that the supervised methods Support Vector Machine (SVM), Random Forest (RF) and Logistic Regression (LR) are common methods that provide high performance, but also show the challenge of leveraging well-balanced data. Unsupervised methods, in contrast, identify clusters in unlabelled data. However, clustering techniques are poorly affected by high-dimensional data, so performing dimensionality techniques is necessary. Additionally, graph analysis makes it possible to examine transactions through graph analysis, which also shows good performance.

Furthermore, semi-supervised methods are reviewed. Two architectures were applied. The first balanced the dataset to improve the classifier and the second reduced the dimensionality of the data before classification. Both resulted in high performance. Performance metrics were also reviewed and the literature showed that metrics such

as Accuracy, Detection Rate (DR), False Positive Rate (FPR) and Area under the Curve (AUC) are commonly used metrics in fraud detection models. At last, the problems were discussed which indicated the challenges that unbalanced data may lead to misleading performance.

1.3 Research Outline

The main objective of this research was to enhance the performance of AML systems by employing semi-supervised learning techniques for fraud detection. This objective was pursued through a mixed-methods approach, integrating qualitative and quantitative research. The study first involved qualitative research to determine the stakeholders, the main challenges of ML in fraud detection, and the metrics that could be used. The findings of the qualitative study determined the methodology for the quantitative study, in which the effects of semi-supervised learning on fraud detection were determined.

Next, the quantitative research evolved a methodology to evaluate the performance of semi-supervised and unsupervised models for detecting fraud. Using the synthetic dataset Banksim [23], derived from real data by a Spanish Bank. The study included three primary analyses. The first compared an unsupervised model against a semi-supervised model, the second compared a supervised model against a semi-supervised model, and the last compared two semi-supervised models with each other. The data underwent minimal preprocessing due to its synthetic nature, but feature engineering was crucial, applying statistical measures and the Recency, Frequency, and Monetary (RFM) principle. The dataset was split into a training, evaluation and test set. Four models were used: XGBoost [24], Isolation Forest [25], a self-learning XGBoost [26], and a semi-supervised Isolation Forest [27]. Model performance was evaluated using AUC-PR, precision, recall and confusion matrices, with the best-performing models tested further to a test dataset. Base models that only predicted randomly assigned labels provided performance benchmarks.

Requirements

The final model must adhere to various requirements. The requirements are listed below.

1. Ethical: The data on which the model is trained must not contain personal information about the customer or receiver.
2. Ethical: For each positively predicted suspicious transaction, the model should explain which characteristics were most influential in detecting it.
3. Legal: Although the model does not use personal data during training, it must remain accessible to comply with legal mandates. In particular, the availability of this data is necessary in accordance with Chapter 6 of WWFT [8].
4. Organizational: Creating, training and iterating the model must be feasible within a time frame of 2 full working weeks.
5. Functional: The model must be able to predict suspicious behaviour for a dataset with at least three days of transactions.
6. Functional: The model should provide predictions for a new set of transactions within 5 seconds.
7. Functional: The model must improve a baseline model which randomly predicts a label.
8. Technical: The model must be able to train on one of the following accessible sources: my PC, the HvA's server or Google Colab.

Chapter 2

Qualitative Research

The qualitative research involves three stages. First, a stakeholder analysis is conducted to determine the influence of different stakeholders in fraud detection. Second, interviews were conducted with two data scientists from an international Dutch bank to discuss the use of machine learning in fraud detection, as outlined. Finally, desk research was performed to further map the landscape of semi-supervised learning methods. Section 2.1 describes the methodology used to answer the corresponding sub-question, and section 2.2 shows the findings of the performed method.

2.1 Methodology

The qualitative research aimed to identify the stakeholders, main challenges, and relevant metrics for Machine Learning in fraud detection. To address these objectives, a stakeholder analysis, two interviews, and desk research on semi-supervised learning architectures were conducted. These methods helped answer the sub-questions on the challenges of using supervised or unsupervised learning techniques for fraud detection (sub-question 1), metrics that determine the quality of machine learning architectures in fraud detection (sub-question 2), and what semi-supervised architectures are available for fraud detection in AML systems (sub-question 3).

2.1.1 Stakeholder Analysis

The quantitative research started with a stakeholder analysis to determine the importance and influence of different stakeholders in fraud detection. Stakeholders such as financial institutions, regulators, AML software providers, and other relevant entities were identified based on existing literature. Stakeholders were then prioritized based on their relevance and influence on AML and fraud detection. This prioritization facilitated the creation of a stakeholder map, which provided an overview of the importance of stakeholders in the research context.

2.1.2 Interviews

Following the stakeholder analysis, two interviews were conducted with key representatives of identified stakeholders. The purpose of these interviews was to gather qualitative insights into fraud detection methods and techniques, formulate the problem further, and further identify the gap based on the stakeholders' interests. The aim was to select participants from key stakeholders, and two Data Scientists from an international Dutch bank were interviewed. Both data scientists use machine learning to develop models that detect money laundering fraud in their AML systems. An attempt was made to interview someone from the other essential stakeholder, AML software providers, but there was no response to the interview proposal.

Open-ended questions were asked to get nuanced answers on the following topics: Architectures used in AML systems to detect fraud, Challenges of Machine Learning in fraud detection, Semi-Supervised Learning in fraud detection and quality assessment of fraud detection and AML systems. To support these topics, the following general questions were asked.

- What kind of architecture do you use for fraud detection or related systems?
- What are the current challenges in supervised & unsupervised techniques in fraud detection?
- What is your perception of semi-supervised learning techniques in fraud detection, do you see opportunities or challenges, and why?
- What is your perception of semi-supervised learning techniques in fraud detection, do you see opportunities or challenges, and why?
- What is your assessment of the quality of AML systems?
- What metrics do you think are important for evaluating fraud detection performance, and why?

Sub-question 1 was answered by the questions on the challenges of supervised and unsupervised techniques, while sub-question 2 was answered by the last question on metrics and evaluating the models in fraud detection.

Participants were asked for permission to record and analyze the interviews, ensuring confidentiality and anonymity where necessary. Interview transcripts were carefully analyzed to derive meaningful interpretations and insights. An interview report was made for both interviews and added to Appendix C. Findings from the interviews contributed mostly to the answer to the first two sub-questions, but they also contributed to sub-questions three. More importantly, the findings formed the topics for further research in the desk research phase and the methodology for the quantitative research phase.

2.1.3 Desk Research

After the interviews, desk research followed with the purpose of gaining a further understanding of the landscape of semi-supervised learning architectures and architectures applicable to fraud detection. The main source for this part of the research was a review article by van Engelen and Hoos [28], categorising semi-supervised learning methods for classification tasks into two primary approaches: inductive and transductive. The review by van Engelen and Hoos provided a broad overview of semi-supervised learning methods and their applications. The taxonomy of semi-supervised learning architectures has been developed, and various applicable architectures supported by research have been elaborated.

2.2 Findings

This findings section covers the findings on the stakeholder analysis, interviews and desk research. Appendix A shows the stakeholder map created to support the analysis. The interview reports for both interviews can be found in Appendix C.1 and C.2. A conclusion on the findings is provided at last.

2.2.1 Stakeholder Analysis

Money laundering is an illegal process, and it has a significant impact on an economy [29]. The Inter-governmental Financial Action Task Force (FATF) was established to counter money laundering and reduce economic impact. Their objective is to set standards and promote the effective implementation of legal, regulatory and operational

measures to combat money laundering [1], making them the first stakeholder. Next, the government and its agencies are the second stakeholders as they implement the recommended regulations established by the FATF. Third, financial institutions must follow government regulations that carry heavy fines for non-compliance. Financial institutions include commercial banks, investment banks, credit unions, insurance companies, brokerage firms, money service businesses, payment processors, and crypto-related companies. The fourth stakeholder, Designated non-financial businesses and professions (DNFBP), are businesses and professions of which nature may be suspicious of money laundering [30]. DNFBPs identified by FATF are Casinos, real estate agents, dealers in precious metals or stones, lawyers, notaries, other independent legal professionals and accounts, and trust and company service providers [30, 31]. The FATF has set up recommendations for DNFBPs to help them mitigate money laundering activities. Focusing more on the anti-money laundering techniques used, software providers that create anti-money laundering techniques are the fifth stakeholder. They aim to develop AML compliance software that supports financial institutions when they do not provide solutions. This research is essential for AML software providers because the research question aligns with their solution. The last stakeholder is the society because AML is part of countering illegal activities. Illegal activities are always part of society as they may have an effect on everyone.

Each stakeholder was ranked as essential, important and interesting using a stakeholder map to determine the importance of this research for each stakeholder. The essential stakeholders are the financial institutions and AML software providers because they specifically use the techniques examined in this study. The key stakeholders are the FATF and government agencies because they create rules for the application of AML techniques but do not actually create or use them themselves. The interested stakeholders are society and NFBPs because they are related to AML and AML techniques are of interest to them because of their nature, but they are not essential to the application of this study. The stakeholder map can be found in Appendix A.

2.2.2 Interviews

The overall answers to the questions asked in both interviews are provided in Appendix C.1 and Appendix C.2. The findings cover the architectures for fraud detection, challenges of machine learning in fraud detection, semi-supervised learning for fraud detection and quality assessment of fraud detection.

Architectures used in fraud detection

When the participants were asked about the architectures used in fraud detection, they said it depends on the use case. For supervised problems, simple but strong architectures are used when the model is aware of the kind of money laundering transactions it seeks. Think of models like XGBoost and Random Forest. It is important that the models are interpretable and stable, which is why models like neural networks are avoided. However, a certain percentage of labels is needed to train the supervised models, which is not always the case when applying fraudulent transactions as they are often sparse. For this reason, new labels of money laundering transactions are trying to be classified by unsupervised learning. Both participants mentioned the use of anomaly detection as an unsupervised method, and specifically, the Isolation Forest architecture is used commonly for unsupervised learning. The anomaly detection model should eventually provide sufficient labels for a supervised model to constantly predict those labels.

Challenges of Machine Learning in fraud detection

The interviews also revealed that the anomaly detection model often provides insufficient labels for a properly supervised approach to training due to the limited information on which to train. Following the interviews, this is a major challenge in using machine learning in AML systems. When an unsupervised learning method is deployed to detect 'unknown unknowns', the model can not properly weigh the underlying importance of features, for which a supervised approach is needed. Further, the interviews revealed that when a label is established, it should be

reviewed by an analyst. This may cause redundant work for the analysts as the anomaly detection model is prone to detecting false positives. Lastly, seasonal effects can cause difficulties in detecting anomalies due to different behaviours of money laundering in seasons.

Semi-Supervised learning for fraud detection

One participant highlighted semi-supervised learning as a promising approach to address the challenges discussed. The participant suggested that semi-supervised learning could bridge the gap where unsupervised techniques identify some patterns in the data but not enough to train a fully supervised model. It has the potential to either assist in creating enough labels for supervised learning or, when performing well, be directly applicable to the specific use case. However, further research is needed to determine the most effective semi-supervised architecture for tackling these challenges.

Quality Assessment of fraud detection

Discussions with participants revealed two key aspects of quality assessment for fraud detection systems. The first aspect focused on overall quality requirements. Model stability emerged as a crucial factor, ensuring consistent predictions when re-trained and tested on the same data. High performance achieved through chance is insufficient. This stability can be evaluated by measuring the overlap between predicted labels across retraining runs. The second aspect addressed performance metrics for the fraud detection models. One participant advocated for the Area Under the Precision-Recall Curve (AUC-PR) due to its stability in imbalanced datasets, where positive fraud cases are much rarer than negative ones. AUC-PR considers the trade-off between precision (correctly identifying fraud) and recall (capturing all true fraud cases), making it more reliable. Another insightful way to determine the performance is by looking at the precision-recall curve and looking at the 90% Recall part of the curve. That determines how well the model predicts fraud.

The participant cautioned against using metrics like F1-score and AUC-ROC. F1-score assumes a 50% probability threshold, which is not applicable in money laundering situations where fraud is much less frequent, typically below 5%. Similarly, AUC-ROC can be misleading in imbalanced datasets. Due to imbalanced data, the points in the ROC graph are clustered into one place, and the AUC metric is defined by just a few points, leading to instability. At last, the participant cautioned for a low overall accuracy given the inherent rarity of fraud cases.

2.2.3 Desk Research

To gain a comprehensive understanding of semi-supervised learning architectures, a survey paper by van Engelen and Hoos [28], was examined. Their work categorized semi-supervised learning for classification tasks into two primary approaches: inductive and transductive. Inductive approaches aim to build a generalizable classification model that can make predictions on unseen data, leveraging both labelled and unlabeled data points. In contrast, transductive semi-supervised models make predictions directly without training a classifier model. This allows transductive models to leverage all the data and adapt constantly to new data that is added [28, 32].

Taxonomy of Semi-supervised learning

The authors of [28] mentioned graph-based models as transductive models, underscoring the limitation for AML applications. Transductive methods, which rely solely on data point relationships for prediction, are unsuited to fraud detection due to their focus on the networks of full datasets. Conversely, inductive learning approaches, capable of leveraging both labelled and unlabeled data, offer generalizability, making them more appropriate for fraud detection tasks where unseen data prediction is paramount. For this reason, transductive methods are beyond

the scope of this research. Transductive methods demand certain computational resources to learn the networks and graphs as well, underscoring the limitation for this research.

Building on the study's findings of [28], which classifies inductive methods into three different models, this desk research went deeper. The survey paper of Miryam Elizabeth Villa-Pérez et al.'s article on semi-supervised anomaly detection (SSAD) [33] was analysed, and SSAD was identified as an overlooked inductive semi-supervised learning method. Motivated by interviews that suggested the relevance of anomaly detection for the detection of new fraud instances, SSAD was integrated into this analysis. The four inductive learning architectures were assessed on their applicability within the context of fraud detection and the scope of this research.

1. **Intrinsically Semi-Supervised Methods:** These types of architectures integrate unlabelled data directly into the learning process's objective function or optimisation procedure. These methods often stem from supervised learning approaches extended to the semi-supervised domain, where unlabelled data is incorporated into the objective function of the classifier. However, most semi-supervised methods rely on the low-density assumption, which assumes that if two input points are close together in input space, the corresponding labels are also close together [28]. This is difficult in fraud detection because the density between fraudulent transactions can be very high as they are sparse, which counteracts this assumption.

2. **Unsupervised Preprocessing:** In this approach, labelled and unlabelled data are used in two separate phases. Data manipulation techniques, such as dimensionality reduction or feature extraction, are used before feeding into a supervised learning algorithm. The cluster-then-label technique is a widely used method for unsupervised preprocessing, but concerns about low-density assumption arise for this method too. The scarcity of labels for fraudulent transactions for unsupervised clustering indicated that unsupervised preprocessing might cause problems in fraud detection applications due to this assumption.

3. **Wrapper methods (pseudo-labeling):** As described in [28], wrapper methods use existing labelled data alongside predictions generated from unlabelled data to train supervised learners. This approach essentially creates "pseudo-labels" for the unlabelled data points based on the predictions of the initial classifier. These pseudo-labelled points are then combined with the original labelled data to re-train the classifier. Wrapper methods show potential to be used for AML systems because they can train a model and evolve along the process continuously. This benefits AML systems as fraud methods can change over time, but they also have limitations. The accuracy of the initial classifier can significantly affect the quality of the pseudo-labelled data, potentially introducing noise and misleading the learning process. This inherent drawback can hinder the ability to identify true fraudulent transactions.

4. **Semi-supervised anomaly detection:** This approach leverages a small set of labelled normal data points along with a larger set of unlabeled data to build a model for anomaly detection. By learning the characteristics of normal data from the labelled examples, the model can identify patterns in the unlabeled data that deviate significantly, potentially indicating anomalies. This technique is particularly advantageous in scenarios where obtaining labelled anomaly data is difficult, but a vast amount of unlabeled transaction data is readily available. [33]

From these findings, it can be concluded that both wrapper and semi-supervised anomaly detect emerge as potential inductive methods for applying fraud detection. Intrinsic semi-supervised learning and unsupervised preprocessing are assumed to be inapplicable to fraud detection due to the assumption of low density, while SSAD and wrapper methods show potential to improve fraud detection. Both methods are further analysed for architectures that potentially contribute to answering the research question.

Applicable architectures

Two wrapper method architectures and two semi-supervised anomaly detection architectures were analysed to further understand the inductive learning methods.

The first architecture is a wrapper method that utilizes pseudo-labelling to improve anomaly detection with limited labelled samples by J. Yoon et al. [34]. The authors created a framework called SPADE that utilizes an ensemble of one-class classifiers to estimate the pseudo-labels of the unlabeled data. It also employs partial matching to automatically select the critical hyper-parameters for pseudo-labelling without relying on labelled validation data, which is crucial given limited labelled data. The authors evaluated the framework financial fraud detection datasets and compared them to a supervised, unsupervised and semi-supervised model. The results showed that the SPADE had the highest AUC on two financial datasets. However, the authors used only the AUC-ROC to evaluate the results. Research [26] also introduced a self-learning XGBoost. The authors created an algorithm where an initial classifier is trained on labelled data used to predict labels for unlabelled data and then retrain the model iteratively with the most confident predicted labelled added to the training datasets. The model was experimented on 25 benchmark datasets in the research, achieving high accuracy on most datasets across different labelled percentages.

The architecture in the study by H. Xu et al. [35] was analyzed, focusing on deep semi-supervised anomaly detection in scenarios with limited labelled data. Their proposed architecture, RoSAS, optimizes anomaly scores using contamination-resistant continuous monitoring signals. RoSAS includes a feature representation module and an anomaly scoring module, which enhance the anomaly detection process. RoSAS propagates anomaly information through interpolation from labelled to unlabeled data, creating elevated samples with continuous anomaly values. This approach allows direct optimization of the anomaly scoring mechanism and ensures consistency between augmented samples' scores and original data scores, resulting in smoother anomaly score descriptions. A feature learning-based objective further isolates labelled anomalies in intermediate representations, making the network more robust against contamination. RoSAS demonstrated superiority on a fraud detection dataset with an AUC-PR score of 0.831.

The second semi-supervised anomaly detection model is the semi-supervised isolation forest [36]. The proposed architecture optimizes anomaly scores using both labelled and unlabelled data. The architecture includes a probabilistic split distribution module and a feature selection module, which enhances the anomaly detection process. It multiplies anomaly information by combining labelled and unlabelled data, creating splits that effectively isolate anomalies. The tree induction further isolates labelled anomalies in the tree structures, making the model more robust against noise. The model was evaluated and consistently performed across varying percentages of labelled data, highlighting its promising potential.

2.2.4 Conclusion

In conclusion, the essential stakeholders are financial institutions and AML software providers because they implement fraud detection in AML systems directly in their business. When labels are known, architectures like XGBoost and random forest are used, while isolation forest is used for anomaly detection to gather new labels. The key challenges are that supervised learning requires sufficient labelled data, which unsupervised models often fail to provide, and poorly performing unsupervised models can cause redundant work for analysts. Interviews revealed that AUC-PR, Precision, Recall, and model stability are important for assessing model quality. Finally, desk research showed that the landscape of semi-supervised learning includes inductive and transductive learning, with inductive learning being applicable to fraud detection due to training on both labelled and unlabeled data. Inductive learning methods wrapper methods (pseudo-labeling), and semi-supervised anomaly detection were considered potential methods for improving fraud detection.

Chapter 3

Quantitative Research

This chapter details the quantitative research employed to evaluate architectures for semi-supervised learning in fraud detection. The research provided a systematic approach to training and evaluating semi-supervised models and compared them to corresponding supervised and unsupervised models to validate their effectiveness. In section 3.1, the methodology is elaborated, followed by 3.2, which shows the findings of the executed method and an answer to sub-question four based on these findings.

3.1 Methodology

The main objective of the quantitative research was to evaluate semi-supervised models and answer the fourth sub-question on how semi-supervised architectures perform compared to supervised and unsupervised architectures. To achieve this, three analyses were set up, each evaluating two models on different training data. The methodology section describes the three analyses first, followed by the models, data collection, feature engineering & pre-processing, data splitting and last, the model training & evaluation.

3.1.1 Analyses

Three analyses were conducted using these four models. All three analyses aimed to gather insights on whether semi-supervised learning could extend unsupervised or supervised models in AML systems. The first two analyses compared the performance of a super- or unsupervised model against a semi-supervised model when there are different percentages of fraudulent transactions in the training data. The methodology for these analyses was based on the findings of the interviews (Section 2.2.2). These interviews mentioned that unsupervised anomaly detection often does not provide enough labels for a full training data set on which a supervised model could train. Therefore, by comparing a semi-supervised model with a super- or unsupervised model, this method intended to gather quantitative results on whether semi-supervised models could close that gap. The aspect of different percentages of fraudulent transactions in the training data resulted from [37], in which the authors used semi-supervised learning on fraud detection. The authors mentioned the effect of how imbalanced the dataset is, and to tackle that problem, this approach was executed.

A third analysis was done, which compared the performance of both semi-supervised models when different distributions of labelled and unlabelled transactions were available in the data. According to [28], semi-supervised models may differ in performance with different distributions, resulting in the aim to evaluate the performances between the two models as a function of known labels. Both [26, 36] also evaluated the model on different percentages of known labels, supporting the chosen methodology. To summarize, the three analyses are described as follows:

1. **Analysis 1:** This analysis compares the unsupervised model with the semi-supervised anomaly detection when trained on different percentages of fraudulent transactions. Comparing an unsupervised model against a semi-supervised model aimed to test if the semi-supervised model could tackle the challenge of unsupervised learning in fraud detection. By evaluating these two models against each other with different percentages of labels, a conclusion was drawn about the impact of using known and unknown labels on different distributions of fraudulent transactions in the training dataset [37].

2. **Analysis 2:** The second analysis does the same as analysis 1 but leverages a supervised and the wrapper method. This approach is next to the challenges that arose from the interviews, also inspired by the methodologies used in [37], where a similar approach was used. This analysis aimed to see how the models performed against each other and whether semi-supervised models are a possible addition to fraud detection.

3. **Analysis 3:** This analysis compared the two semi-supervised methods based on the percentage of known labels. Semi-supervised models may differ in performance when a certain percentage of labels is known [28]. Therefore, this analysis was used to evaluate the performances of different distributions of labelled data in the training data.

3.1.2 Models

The analyses required the selection of the four models. To provide a quantitative analysis that supports answering sub-question three, it was tested if a supervised or unsupervised model could be added with unlabelled or labelled data to close the gap in a situation where there is not enough data to fulfil one of both completely. Following the findings in Section 2.2.2 and 2.2.3, XGBoost was chosen as it proved to be ideal based on these findings. Secondly, isolation forest was chosen as an unsupervised model because the findings of 2.2.2 revealed that this model is widely used for fraud detection within banks. A semi-supervised addition to both models was chosen to further close the gap between the mentioned models. To clarify, a wrapper method was used to help the supervised model when the dataset contains unlabelled data, and a semi-supervised anomaly detection method was used to help the unsupervised model when there is unlabelled data. More specifically, a self-learning XGBoost algorithm and a semi-supervised isolation forest algorithm were chosen. The corresponding parameters and the algorithms of each model will be further elaborated.

Supervised model

XGBoost was introduced by T. Chen and C. Guestrin in [24]. It is a tree-boosting method that works with Gradient Boosting to correct the model's errors iteratively and reach for the best performance. The XGBoost was mentioned in both interviews (Section 2.2.2) as a commonly used supervised model due to its strong performance on imbalanced datasets. However, the XGBoost also covers a lot of hyperparameters that need tuning.

The following six parameters were considered as most important based on the results of [38], who applied hyperparameter tuning on an imbalanced credit card dataset for fraud detection: (i) *n_estimators* specifies the number of trees to build in the model, (ii) *min_child_weight* determines the minimum sum of instance weight needed in a child, (iii) *max_depth* defines the maximum depth of a tree, (iv) *learning_rate* controls the step size shrinkage used to prevent overfitting (v), *gamma* is the minimum loss reduction required to make a further partition, and (vi) *colsample_bytree* specifies the fraction of features to be randomly sampled for each tree. For all other parameters, the standard value is used. The tuning of each parameter is done through an iterative process. The final parameters of the model were provided in Appendix E.

Unsupervised model

The isolation forest model, introduced by F.T. Liu et. al. in [25], was chosen as the unsupervised model based on the findings of Section 2.2.2 and 2.2.3. The isolation forest model is also a tree-based model and isolates certain points based on trees created by the model. Anomalies can be detected by calculating distances between nodes, and based on these distances, an anomaly score can be conducted. A threshold value then determines what is labelled as an anomaly and what is not. Based on the findings of the interviews (Section 2.2.2). The isolation forest turned out to be used for fraud detection due to its strong anomaly detection performance, but a downside of the model is the noise and instability the model can give.

There are four parameters[1] considered to tune for the Isolation Forest. (i) $n\_estimators$: the number of base estimators in the ensemble, (ii) $max\_samples$: The number of samples to draw from X to train each base estimator, (iii) *contamination* : The amount of contamination of the data set, i.e. the proportion of outliers in the data set, used when fitting to define the threshold on the scores of the samples. (iv) $max\_features$: The number of features to draw from X to train each base estimator. Domain knowledge is needed to determine the model's parameters because it is unsupervised, but through an iterative process, the model was improved by tuning the parameters.

Semi-superivsed iForest

The semi-supervised isolation forest model introduced by [36] was chosen as a semi-supervised anomaly detection model. The semi-supervised isolation forest model leverages labelled and unlabeled data when building the isolation trees, enhancing the anomaly detection performance by using the labelled instances to guide the splitting criteria. This architecture was more preferable than RoSAS [35], due to the required computational resources and complexity of the model. This would not have met requirement eight.

The Algorithm for building the tree is shown in Algorithm 1. The semi-supervised isolation forest model computes anomaly scores similar to those of unsupervised isolation forests. The model was available via the GitHub page of the author [2], making it compatible with requirement five.

The main difference between the isolation and semi-supervised isolation forest models is the tree induction. Each tree determines the best way to split the data by evaluating the labelled and unlabelled components. The labelled component calculates and compares the entropies of the labels to determine a split. The unlabelled component calculates the variance of the data on either side of the potential split points and combines the variances with the proportion of samples on each side to compute a weighted measure that incorporates both variance and entropy. It then determines the split based on the combined scores of the two components. The incorporation of labelled and unlabelled data into the tree induction provided an advantage for fraud detection. However, the downside of applying this architecture is the increased complexity, leading to more computational usage.

Four parameters are used in the semi-supervised isolation forest model: (i) *ntrees* indicates the number of trees in the ensemble. A higher number of trees can increase the model's robustness and the computational load. (ii) *sample* is the number of samples to draw from X to train each node to help ensure tree diversity. (iii) *nattr* determines the number of features to consider for the split distribution and helps manage the model's complexity. (iv) *max_depth* is the maximum depth of each tree.

Semi-supervised XGBoost

The architecture of the wrapper method, self-learning XGBoost, was based on [26]. The authors introduced a self-learning XGBoost algorithm that leverages labelled and unlabelled data. The pseudo-code of the self-learning XGBoost model is displayed in Algorithm 2. This algorithm was chosen over the SPADE framework [34] because

[1] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html
[2] https://github.com/stralu9/Semi_Supervised_Isolation_Forest

the SPADE framework was complex to implement and would not have met requirement five. The self-learning approach could help improve the model by iteratively retraining with the most confident predictions from the unlabelled data. The disadvantage is that the process involves multiple iterations of training and prediction, which makes it computationally intensive to perform.

The algorithm used the training and evaluation dataset (D_{train}, D_{val}). It splits the training dataset D_{train} into labelled and unlabelled instances (L^0 and U^0). The first parameter, $MaxIter$, determined the maximum number of iterations allowed for self-learning before the loop ends, while the second parameter was the percentage of instances (T) that can be allowed at each iteration of the self-learning process, which is proportional to the size of the labelled instances (L^0). The XGBoost model was trained on the initial labelled set (L^0), after which the self-training algorithm started. The trained XGBoost model predicted labels for U^0 and was sorted descending on prediction probabilities in matrix M_{pr}. Next, the most confident predicted instances were selected on $T * size(L^0)$, and the most confident predicted instances were stored in the new matrix M_{mcp}. Matrix M_{mcp} was then used to remove the instances from U^i and add the instances to L^i. At last, the XGBoost was retrained on L^i, and the next iteration started. After $MaxIter$ iterations, an augmented labelled dataset L^{last-i} was created, which was used to train the XGBoost again, and that model was used to evaluate on D_{val}. This research used a $MaxIter$ of 256 and T of 100 to meet the functional requirement, and T follows the % of fraudulent transactions.

Algorithm 1 Semi-Supervised Isolation Forest (SSIF) Algorithm [36]

Input:
X - The dataset with labeled and unlabeled instances
Y - Labels for the dataset, where $Y[i] = 1$ for anomaly, $Y[i] = -1$ for normal, and $Y[i] = 0$ for unlabeled
ntrees - Number of trees in the ensemble
sample - Number of samples to draw from X for each tree
nattr - Number of features for split distribution computation
max_depth - Maximum depth of the trees

1. Initialization:
$L^0 = \{(X_i, Y_i) \mid Y_i \neq 0\}$
$U^0 = \{X_i \mid Y_i = 0\}$
Train initial SSIF classifier on L^0

2. Build Forest:
for $i = 1$ to *ntrees* **do**
    Sample *sample* instances from X to form X_p and corresponding Y_p
    Train tree T_i on X_p and Y_p
    Add T_i to the forest
end for

3. Compute Anomaly Scores:
for each instance x in the dataset **do**
    Initialize path lengths array $paths = \{0, \ldots, 0\}$
    for each tree T_i in the forest **do**
        Compute path length p_i for x in T_i
        Add p_i to *paths*
    end for
    Compute anomaly score $S_x = 2^{-\frac{\text{mean}(paths)}{c}}$
end for

4. Output:
Use the computed anomaly scores to identify anomalies in the dataset

Algorithm 2 Self-learning XGBoost algorithm [26]

Input:
D_{train} - The training dataset with 2M, 4M, ..., 10M labelled instances
L^0 - Set of labelled training instances, $L^0 \subseteq D_{train}^{A1,..M}$
U^0 - Set of unlabelled training instances, $U^0 \subseteq D_{train}^{A1,..M}$
$MaxIter$ - Maximum number of iterations
T - % Accepted instances per iteration
L^i - Enlarged labelled set during the i-th iteration
U^i - Reduced unlabelled set during the i-th iteration

1. Initialization:
$L^i = L^0$
Train XGBoost classifier on initial labelled set (L^0)

2. Self-train:
for i iterations where $0 \leq i < MaxIter$, while $size(U^i) > 0$ **do**
    a. Utilize XGBoost on U^i and store the predictions on M_{pr} matrix
    b. Sort M_{pr} matrix descending, according to the prediction probabilities of XGBoost
    c. Select $T * size(L^0)$ most confident predicted instances and store them in M_{mcp}
    d. Remove M_{mcp} from U^i and add them in L^i using as labels the XGBoost predicted labels accordingly and retrain the classifier on L^i
end for

3. Output:
Use the augmented labelled set (L^{last-i}) to train XGBoost and apply it on the unknown test instances to produce predictions.

3.1.3 Data Collection

A quality dataset with accurate labels was crucial for the quantitative research. However, transactional data is typically confidential and private, making using real data impractical. Consequently, a synthetic dataset was necessary for this research. The scientific transactional dataset introduced by E. A. Lopez-Rojas and S. Axelsson in [23], Banksim, was used in this research. Banksim is created through an agent-based simulator on a sample of aggregated transactional data provided by a bank in Spain. The dataset was available for download via Kaggle [3].

The dataset leveraged statistical and Social Network Analysis (SNA) of relations between merchants and customers. It contains approximately six months of transactions, which resulted in 594643 rows, of which 587443 are labelled normal, and 7200 are labelled fraudulent (1.21%). The dataset includes six main features: Step, Customer, Merchant, Category, Amount and Fraud. The step represents a day of commercial activity, and the dataset contains 180 steps. The customer is the one who performs the transaction. Every Customer also has a unique ID with a related zip code, Age, and Gender. To compile with requirement 1, Age and Gender are deleted from the dataset. The merchant is the one who serves the customer and can also be seen as the receiver. Each unique merchant has an ID and a related ZipCode in the dataset. There were 50 unique merchants in the dataset. The ZipCode is deleted from the dataset to meet the first requirement for this research. The category entity shows the purchased service or goods category with the corresponding transaction. 15 different categories were used in the dataset. The third feature is the transaction amount, showing the exact amount in euros. At last, the fraud feature showed whether the transaction was fraudulent or not.

[3] https://www.kaggle.com/datasets/ealaxi/banksim1

Pros and Cons of Banksim

Both the positive and negative sides of using the dataset were considered to be aware of possible effects that may have arisen during the training and evaluating phase that could affect the results and answering of research questions.

- **Pros**
  - The synthetic approach ensures the privacy and confidentiality of the data, meeting the first ethical requirement.
  - The statistical properties and behaviour of the transactions are preserved by using data from a Spanish Bank. This increases the validity of machine learning models trained on this data, making them more applicable to real-world scenarios.
- **Cons**
  - Despite the techniques used to create the synthetic dataset, it may still lack certain nuances that are present in real data. These limitations can lead to models that perform well on synthetic data but may not effectively generalise to real transaction data in live fraud detection systems.
  - The dataset contains many data points which require resources to use the dataset. This may be inherent in the functional requirement, and a data cut was necessary for the semi-supervised isolation forest.
  - The data initially contains Gender and Age groups for a customer, which may cause indirect biases based on the customers of the Spanish Bank, although not trained on this information.

3.1.4 Feature Engineering & Pre-processing

After the data collection phase, feature engineering was performed, which was a important step for the validity of this research. In the initial dataset, only the Step, Merchant, Category and Amount features could be used for training, as Fraud is the prediction label. However, machine learning models rely very much on features to classify the labels. Additionally, fraudulent transactions are rare in a fraud detection dataset, so specific features are necessary to support the model in classifying these fraudulent transactions [39]. Although fulfilling the ideal feature engineering is beyond the scope of this research since it mostly requires specific domain knowledge, some feature engineering steps were necessary for the models to have some performance to compare to each other. Performing a more complex feature engineering would supply higher performances, but due to time limitations, basic feature engineering was performed. Two feature engineering steps were performed: (i) calculating statistical features per alert by aggregating on current features in the dataset, which was based on the method applied in [40], and (ii) applying the Recency, Frequency and Monetary (RFM) Principle to each transaction per alert which was based on the method of [39].

Statistical features per Alert

First, the statistical features were calculated using the method of [40]. For each customer, the minimum, mean, median, maximum, standard deviation, count and sum per unique Marchant and Category were calculated. This resulted in 462 features $((7 * 50) + (7 * 16))$. All features were scaled to mean zero and unit variance to make them compatible with training the model.

RFM principle

The second method was based on the research of [39], who did research on data engineering for fraud detection. This research was applicable due to its focus on fraud detection, and the RFM principle was applied as their feature engineering process. That process was used to apply, and the recency, frequency and monetary were calculated. Recency measures how long ago a certain event took place, whereas frequency counts for the number of specific events per unit of time and the monetary feature measures the intensity of a transaction.

Recency

The recency was calculated based on equations 3.1 and 3.2

$$x_i^{M,C,recency} = \exp\left(-\gamma \cdot \Delta t_i\right) \quad \text{where} \tag{3.1}$$

$$\Delta t_i = \min\left\{\text{days}\left(x_i^{\text{time}}, x_j^{\text{time}}\right) \mid x_j^{id} = x_i^{id} \text{ and } \left(x_j^M = x_i^M \text{ for } x_i^{M,recency} \vee x_j^C = x_i^C \text{ for } x_i^{C,recency}\right)\right\}_{j=1}^{N} \tag{3.2}$$

The recency of a certain transaction given a specific Merchant (M) or Category (C) ($x_i^{M,C,recency}$) was calculated by the exponential of $-\gamma * \Delta t_i$. Equation 3.2, is the condition of Δt_i for 3.1, which is defined as the minimum days between a certain transaction given that the Customer and Merchant or Category are equal. To complete the calculation, γ is the parameter that determines how fast the recency decreases. For larger values of γ, recency will decrease quicker with time and vice versa. Small recency shows atypical behaviour and might indicate fraud, and for that reason, the standard γ of 0.02 was chosen. This decision was based on the figure displayed in Appendix B, which is a graph indicating the effect of γ on recency over a time interval [39]. In this graph, it can be seen that a γ of 0.02 shows a not-so-sensitive movement over when the time interval increases, which is assumed to reflect fraud detection as it is not suspicious to have many transactions in a short period and otherwise have a certain specific transaction while this has not happened in a long time is suspicious. To execute the calculation of this feature, the data was grouped on customer_id and merchant_id or category_type, and the Δt_i was calculated followed by $\exp\left(-0.02 \cdot \Delta t_i\right)$. This resulted in a recency for each merchant and customer combination and for each category and customer combination. For every unique combination of features, the recency was defined to be zero.

Frequency

The second feature of the RFM principle is the frequency, which calculates how many transactions were made during a sliding time window that satisfies predefined conditions. It is calculated by the following equations:

$$x_i^{\text{freq}} = \left|D_{t_p,i}^{\text{freq}}\right| \tag{3.3}$$

Equation 3.3 is further calculated by equation 3.4:

$$\begin{aligned} D_{t_p,i}^{\text{freq}} &= AGG^{\text{freq}}\left(D, i, t_p, \text{M} \vee \text{C}\right) \\ &= \left\{x_j^{\text{amount}} \mid \left(x_j^{id} = x_i^{id}\right) \text{ and } \left(x_j^{\text{M}} = x_i^{\text{M}}\right)\right. \\ &\quad \left. \vee \left(x_j^{\text{C}} = x_i^{\text{C}}\right) \text{ and } \left(\text{days}\left(x_i^{\text{time}}, x_j^{\text{time}}\right) < t_p\right)\right\}_{j=1}^{N} \end{aligned} \tag{3.4}$$

In 3.4, the $AGG\left(\cdot\right)$ aggregates the transactions of D into a subset associated with a transaction i with respect to the time frame t_p. In equation 3.3, x_i^{freq} is the timestamp of transaction i. Further in equation 3.4 is x_i^{amount} the amount of a certain transaction i, referred to the amount feature in D. The frequency is calculated with the same requirements as the recency equation, meaning the first condition is an equal customer_id, and the second

condition is either an equal merchent_id or an equal category_type. The third condition is determined by t_p because the chosen sliding time window is considered at this condition with days $\left(x_i^{\text{time}}, x_j^{\text{time}}\right) < t_p$. The authors of [39] proposed a fixed time window of 90, 120 or 180 days and since Banksim contains around 180 days, a t_p of 180 was chosen to make sure the whole dataset is used as time window. It has been assumed that it is more suspicious when a transaction happens non-frequently, and the higher the sliding window, the more suspicious it gets. The equation is performed by the aggregation function in Python, and the count function calculates the frequency per alert for a given unique transaction within a sliding window.

Monetary

The third and last feature used for this research is the Monetary value, which focuses on the amount that is transferred within a given time window. It leverages same the aggregation and conditions $D_{t_p,i}^{\text{freq}}$ as the frequency feature, and the monetary feature is calculated by the following equation 3.5:

$$x_i^{\text{total}} = \sum_{j=1}^{N} x_j^{\text{amount}} I\left(x_j^{\text{amount}} \in D_{t_p,i}^{\text{freq}}\right) \tag{3.5}$$

This equation 3.5 represents the total amount of transactions of x_i, which is calculated by the sum of each element given a certain condition. The summation includes an indicator function $I\left(x_j^{\text{amount}} \in D_{t_p,i}^{\text{freq}}\right)$, which takes the value of x_j^{size} if it is an element of $D_{t_p,i}^{\text{freq}}$ and 0 otherwise. In other words, x_i^{total} is the sum of the amount x_j^{amount} for all j such that x_j^{amount} is in the set $D_{t_p,i}^{\text{freq}}$.

In addition, the authors of [39] mentioned that implementing a z-score would help indicate whether the amount is high or atypical for a certain customer as, for example, 500 euros are not of the same value for each customer. The z-score is calculated by Equations 3.6 and 3.7.

$$z_i = \frac{x_i^{amt} - \hat{\mu}_D}{\hat{\sigma}_D} \tag{3.6}$$

where $\hat{\mu}_D$ and $\hat{\sigma}_D$ are the sample mean and sample standard deviation, respectively,

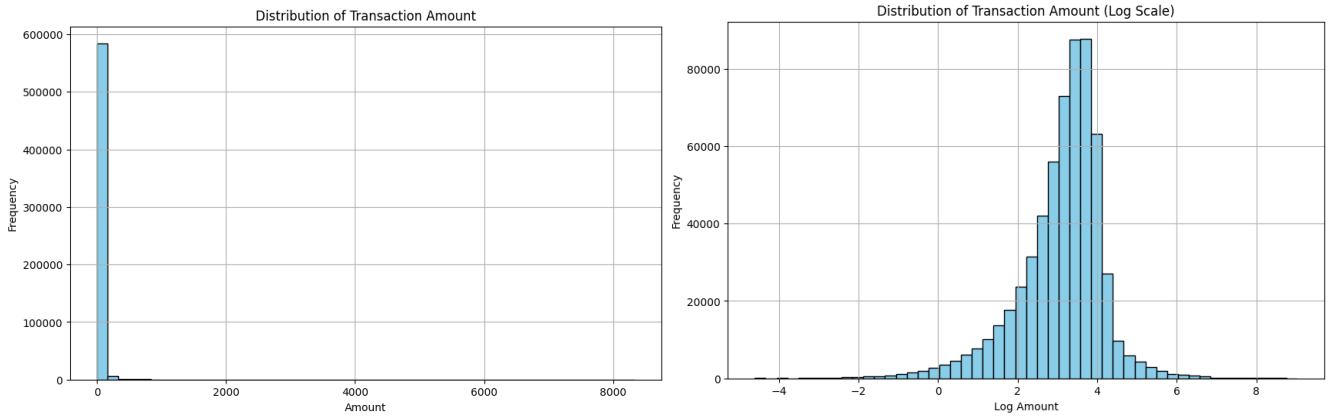
$$\hat{\mu}_D = \text{Mean}\left(D_{tp,i}^{freq}\right) \quad \text{and} \quad \hat{\sigma}_D = \text{Stdev}\left(D_{tp,i}^{freq}\right). \tag{3.7}$$

The calculation of the monetary value was calculated in the same aggregation as the frequency and monetary value.

Data pre-processing

After performing the feature engineering, the dataset consisted of 470 features, including the original features and the fraud column. The dataset underwent pre-processing, beginning with an exploratory data analysis (EDA) to understand its structure and characteristics. First, the amount value distribution was examined, illustrated in figure 3.1a. The figure shows that the dataset mainly consists of low transactions. To prevent the models from overfitting high-amount transactions, the transaction amounts were log-transformed to combat the skewness. The distribution of the transactions after the transformation is illustrated in figure 3.1b

As the dataset was synthetically made, it did not contain missing values, duplicates, or incorrect data that had to be cleaned or resolved. All the features were scaled to have a mean zero and unit variance to make it compatible for model training.



(a) Distribution of amount feature

(b) Distribution of amount feature (log scale)

Figure 3.1: Distribution of Transaction amount before and after log transformation

Feature Selection

Many features were made in the feature engineering phase, but to meet requirement eight, features were selected based on their correlation to the fraud feature so that the models could train on the accessible resources. All the features with a lower correlation than 0.2 to the fraud feature were removed from the data, resulting in 92 features with a higher correlation than 0.2.

3.1.5 Data splitting

As mentioned earlier, this research conducted three qualitative analyses. To conduct these analyses, suitable datasets had to be created. This section discusses the methods for splitting the data. First, a test and evaluation set were created, followed by a dataset specifically for the analyses.

Creating test and evaluation set

The original dataset D was divided into a training set D_{train}, an evaluation set D_{val} and a test set D_{test}. The purpose of the validation set was to determine a model's final performance and avoid data leakage. The test set was only used to determine the final performance. The evaluation set was used to compare models in the analyses. Using the same data for the analyses ensured that results could be properly evaluated without data leaks that could affect the performances.

The original dataset was divided using a stratified sampling method to split the test and evaluation sets to ensure a similar distribution of fraudulent and normal transactions. The proportion of the testing and evaluation set is 20% of the original set, while the other 60% is used as the main training set. This resulted in a 60:20:20 split. The data was split using the train_test_split function in the sklearn library.

Analysis 1: Performance of Isolation Forest versus Semi-Supervised Isolation Forest

The first analysis aimed to evaluate the performance of the isolation forest against the semi-supervised Isolation forest as a function of the percentage of fraudulent transactions in the training data. In doing so, the analysis helped understand how the performance of the two models proceeded when a certain percentage of fraud was available. The semi-supervised isolation forest required a lot of computational power, as it stores the labelled and unlabelled components. A training sample of only 20000 instances was used to meet requirement eight. A distribution of 75% labelled and 25% unlabelled transactions was assumed to recreate a scenario with labelled and unlabelled data.

The dataset D_{train} originally contains 1.2% fraudulent transactions, and for this research, 8 datasets were made with 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 and 1.1 % fraudulent transactions, respectively. These percentages were assumed to represent a real-world situation where only a very small number of fraudulent transactions are available when unsupervised models are employed to gain labels.

The datasets were obtained by randomly assigning 50% of the labels from D_{train} as unlabelled. An extra column was added with *Labelled* [0,1] to maintain the original labels. The dataset was split into four groups: fraudulent & unlabelled transactions, normal & unlabelled transactions, fraudulent & labelled transactions and normal & labelled transactions. Subsequently, a sample of 15000 normal & labelled transactions was created together with a sample of 5000 normal & unlabelled transactions. Then, for each percentage corresponding to the dataset, that percentage of fraudulent & labelled transactions, a fraudulent & unlabelled transactions were added. All the samples were added into one dataset representing the dataset with the corresponding percentage. This approach ensured that each dataset contained the same fraudulent transactions, but depending on the percentage, it contained more fraudulent transactions.

Analysis 2: Performance of XGBoost versus Semi-Supervised XGBoost

The second analysis aimed to evaluate both XGBoost and semi-supervised XGBoost. This analysis involved comparing XGBoost with semi-supervised XGBoost. In the semi-supervised XGBoost, the model is retrained, and the predicted labels are added to the training data. This process consumes time and computational resources to complete the algorithm. To meet requirement eight, analysis 2 also utilised smaller training data samples to train the models. To compare the performance of the models and account for computational limitations, the same datasets with the same percentages of fraudulent transactions were used in analysis 2.

Analysis 3: Performance of Semi-Supervised Models

The third analysis aimed to evaluate how well two semi-supervised methods performed based on the amount of known unlabelled data in the training set. This helped in understanding how the methods' performance was affected as the availability of labelled data increased or decreased.

For this analysis, the data was split into six parts based on the *Step* attribute, which indicates the time or step of each transaction. The first dataset had the first part of the six (1/6) as labelled data and the remaining parts as unlabelled data. The second dataset had two labelled parts (2/6) and so on, until five datasets were created with different percentages of labelled and unlabelled data based on the time attribute *Step*. This approach aimed to simulate a real situation where new labels became known over time.

A sample of 30000 rows was created for each subset to manage computational limitations. This ensured that the models could train within the computational resources and comply with requirement eight.

3.1.6 Model training & evaluation

After the datasets were created, the model training and evaluation phase was conducted. First, the evaluation metrics used for the evaluation phase are explained, followed by the training and evaluating phase. At last, the baseline model for this research is explained.

Evaluation metrics

Four evaluation metrics were chosen based on the findings of 2.2.2. The AOC-PR served as the main metric and measured the ability of a model to distinguish between positive and negative classes by summarising the trade-off between precision and recall across different threshold values. This metric was leading due to its ability to indicate performance in the unbalanced dataset, which was applicable to the datasets used and fraud detection in general.

Higher values of the AUC-PR indicate better performance. To support the AUC-PR, the precision and recall were also calculated. Precision is the ratio of true positive predictions to the total number of positive predictions made, while recall is the ratio of true positive predictions to the total number of actual positive instances in the dataset. This research did not consider the AOC-ROC and accuracy metrics due to their inability to provide insights on imbalanced datasets.

Evaluation approach

An iterative approach was obtained to evaluate the models in the three analyses. For the first two analyses, eight datasets were created. For the third analysis, five datasets were made. The evaluation dataset D_{val} was then used to assess the performance of the models. Since the isolation forest model is designed to identify anomalies in the training data and predict these labels, its evaluation should be based on the training data labels. However, for model comparison, the validation set was also used for label prediction. The performance of the label predictions on the training set is presented in Appendix D, while the predictions on the validation set were used for comparing the models.

The datasets were stored locally, and the training process of both models was performed in Python one by one. After each training phase, the models were evaluated on D_{val}. The AUC-PR, Precision and Recall metrics were calculated and saved so that they could be compared to each other and evaluate the model through an iterative approach. After training the models on all datasets, the results were gathered in a table with all metrics. Additionally, a figure was made that compares the AUC-PR of the models in the analysis and the AUC-PR of the baseline model.

After completing these evaluation results, the best-performing model on a certain dataset was selected for both analyses. That model was evaluated again on the final test set, which was held back from the beginning to prevent data leakage. The performance of that model by the AUC-PR curve and a confusion matrix. The performance results and comparison of the performance of the models supported the answering of the sub-questions in the research.

Baseline

Two baseline models were initially created to meet the third functional requirement, which should improve all the models in the analyses. If the models don't outperform the baseline model, they are not learning. The first model always selects the transaction as normal, and the second randomly selects a label for a transaction. However, the datasets used were imbalanced, making the first baseline model unusable. The random selection of labels was eventually used as only baseline model. The baseline performances were also added to the performance table to compare the models' performance.

3.2 Findings

This section shows the findings of the three analyses executed in the quantitative research, followed by the performance of each model on the test set. The observations in all analysis and test performances are provided. At last, a conclusion is given that concludes the findings and answers sub-question four.

3.2.1 Analysis 1

Analysis 1 compared the isolation forest with the semi-supervised isolation forest on the AUC-PR, Precision and Recall. Randomly assigned labels were used as a baseline. The datasets contained around 20000 instances, depending on the percentage of fraudulent labels. The results of all performance metrics on the evaluation set are

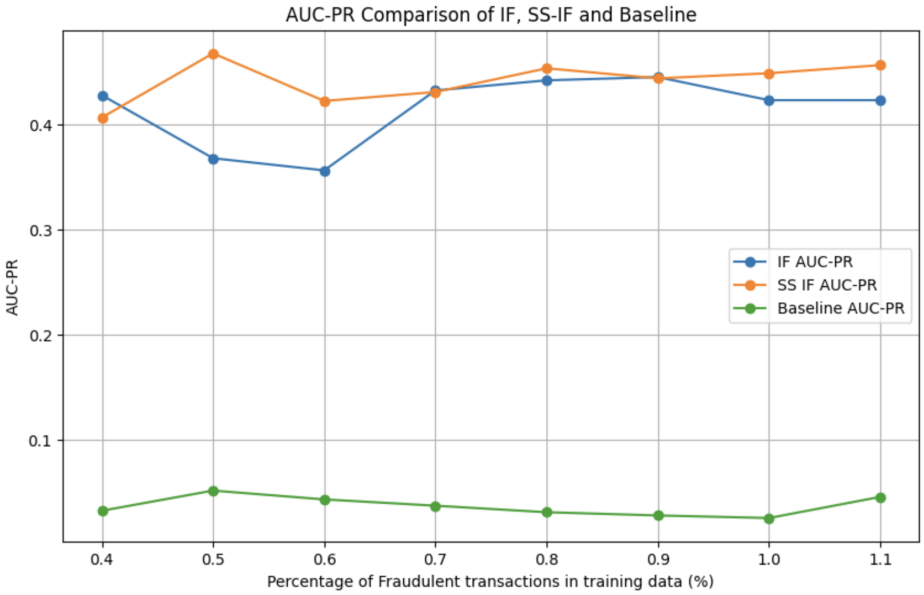


Figure 3.2: Comparison of unsupervised models on AUC-PR

shown in Table 3.1, and Figure 3.2 compares the AUC-PR of both models together with the baseline AUC-PR. The performance of the unsupervised isolation forest on the train set is shown in Appendix D. The parameters used for the models are shown in Appendix E. Table 3.1 showed that both the isolation forest and the semi-supervised model outperformed the base model, demonstrating that regardless of the fraud rate, the models learned the data and found the correct anomalies. Overall, the semi-supervised isolation forest outperformed the unsupervised isolation forest across all percentages of fraudulent transactions in the training data. The Semi-supervised isolation forest showed the highest AUC-PR. The semi-supervised isolation forest AUC-PR fluctuated above 0.4, while the unsupervised AUC-PR fluctuated around 0.4. The difference was not significant, but generally, it showed improvement.

As the percentage of fraudulent transactions increased, both models showed varying performances without a significant increase, indicating that there was no support provided for a correlation between the percentage of fraudulent transactions percentage and the performance. In table 3.1 was observed that the precision was high and recall was low for both models, which showed that it correctly identified normal transactions, but it also missed many true fraudulent labels. At 0.08% fraudulent transactions, the AUC-PR was highest for the unsupervised isolation forest, while for the semi-supervised isolation forest, the highest AUC-PR was at 0.05%.

The findings suggested that the semi-supervised model outperformed the unsupervised model, although not significantly, and that no correlation was found between the percentage of fraud in the training data and the performance of both models when evaluated on the evaluation set.

3.2.2 Analysis 2

Analysis 2 compared the XGBoost with the semi-supervised XGBoost on the AUC-PR, Precision and Recall. The same datasets were used as in analysis 1. The results of all performance metrics on the evaluating dataset are shown in Table 3.2. Figure 3.3 compared the AUC-PR of both models together with the baseline AUC-PR. The parameters used for the models are shown in Appendix E. Again, both models outperformed the baseline on all metrics, indicating that the model learned from all training datasets and could classify fraudulent transactions. Looking at figure 3.3, the observation was made that the semi-supervised XGBoost outperformed XGBoost slightly but consistently. The AUC-PR of the semi-supervised XGBoost showed the highest values because it fluctuated around 0.56, compared to the supervised XGBoost, which fluctuated around 0.54.

	Isolation Forest			Semi-supervised IF			Baseline		
	AUC-PR	PR	Recall	AUC-PR	PR	Recall	AUC-PR	PR	Recall
$D^{0.4\%}$	0.428	0.726	0.119	0.407	0.620	0.188	0.032	0.001	0.063
$D^{0.5\%}$	0.368	0.623	0.102	0.468	0.786	0.190	0.051	0.002	0.100
$D^{0.6\%}$	0.356	0.598	0.103	0.422	0.710	0.143	0.043	0.002	0.083
$D^{0.7\%}$	0.432	0.677	0.178	0.431	0.785	0.136	0.037	0.001	0.071
$D^{0.8\%}$	0.442	0.681	0.193	0.454	0.814	0.123	0.031	0.001	0.059
$D^{0.9\%}$	0.445	0.680	0.201	0.444	0.815	0.110	0.028	0.001	0.053
$D^{1.0\%}$	0.423	0.643	0.194	0.449	0.852	0.103	0.025	0.001	0.048
$D^{1.1\%}$	0.423	0.642	0.194	0.457	0.865	0.095	0.045	0.002	0.087

Table 3.1: Results of Analysis 1

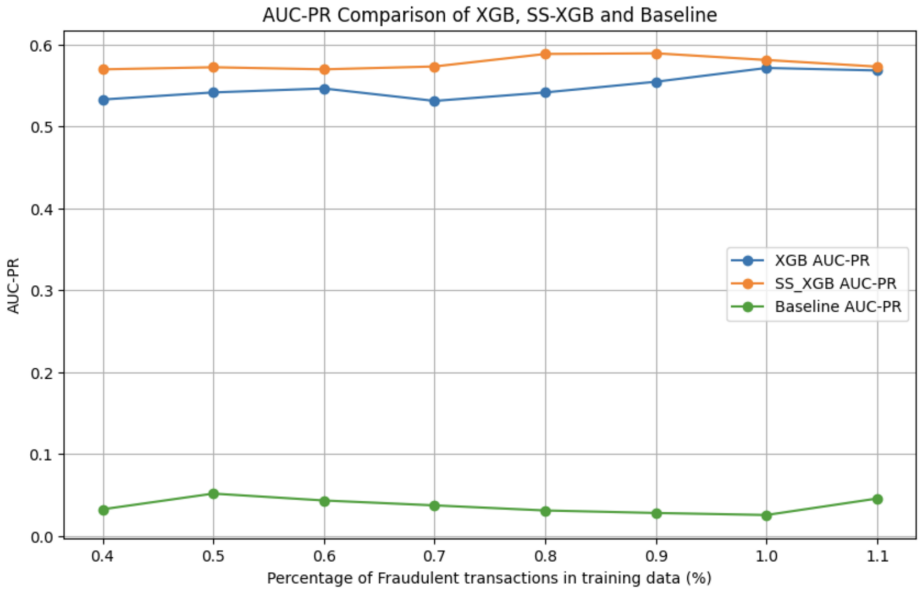


Figure 3.3: Comparison of supervised models on AUC-PR

The increase in fraudulent percentages did not show a significant increase or decrease in AUC-PR, indicating no proof of correlation between fraudulent transactions in the training data and the performance of AUC-PR in the evaluating data. The precision of the supervised XGBoost showed an increase when the number of fraud instances increased. This implied that the model got better at predicting normal instances. However, the recall stayed almost constant, explaining why the AUC-PR did not increase. In the semi-supervised XGBoost, the same was observed as the precision increased while the recall stayed constant. Yet, the recall is overall higher than the recall of the supervised XGBoost. This indicated that the semi-supervised XGBoost was better at predicting fraudulent transactions than the supervised XGBoost, supporting the overall observation that the model outperformed the supervised XGBoost model.

The findings implied that the semi-supervised XGBoost predicted the labels in the evaluation dataset more effectively than the supervised XGBoost. Although a slight increase in precision was seen, the overall performance did not increase significantly enough when the fraudulent transactions increased to distinguish a correlation between the two.

	XGBoost			Semi-supervised XGBoost			Baseline		
	AUC-PR	PR	Recall	AUC-PR	PR	Recall	AUC-PR	PR	Recall
$D^{0.4\%}$	0.533	0.365	0.685	0.569	0.338	0.749	0.032	0.001	0.063
$D^{0.5\%}$	0.541	0.375	0.669	0.572	0.357	0.733	0.051	0.002	0.100
$D^{0.6\%}$	0.546	0.407	0.659	0.569	0.388	0.711	0.043	0.002	0.083
$D^{0.7\%}$	0.531	0.404	0.641	0.573	0.408	0.719	0.037	0.001	0.071
$D^{0.8\%}$	0.541	0.423	0.655	0.588	0.427	0.726	0.031	0.001	0.059
$D^{0.9\%}$	0.554	0.430	0.664	0.589	0.424	0.731	0.028	0.001	0.053
$D^{1.0\%}$	0.571	0.474	0.652	0.581	0.423	0.756	0.025	0.001	0.048
$D^{1.1\%}$	0.568	0.470	0.670	0.573	0.436	0.731	0.045	0.002	0.087

Table 3.2: Results of Analysis 2

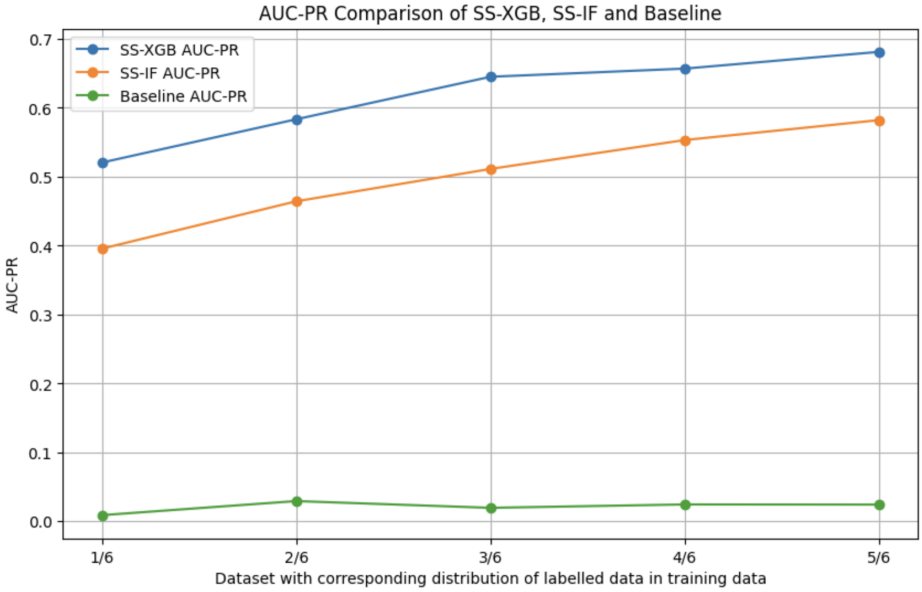


Figure 3.4: Comparison of Semi-supervised models on AUC-PR

3.2.3 Analysis 3

Analysis 3 compared both semi-supervised models with each other when different percentages of known labels were available in the dataset. The main dataset was split into 6 parts, and five subsets were made with 1/6, 2/6, 3/6, 4/6 and 5/6 parts of the dataset labelled respectively. Each dataset was sampled to 30.000 instances due to computational limitations. The models were evaluated on the same evaluation dataset. The performance metrics are shown in table 3.3, and the comparison of the AUC-PR with the baseline is illustrated in figure 3.4. Both models outperformed the baseline on all metrics.

First, an increase in AUC-PR was seen in figure 3.4, while the semi-supervised XGBoost showed higher performance on the AUC-PR than the semi-supervised isolation forest. This indicated a correlation between the increase of known labels and the performance of both models. The highest AUC-PR is 0.681, obtained by the semi-supervised XGBoost, and the highest AUC-PR for the semi-supervised isolation forest is 0.582.

From table 3.3, a higher recall of the semi-supervised XGBoost was seen, which explains the better performance of the AUC-PR. This indicated that the model predicted more fraudulent transactions correctly.

The findings suggested that the semi-supervised XGBoost outperformed the semi-supervised isolation forest and that there was a correlation between the known labels in the training data and the performance of the semi-supervised models.

	Semi-supervised XGBoost			Semi-supervised IF			Baseline		
	AUC-PR	PR	Recall	AUC-PR	PR	Recall	AUC-PR	PR	Recall
$D^{1/6}$	0.521	0.341	0.836	0.396	0.419	0.433	0.009	0.003	0.002
$D^{2/6}$	0.583	0.351	0.859	0.464	0.477	0.488	0.029	0.023	0.024
$D^{3/6}$	0.645	0.305	0.899	0.512	0.512	0.526	0.020	0.014	0.013
$D^{4/6}$	0.657	0.283	0.922	0.553	0.552	0.566	0.025	0.018	0.019
$D^{5/6}$	0.681	0.286	0.922	0.582	0.580	0.594	0.024	0.017	0.019

Table 3.3: Results of Analysis 3

3.2.4 Performances on test data

Each of the best-performing models in the first and second analyses were tested on the test dataset. The AUC-PR curve is printed together with the confusion matrix to observe the final performance of the models. The models' performances are revealed in the same order as they were trained in analysis 1 and 2. The isolation forest will be followed by the semi-supervised isolation forest, XGBoost, and semi-supervised XGBoost.

Isolation Forest

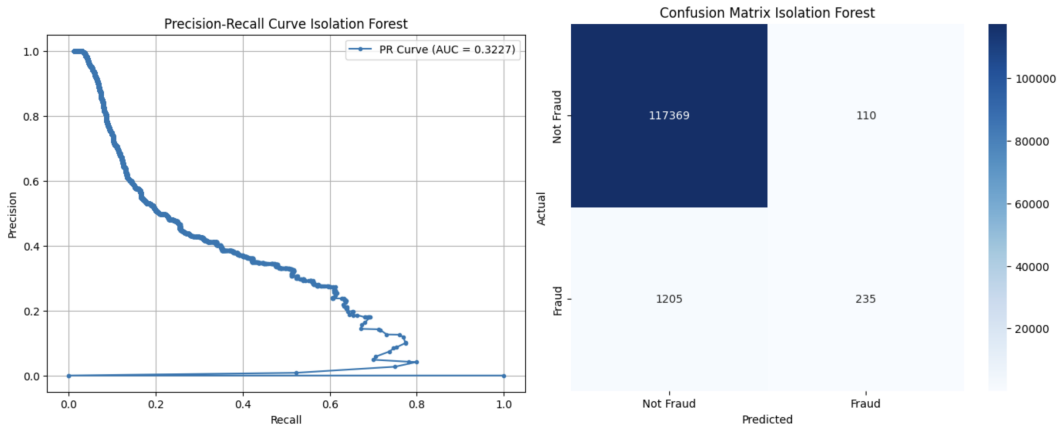
The isolation forest model scored the highest AUC-PR on the training dataset with 0.09% fraudulent transactions. The AUC-PR on the evaluation set was higher (0.445) than the test set (0.323), looking at the AUC-PR shown in the PR Curve in figure 3.5a. The isolation forest model showed an unstable PR curve, with a steep downward slope moving with a strongly decreasing precision as the recall increases. At high recall values, the precision is very noisy. This indicated that the model struggled to correctly identify fraud labels without misclassifying a significant number of normal (Not Fraud) labels. The confusion matrix in table 3.5b supported that observation because only 235 fraud labels were correctly predicted, and 1205 labels were missed. The model predicted many normal labels correctly and only misclassified 110 normal instances. However, that also showed that approximately $1/3th$ of the fraud labels were incorrectly predicted.

Semi-supervised Isolation Forest

The performance of the semi-supervised isolation forest also decreased when it was tested on the test dataset. Initially, the model scored an AUC-PR of 0.468 on the training dataset with 0.05% fraudulent transactions, while it had an AUC-PR of 0.38 on the test dataset. From the PR Curve in figure 3.8a, it was observed that the curve is significantly more stable than the isolation forest. It showed the same decrease in precision when the recall increased, but at higher recall values, the precision was more stable. This effect was also seen in the confusion matrix in figure 3.8b. From the 1440 fraud cases in the test dataset, 512 were correctly identified, and 928 were not identified. This was almost a double increase in true positives. However, the false positives also increased, and now approximately half of the fraud predictions made by the model were incorrect. Overall, the semi-supervised isolation forest improved performance compared to the isolation forest as the AUC-PR improved. Furthermore, the PR Curve was more stable, and more fraud labels were correctly predicted.

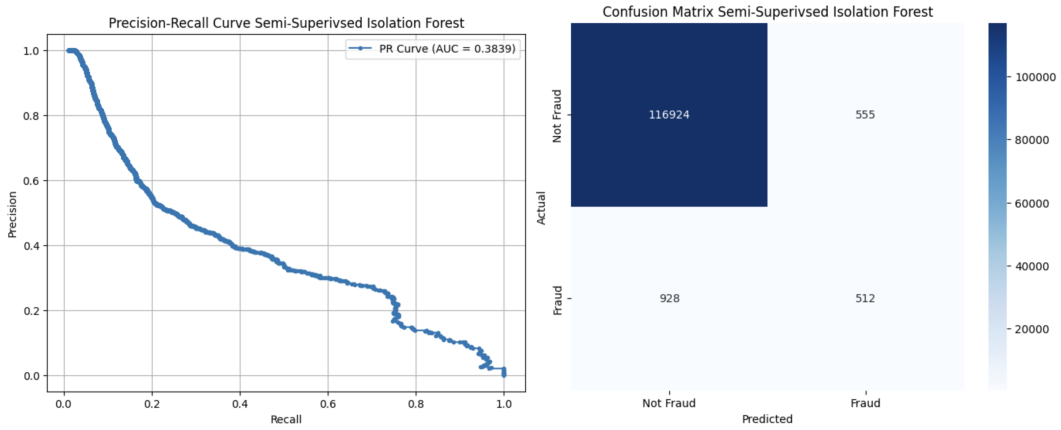
XGBoost

The best performing XGBoost model was with 0.1% of fraudulent transactions. The PR curve and confusion matrix are illustrated in figure 3.7a and 3.7b. The model received a 0.587 AUC-PR on the test set, slightly improving the initial 0.571 AUC-PR on the evaluation set. Looking at the XGBoost PR curve, the overall downward slope indicated that when more actual fraud labels were trying to be predicted, the precision dropped, meaning more normal transactions were mistakenly identified as fraud. However, comparing the model with both unsupervised



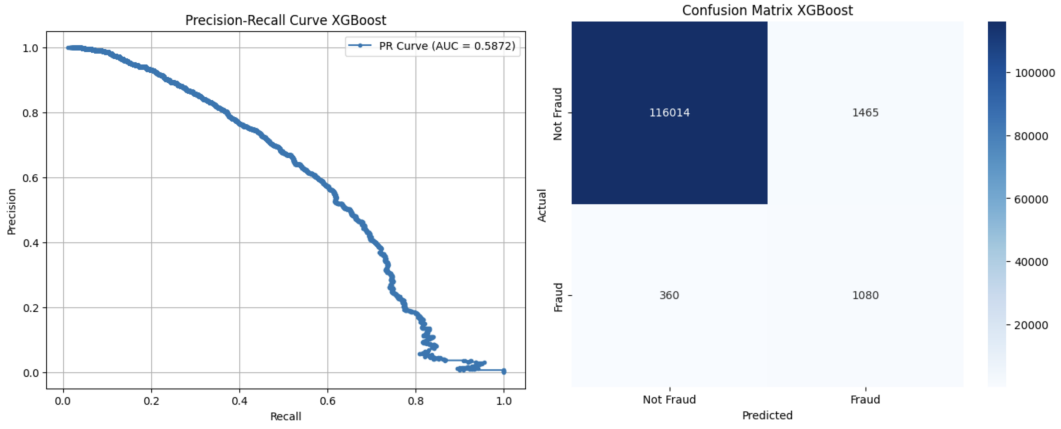
(a) PR Curve Isolation Forest (b) Confusion Matrix Isolation Forest

Figure 3.5: Results of Isolation Forest on test dataset



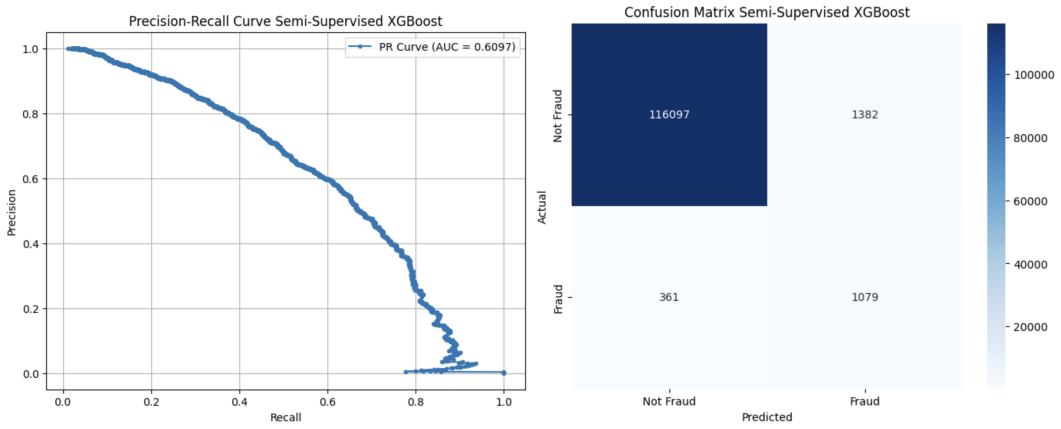
(a) PR Curve Semi-Supervised Isolation Forest (b) Conf Matrix Semi-Supervised Isolation Forest

Figure 3.6: Results of Semi-Supervised Isolation Forest on test dataset



(a) PR Curve XGBoost (b) Confusion Matrix XGBoost

Figure 3.7: Results of XGBoost on test dataset



(a) PR Curve Semi-Supervised XGBoost (b) Confusion Matrix Semi-Supervised XGBoost

Figure 3.8: Results of Semi-Supervised XGBoost on test dataset

and semi-superivsed isolation forests, the precision did not decrease as much when the recall increased, and there was slightly more noise at a high recall compared to the other models where the noise was more at a low recall. This suggested that the model was better at predicting fraud and slightly worse at predicting normal labels. This result was also visible in the confusion matrix (3.8b). Here, the number of true positives was higher, with 1080 correct instances and 360 missed instances. The false positives were higher with 1465, supporting the effect of the noise in the PR curve at low recall. Although there are more false positives, the model performed significantly better than both isolation forest models.

Semi-supervised XGBoost

The semi-supervised XGB had the highest AUC-PR on the evaluation dataset with 0.09% fraudulent labels in the test dataset. The AUC-PR on the test dataset was 0.610, which was higher than the AUC-PR on the evaluation set (0.589) but also improved the AUC-PR of the supervised XGBoost (0.587). Observing the PR Curve in figure 3.8a, a similar curve was seen. Identical noise was detected at low recall, followed by a similar precision decrease when the recall increased. The improvement of the AUC-PR was explained by the confusion matrix in figure 3.8b. The confusion matrix showed a similar number of false negatives and true positives as before (figure 3.7b), but there was a decrease in the number of false negatives. This indicated an improvement in predicting normal labels, and it predicted fewer fraudulent labels in general while still having the same number of correct fraud labels. Concluding, the semi-supervised XGBoost outperformed the supervised XGBoost by correctly predicting more normal labels.

3.2.5 Conclusion

The findings of the three analyses showed that semi-supervised architectures generally perform slightly better than both supervised and unsupervised architectures in fraud detection, answering sub-question four. Analysis 1 showed that the semi-supervised isolation forest consistently, but not significantly, outperformed the unsupervised isolation forest with a higher AUC-PR. The semi-supervised isolation forest correctly predicted more fraudulent labels and had more stable values in the precision-recall curve at high recall. The findings did not indicate a correlation between the number of fraudulent transactions and the performance of either model. Analysis 2 showed that the semi-supervised XGBoost performed slightly better than the supervised XGBoost but consistently. The semi-supervised model had a higher AUC-PR and better recall, suggesting it more effectively identified fraudulent transactions. A correlation between the percentage of known fraudulent transactions was observed, with the semi-supervised XGBoost performing better than the semi-supervised isolation forest.

Chapter 4

Discussion

The aim of this research was to improve the performance of AML (anti-money laundering) systems using semi-supervised learning for fraud detection, to investigate how semi-supervised learning techniques could improve fraud detection, and to define which metrics determine the quality of performance. Based on that objective, qualitative and quantitative research was used, with the qualitative research mainly providing insights into the challenges, quality metrics and semi-supervised taxonomy. The quantitative research compared semi-supervised models with super- and unsupervised models to compare how they can improve fraud detection. The results showed that the main challenges in using machine learning in fraud detection are unsupervised anomaly detection models that do not provide sufficient labels, supervised models that do not have sufficient labels and unsupervised models that underperform, leaving the analyst with redundant work in assessing the labels. The metrics that define performance are AUC-PR, Precision and Recall, while model stability and explainability define further quality. Semi-supervised anomaly detection and wrapper methods proved to be the most promising techniques for fraud detection. Finally, the results of the quantitative study showed that semi-supervised models outperformed super- and unsupervised models slightly but consistently, with a correlation between the percentage of known labels in the training dataset and no correlation between the percentage of fraudulent labels in the training dataset.

The semi-supervised isolation forest correctly predicted more fraudulent labels than the unsupervised isolation forest, which could contribute to the challenge that unsupervised models do not produce enough labels. In contrast, the semi-supervised XGBoost slightly outperformed the supervised XGBoost but not significantly enough to argue that the model can tackle the challenge of supervised models that do not have sufficient labels. This implication is supported by the positive correlation found between the known labels and the performance because that suggests that in situations where little labelled data is available, semi-supervised models do not improve in performance, while with a higher percentage of labelled data, the models may improve in performance. One could argue that this effect is not beneficial for fraud detection because semi-supervised models are meant to improve situations with insufficiently labelled data.

The analyses showed a correlation between the known labels in the dataset and the performance of semi-supervised models. This finding is in line with the expectations from research [28]. Conversely, the analyses indicate no correlation between the number of fraudulent transactions in the dataset, while research [37] did indicate such correlation. Further, the performances of the models are difficult to compare with literature from the literature review because most research applied the Detection Rate [15, 17, 16], False Positive Rate (FPR) [15, 17, 16], Accuracy [15, 12], F1 Score [7, 6, 20], and Area Under The Receiver Operator Curve (AUC-ROC) [6, 20, 21, 13, 7]. The findings of this research indicate that these metrics are invalid for use due to the imbalanced dataset, which implies that the findings of this research are difficult to compare to other research. Next to that, not all possible architectures were researched. There are many different wrapper methods or semi-supervised anomaly detection

applications that may outperform the researched architectures.

Limitations have also affected the results of this study because this research leveraged small training data due to computational limitations, which could affect performance and not reflect on real-world situations. Next to that, feature engineering was applied, but only statistical features were created. This research did not leverage any domain knowledge to create features, and a simple feature selection based on the correlation was applied. Moreover, due to time limitations, hyperparameter tuning was not performed very thoroughly on all models. Thorough hyperparameter tuning could improve performance. In addition, model stability and explainability are not considered when comparing the evolved models as quality metrics.

Future research can extend this research in several ways. First, the models can be trained on bigger training data when more computational power is available. Second, more feature engineering on the dataset can be applied. Third, hyperparameter tuning can be performed better to utilize the models more. Finally, the model can be evaluated based on model stability and explainability to provide another dimension to the quality of semi-supervised learning for fraud detection.

Nevertheless, the results of this research contribute insights to the field of Machine Learning and Applied AI, especially in the context of fraud detection. Essential stakeholders, financial institutions, and AML software providers could use these results to gain insights into using semi-supervised learning. They could do further research based on this methodology or use the findings to decide whether it is interesting to implement. The field of Machine Learning and Applied AI could use these findings to look at the effects of semi-supervised learning generally or use the methodology applied to gather insights. The results also indirectly contribute to society as the improvement of fraud detection contributes to fighting money laundering, which has an impact on the economics of the world [29].

Chapter 5

Conclusion

This thesis explored how semi-supervised learning techniques could improve fraud detection and what metrics determine the performance quality. It used qualitative and quantitative research to answer that question. Overall, the findings suggest that the performance of fraud detection can be improved using semi-supervised anomaly detection to predict more fraudulent labels and that the AUC-PR, Precision, and Recall determine the performance quality.

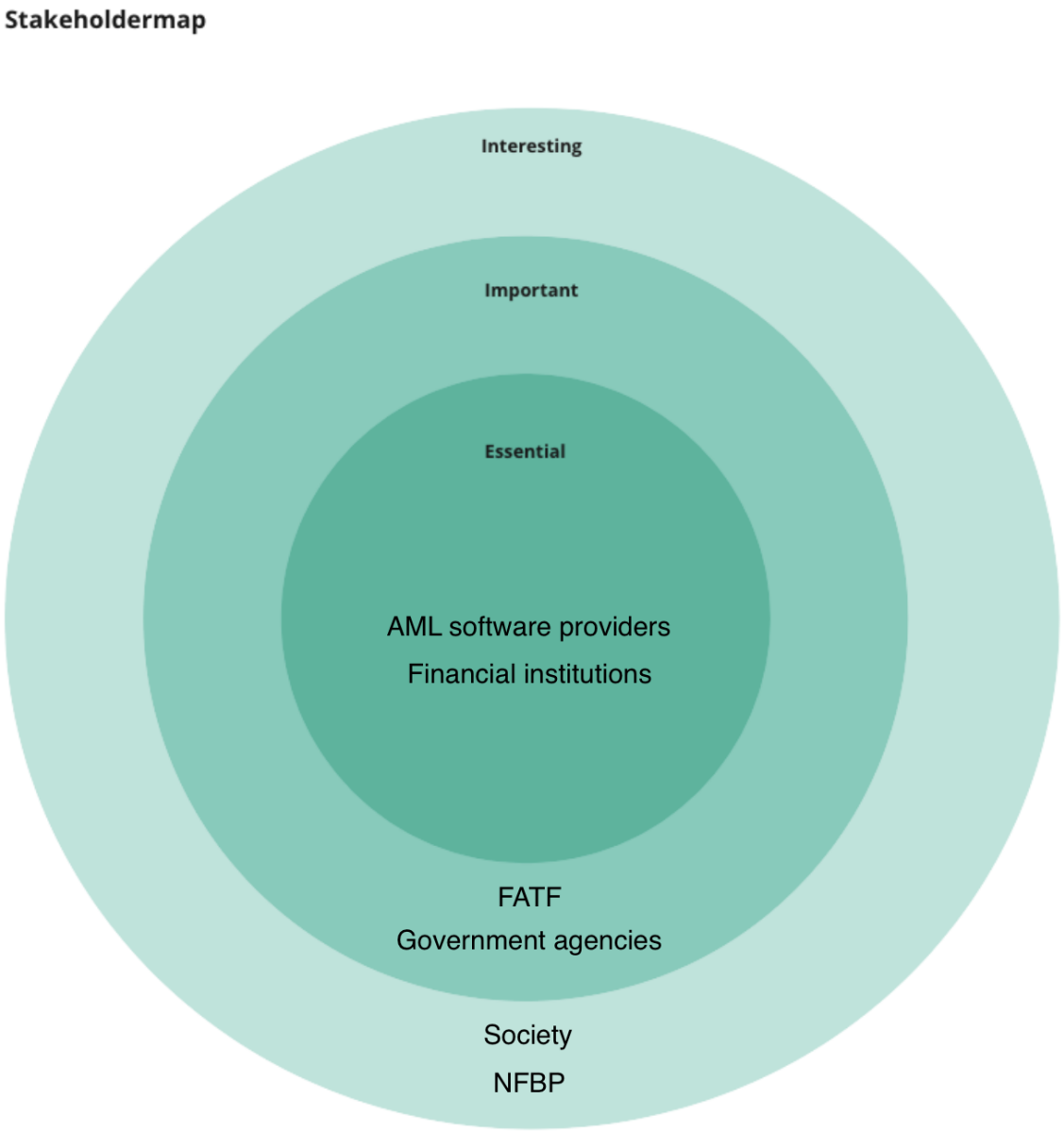
The results of the qualitative research revealed that financial institutions and AML software providers are the main stakeholders involved in this research because they directly implement fraud detection in AML systems in their business. In addition, the interviews highlighted two key challenges in fraud detection in AML systems. First, unsupervised models are used to detect 'unknown unknowns', which can lead to new training labels for supervised models, but often the unsupervised models provide insufficient labels. Secondly, the review of newly detected labels by an analyst can result in redundant work if the predicted labels are not fraudulent. The interviews also showed that the AUC-PR, Precision and Recall are metrics that determine the performance of the model, and model stability is another metric to determine the quality of fraud detection. The desk research showed that two applications of inductive learning methods, wrapper methods and semi-supervised anomaly detection, were applicable for fraud detection, while transductive learning methods were considered unusable for this research.

In three analyses, the quantitative research compared semi-supervised models with supervised and unsupervised models. Specifically, an isolation forest was used as an unsupervised model, XGBoost was used as a supervised model, a semi-supervised XGBoost was used as a wrapper method, and a semi-supervised isolation forest was used for anomaly detection. The results demonstrated that semi-supervised models generally outperform both supervised and unsupervised models slightly, with the semi-supervised XGBoost having the highest performance. There was no correlation observed between the performance of all models and the percentage of fraudulent labels in the training data, while there was a correlation observed between the percentage of known labels in the training and the performance of the semi-supervised learning.

In conclusion, this research contributes insights into the application of semi-supervised learning in fraud detection, suggesting that these techniques slightly but consistently improve the detection of fraud in AML systems. Future research could further explore the use of bigger training data, more feature engineering, hyperparameter tuning, and more quality assessment by model stability and explainability in real-world situations.

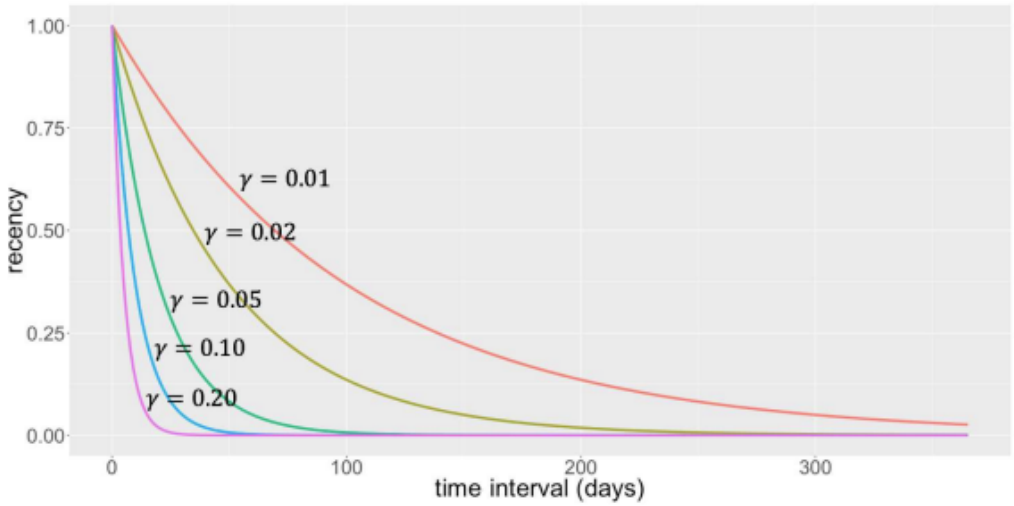
Appendix A

Stakeholder Map



Appendix B

γ Parameter



Appendix C

Interview Reports

C.1 Interview 1

- Participant: Data Scientist at an international Dutch bank
- Date: 24th of March, 2024

What kind of architecture do you use for fraud detection or related systems?

I find that XGBoost works very well; I often refer to it as a workhorse. For anomaly detection, I always choose isolation forest, which is also very reliable. In my opinion, these models are effective for research purposes.

What are the current challenges in supervised & unsupervised techniques in fraud detection?

AML is constantly evolving, and new challenges arise regularly. Payment service providers and bitcoins, for example, are relatively new. In these cases, we typically start with a new model without labels, requiring us to use unsupervised anomaly detection methods. In anomaly detection, it's crucial to select risk factors that highlight laundering patterns as outliers in feature values.

However, with many features, these can correlate heavily, making it difficult for an anomaly model to weigh their importance or to identify specific similar transactions. For that, a supervised model is needed, but labels are often missing. After deploying an anomaly model and receiving initial features, there is a phase with a small number of labels, typically between 20 and 200, which are too few for supervised learning.

What is your perception of semi-supervised learning techniques in fraud detection, do you see opportunities or challenges, and why?

Semi-supervised learning could bridge the gap in the challenges mentioned. In cases like terrorism prevention, where labels are scarce, semi-supervised learning can help integrate new labels gathered from unsupervised models to support supervised models.

However, many semi-supervised models are not suitable for our use case. For us, semi-supervised anomaly detection would be particularly interesting for money laundering. It would be valuable to compare the performance of semi-supervised models against supervised and unsupervised models based on the percentage of known labels, to see how they perform at different levels of label availability.

What is your assessment of the quality of AML systems?

It is crucial that multiple models trained on the same data exhibit consistent performance, indicating model stability. Additionally, the model must be explainable, which can be achieved using SHAP values.

What metrics do you think are important for evaluating fraud detection performance, and why?

I find that scientific papers often report very high performances, suggesting the use of filters to maintain high performance and possibly incorrect metrics. The AUC-ROC is often used but is less relevant for imbalanced datasets. The AUC-PR is a better metric, providing a good overall performance measure. More importantly, the 95

Using the F1 score is a major mistake because it assumes a 50% probability cutoff, which is not suitable for our purposes as our cutoff is closer to 5%. Hence, we prefer the AUC-PR for evaluating performance.

Important note: *The interviews covered the topic of semi-supervised learning in AML systems and fraud detection. Although most of the questions were asked, the answers above are not word-for-word the exact answers to the questions. For readability, these are the general answers to the questions*

C.2 Interview 2

- Participant: Data Scientist at an international Dutch bank
- Date: 26th of March, 2024

What kind of architecture do you use for fraud detection or related systems?

If there are enough labels, we typically train a supervised model, depending on the number of transactions involved. Even if we don't have all the labels, we aim to identify known cases. We often use well-known supervised models such as XGBoost or Random Forest. Additionally, we employ isolation forest for unsupervised learning.

What are the current challenges in supervised & unsupervised techniques in fraud detection?

Supervised models work well when there are sufficient examples and clear targets. However, when we are uncertain about what we are looking for, or when dealing with 'unknown unknowns', we use unsupervised models to detect anomalies. These anomalies are then included in the next training cycle of the supervised model.

The disadvantage of supervised learning is that it only identifies patterns it has been trained on. Unsupervised learning can detect a variety of patterns, including normal ones, which can result in noise and instability if the model is not well-tuned.

What is your perception of semi-supervised learning techniques in fraud detection, do you see opportunities or challenges, and why?

Semi-supervised learning is efficient as it allows a limited number of cases to be presented to analysts, saving time by focusing on valuable cases in my idea.

What is your assessment of the quality of AML systems?

The model must be explainable and scientifically justifiable. Next to that is should be stable. If using different seeds for the same model yields significantly different outcomes, it indicates model instability.

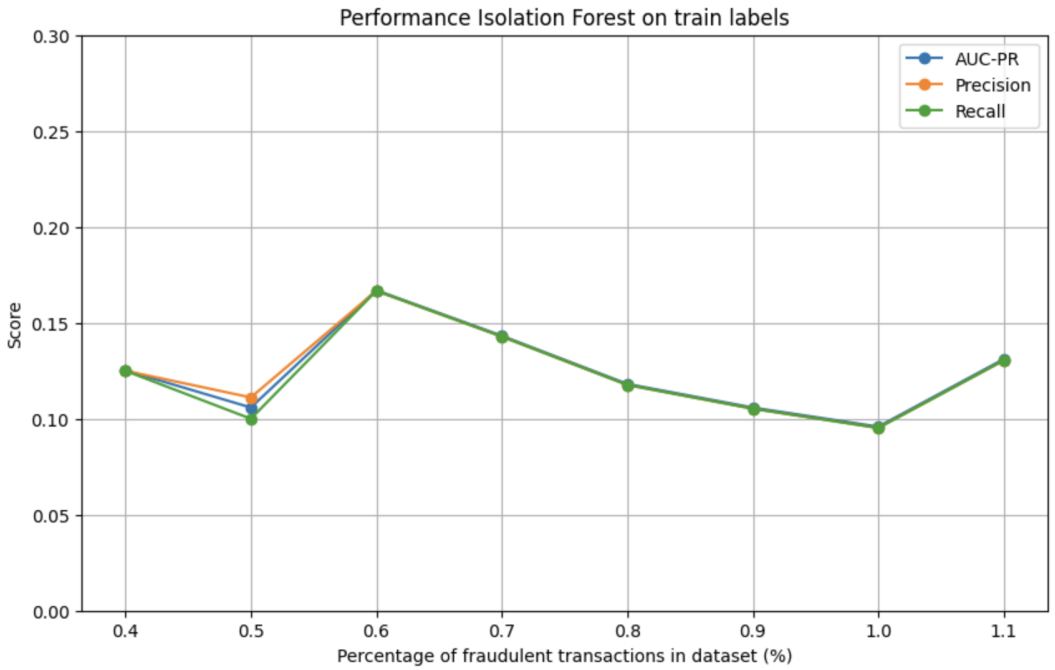
What metrics do you think are important for evaluating fraud detection performance, and why?

Model stability is a metric. It is measured by retraining the model with different seeds and comparing the results. The acceptable overlap between results depends on the business case, typically around 80%. This comparison is preferably done seasonally to account for seasonal effects.

Important note: *The interviews covered the topic of semi-supervised learning in AML systems and fraud detection. Although most of the questions were asked, the answers above are not word-for-word the exact answers to the questions. For readability, these are the general answers to the questions*

Appendix D

Performance of Isolation Forest on Training data



	Isolation Forest			Semi-supervised IF			Baseline		
	AUC-PR	PR	Recall	AUC-PR	PR	Recall	AUC-PR	PR	Recall
$D^{0.4\%}$	0.125	0.125	0.125	0.407	0.620	0.188	0.032	0.001	0.063
$D^{0.5\%}$	0.106	0.111	0.100	0.468	0.786	0.190	0.051	0.002	0.100
$D^{0.6\%}$	0.167	0.167	0.167	0.422	0.710	0.143	0.043	0.002	0.083
$D^{0.7\%}$	0.143	0.143	0.143	0.431	0.785	0.136	0.037	0.001	0.071
$D^{0.8\%}$	0.118	0.118	0.118	0.454	0.814	0.123	0.031	0.001	0.059
$D^{0.9\%}$	0.106	0.105	0.105	0.444	0.815	0.110	0.028	0.001	0.053
$D^{1.0\%}$	0.096	0.095	0.095	0.449	0.852	0.103	0.025	0.001	0.048
$D^{1.1\%}$	0.131	0.130	0.130	0.457	0.865	0.095	0.045	0.002	0.087

Table D.1: Performance of Isolation Forest on training data

Appendix E

Parameters of models

Parameter	Value
objective	binary:logistic
evaluation metric	aucpr
scale_pos_weight	[0.05, 0.12] (depending on the training dataset)
max Depth	6
learning Rate (eta)	0.1
seed	42

Table E.1: XGBoost & Semi-supervised XGBoost Parameters

Parameter	Value
n_estimators	100
contamination	[0.05, 0.12] (depending on the training dataset)
max_samples	auto
max_features	1.0
n_jobs	-1
random_seed	42

Table E.2: Isolation Forest Parameters

Parameter	Value
n_estimators	100
contamination	[0.05, 0.12] (depending on the training dataset)
ntrees	30
max_samples	auto
max_features	1.0
n_jobs	-1
random_seed	42

Table E.3: Semi-supervised Isolation Forest Parameters

Bibliography

[1] D. Cox, *Handbook of anti-money laundering*. John Wiley & Sons, 2014.

[2] I. Ofoeda, E. K. Agbloyor, J. Y. Abor, and K. A. Osei, "Anti-money laundering regulations and financial sector development," *International Journal of Finance & Economics*, vol. 27, no. 4, pp. 4085–4104, 2022.

[3] B. A. Raweh, C. Erbao, and F. Shihadeh, "Review the literature and theories on anti-money laundering," *Asian Development Policy Review*, vol. 5, p. 140–147, Jul. 2017.

[4] E. Altman, B. Egressy, J. Blanuvsa, and K. Atasu, "Realistic synthetic financial transactions for anti-money laundering models," *ArXiv*, vol. abs/2306.16424, 2023.

[5] A. N. Bakry, A. S. Alsharkawy, M. S. Farag, and K. R. Raslan, "Combating financial crimes with unsupervised learning techniques: Clustering and dimensionality reduction for anti-money laundering," *Al-Azhar Bulletin of Science*, vol. 35, Apr. 2024.

[6] Z. Chen, W. M. Soliman, A. Nazir, and M. Shorfuzzaman, "Variational autoencoders and wasserstein generative adversarial networks for improving the anti-money laundering process," *IEEE Access*, vol. 9, pp. 83762–83785, 2021.

[7] M. R. Karim, F. Hermsen, S. A. Chala, P. de Perthuis, and A. Mandal, "Catch me if you can: Semi-supervised graph learning for spotting money laundering," 2023. Copyright - © 2023. This work is published under http://creativecommons.org/licenses/by-nc-nd/4.0/ (the "License"). Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License; Last updated - 2023-03-09.

[8] Ministerie van Financiën Ministerie van Justitie en Veiligheid, "Algemene leidraad wet ter voorkoming van witwassen en financieren van terrorisme (wwft)," 2020. https://www.rijksoverheid.nl/documenten/richtlijnen/2020/07/21/algemene-leidraad-wet-ter-voorkoming-va

[9] K. Sullivan, *Anti-Money Laundering in a Nutshell : Awareness and Compliance for Financial Personnel and Business Managers*. Berkeley, CA: Apress L. P., 2nd ed. ed., 2024.

[10] R. I. T. Jensen and A. Iosifidis, "Fighting money laundering with statistics and machine learning," *IEEE Access*, vol. 11, pp. 8889–8903, 2023.

[11] Z. Chen, V. K. Le Dinh, E. N. Teoh, A. Nazir, K. K. Ettikan, and S. L. Kim, "Machine learning techniques for anti-money laundering (aml) solutions in suspicious transaction detection: a review," *Knowledge and Information Systems*, vol. 57, pp. 245–285, 11 2018. Copyright - Knowledge and Information Systems is a copyright of Springer, (2018). All Rights Reserved; Last updated - 2023-11-29.

[12] A. Alsuwailem and A. Saudagar, "Anti-money laundering systems: a systematic literature review," *Journal of Money Laundering Control*, vol. ahead-of-print, 05 2020.

[13] D. Savage, Q. Wang, P. Chou, X. Zhang, and X. Yu, "Detection of money laundering groups using supervised learning in networks," 2016.

[14] Y. Zhang and P. Trubey, "Machine learning and sampling scheme: An empirical study of money laundering detection," *Computational Economics*, vol. 54, 10 2019.

[15] Y. Tingting and L. Keyan, "An improved support-vector network model for anti-money laundering," in *Management of e-Commerce and e-Government, International Conference on*, (Los Alamitos, CA, USA), pp. 193–196, IEEE Computer Society, nov 2011.

[16] J. Tang and J. Yin, "Developing an intelligent data discriminating system of anti-money laundering based on svm," in *2005 International Conference on Machine Learning and Cybernetics*, vol. 6, pp. 3453–3457 Vol. 6, 2005.

[17] L.-T. Lv, N. Ji, and J.-L. Zhang, "A rbf neural network model for anti-money laundering," *2008 International Conference on Wavelet Analysis and Pattern Recognition*, vol. 1, pp. 209–215, 2008.

[18] D. K. Cao and P. Do, "Applying data mining in money laundering detection for the vietnamese banking industry," in *Intelligent Information and Database Systems* (J.-S. Pan, S.-M. Chen, and N. T. Nguyen, eds.), (Berlin, Heidelberg), pp. 207–216, Springer Berlin Heidelberg, 2012.

[19] Y. Yang, X. Guan, and J. You, "Clope: a fast and effective clustering algorithm for transactional data," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, (New York, NY, USA), p. 682–687, Association for Computing Machinery, 2002.

[20] X. Li, S. Liu, Z. Li, X. Han, C. Shi, B. Hooi, H. Huang, and X. Cheng, "Flowscope: Spotting money laundering based on graphs.," 2020.

[21] F. Johannessen and M. Jullum, "Finding money launderers using heterogeneous graph neural networks," 2023.

[22] M. Weber, J. Chen, T. Suzumura, A. Pareja, T. Ma, H. Kanezashi, T. Kaler, C. E. Leiserson, and T. B. Schardl, "Scalable graph learning for anti-money laundering: A first look," 2018.

[23] E. A. Lopez-Rojas and S. Axelsson, "Banksim: A bank payment simulation for fraud detection research," 09 2014.

[24] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (New York, NY, USA), p. 785–794, Association for Computing Machinery, 2016.

[25] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, 2008.

[26] N. Fazakis, G. Kostopoulos, S. Karlos, S. Kotsiantis, and K. Sgarbas, "Self-trained extreme gradient boosting trees," in *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pp. 1–6, 2019.

[27] P.-F. Marteau, S. Soheily-Khah, and N. Béchet, "Hybrid isolation forest - application to intrusion detection," *ArXiv*, vol. abs/1705.03800, 2017.

[28] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, p. 373–440, 2020.

[29] N. Hendriyetty and B. S. Grewal, "Macroeconomics of money laundering: effects and measurements," *Journal of Financial Crime*, vol. 24, no. 1, pp. 65–81, 2017.

[30] N. Omar, Z. Johari, and I. Mohamed, "A review on the role of designated non-financial business and professions (dnfbps) as preventive measures in mitigating money laundering," *International Scientific Researches Journal*, vol. 72, no. 7, pp. 93–105, 2016.

[31] K.-K. R. Choo, "Designated non-financial businesses and professionals: A review and analysis of recent financial action task force on money laundering mutual evaluation reports," *Security Journal*, vol. 27, 02 2014.

[32] V. Vladimir, "Transductive Inference and Semi-Supervised Learning," in *Semi-Supervised Learning*, ch. 24, p. 452–472, The MIT Press, 09 2006.

[33] M. E. Villa-Pérez, M. Álvarez Carmona, O. Loyola-González, M. A. Medina-Pérez, J. C. Velazco-Rossell, and K.-K. R. Choo, "Semi-supervised anomaly detection algorithms: A comparative summary and future research directions," *Knowledge-based systems*, vol. 218, pp. 106878–, 2021.

[34] J. Yoon, K. Sohn, C.-L. Li, S. O. Arik, and T. Pfister, "SPADE: Semi-supervised anomaly detection under distribution mismatch," *Transactions on Machine Learning Research*, 2023. Featured Certification.

[35] H. Xu, Y. Wang, G. Pang, S. Jian, N. Liu, and Y. Wang, "Rosas: Deep semi-supervised anomaly detection with contamination-resilient continuous supervision," *Information Processing Management*, vol. 60, no. 5, p. 103459, 2023.

[36] L. Stradiotti, L. Perini, and J. Davis, *Semi-Supervised Isolation Forest for Anomaly Detection*, pp. 670–678.

[37] J. Yoon, K. Sohn, C.-L. Li, S. Ö. Arik, C.-Y. Lee, and T. Pfister, "Self-supervise, refine, repeat: Improving unsupervised anomaly detection," *Transactions on Machine Learning Research*, vol. 8, no. 2022, 2021.

[38] C. V. Priscilla and D. P. Prabha, "Influence of optimizing xgboost to handle class imbalance in credit card fraud detection," in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pp. 1309–1315, 2020.

[39] B. Baesens, S. Höppner, and T. Verdonck, "Data engineering for fraud detection," *Decision support systems*, vol. 150, pp. 1–13, 2021.

[40] R. I. T. Jensen, J. Ferwerda, K. S. Jørgensen, E. R. Jensen, M. Borg, M. P. Krogh, J. B. Jensen, and A. Iosifidis, "A synthetic data set to benchmark anti-money laundering methods," *Sci Data*, vol. 10, no. 661, 2023.