Reducing Alert Fatigue in Credit Card Fraud Detection Through AI Model Optimization and False Positive Reduction.

Mitchell de Vries
Master Applied Artificial Intelligence
Hogeschool van Amsterdam
Amsterdam, Netherlands
Email: mitchell.de.vries3@hva.nl

***Abstract*—This paper discusses ways to decrease alert fatigue within credit card fraud detection systems. A state of alert fatigue may develop in fraud detection employees because of an overwhelming number of false positive alerts, potentially diverting their attention from genuine fraud cases. In this paper, we take a baseline model in the form of a winning submission from a Kaggle competition and work on several types of sampling strategies and feature engineering methodologies. We worked on enhancing the AI architecture to bring down the rate of false positives without performance degradation. Better feature engineering and oversampling can reduce the rate of false positives from 41% to 15%—a significant promise in using such techniques to cut down alert fatigue and enhance fraud detection process efficiency.**

I. Introduction

One of the biggest problems facing the financial sector is credit card fraud detection, which seeks to quickly detect fraudulent transactions in order to reduce losses [1]. Picture yourself trying to buy a 500-dollar Parmesan cheese. When you want to pay for the cheese at the register, you use your credit card. You swipe your card, and it says denied. You try again. Denied. You walk out of the store not with a cheese but with some embarrassment. When walking outside, you get a call from your credit card company telling you that your card is blocked because of a suspicious payment. The credit card company thought there was a fraudulent transaction happening with your card, so it blocked it.

A. Problem

The problem that clients of a credit card company are facing is that sometimes their transactions are seen as fraudulent. When this happens, their cards get blocked, and the client can no longer pay with their credit card. This creates an awkward position for the client. On top of that, it creates a distrust between the client and the credit card company. This distrust can potentially lose the credit card company a client, thus losing money [2].

Furthermore, a second problem is that fraud detection employees can overlook real fraud cases when there are a lot of false positives. This is also called alert fatigue [3] or the crying wolf syndrome. For this research, we will be referring to alert fatigue. These 2 problems give significance to the research for credit card companies. Firstly, the company retains more clients when there is a large trust between the client and the company. Secondly, finding fraudulent cases will become more efficient for employees when alert fatigue is reduced [3].

According to this research [3], alert fatigue can lead to different consequences, such as missing genuine fraud cases, increased workload stress, and decreased efficiency. The study [3] has shown that employees show alert fatigue when alerts are repetitive, which is something that happens when there are a lot of false alerts.

B. Existing work highlights

There is a lot of literature and existing work in credit card fraud detection. Credit card fraud detection started with rule-based systems [4]. Like blocking the card when the transaction amount is higher than 1000,- euros. A study in 2022 [5] shows that when applying CNN layers on top of an already existing model like XGBoost,KNN, or logistic regression, a precision of 93% can be achieved. Different AI techniques could be used for credit card fraud detection, supervised-, unsupervised learning, and hybrid models. Some techniques that are used a lot in credit card fraud detection are naive bayes [6], Random forest [7], Support vector machine [8] , logistic regression [9] or decision trees [8] . All these researches are focused on getting the best accuracy by improving the AI model. However, they do not show the importance of alert fatigue and how to reduce alert fatigue. But only show how to create the best working models for credit card fraud. Reducing alert fatigue is not something we can find in the literature regarding credit card fraud detection. We do see it in the medical field or the cyber security field [10] [11]. For example, in the medical field, when AI is implemented on an MRI scan, it alerts the doctor when it sees an anomaly. When the alerts are repetitive, alert fatigue can

start to show its effects. These effects are increased workload stress and decreased efficiency.

C. Gap

There are already enough studies available to improve an AI model for credit card fraud detection; for instance, various studies have been conducted to improve the detection of credit card fraud using advanced AI models combined with the most sophisticated techniques [1], [4], [12]. The most important element still missing in these researches is the phenomenon of alert fatigue. This research attempts to fill this gap by examining the AI model and providing ways to prevent alert fatigue. With this research, there will be an argument to redirect the aim from wanting the best performance like AUC-ROC or precision and recall. To making an AI model that does not create alert fatigue for the end-user when using it.

Alert fatigue can be solved by improving an Alert Management System. This means improving the design of the application a user is using. However, this is beyond the scope of our research. This research will only focus on reducing alert fatigue by improving an AI model. Given the time constraint of this research and the desire to reduce alert fatigue, which can be done by improving the AI model, this research will use an existing AI architecture. This AI architecture will be improved using different techniques. The choice of what kind of AI architecture will be used has to be researched in the next chapter.

D. Research question

Right now, the problem is clearly described which is that credit card company clients face inconvenience and distrust when legitimate transactions are flagged as fraudulent and their cards are blocked. At the same time, fraud detection employees experience alert fatigue from frequent alerts. The research question that is going to be solved is.

RQ: How to reduce alert fatigue for the employee in the process of credit card fraud detection?

The sub-questions that will be discussed in this paper are:

SQ1: How to measure alert fatigue in an AI model?
SQ1: How to reduce alert fatigue in an AI model?
When these questions are answered, there will be an understanding of reducing alert fatigue in credit card fraud detection systems. Thus improving both the efficiency of fraud detection and the working conditions of fraud detection personnel.

E. Stakeholder analyse

After identifying the problem, we must determine the target group and stakeholders. In figure 1, a stakeholders map can be seen. Here, all the essential, important and interesting stakeholders are mentioned. These stakeholders are chosen

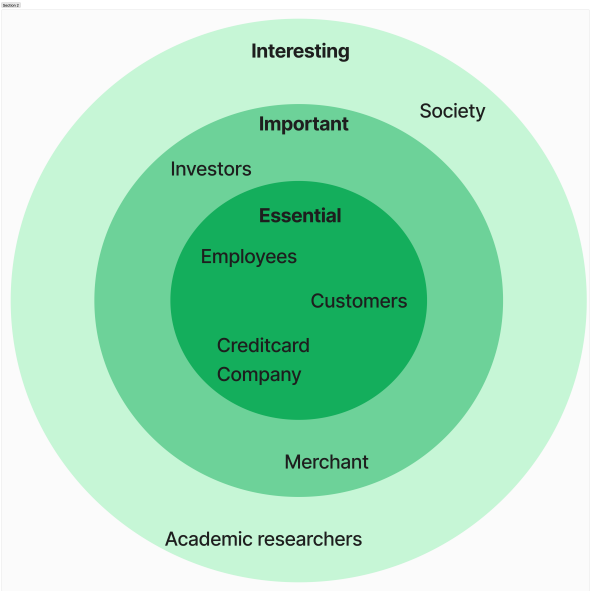


Fig. 1: Stakeholder map

by looking who is directly impacted by credit card fraud detection. Moreover, a presentation by a bank gave more insights into different stakeholders which were then added to the stakeholder map.

1) Essential:

- Employees are an essential group because the research directly impacts their workload and efficiency. Reducing alert fatigue allows them to process alerts more accurately and quickly, leading to lower stress levels, improved mental health, and a more positive work environment as is stated in [3].
- The credit card company itself is another essential stakeholder. Employees can perform their work more effectively by reducing alert fatigue. This can result in cost savings for the department and an improved workplace.
- Customers are also essential stakeholders. They benefit from a more efficient fraud detection system, which reduces the likelihood of their legitimate transactions being flagged as fraudulent. This gives a better relationship to the client and credit card company.

2) Important:

- Investors find a more efficient credit card fraud detection important because this means the company will make more profit.
- Merchants benefit because a more accurate fraud detection system blocks fewer legitimate transactions. This can potentially mean higher sales.

3) Interesting:

- Society benefits when fraud is detected and prevented. Faster and more accurate fraud detection helps maintain the financial system's integrity, which is important for public trust.

- Academic researchers find this research interesting as it helps with creating a better understanding of alert fatigue. A concept often explored in medical fields or cybersecurity but less so in credit card fraud detection.

A certain level of trust is needed in the AI application. When looking at book [13] we can see that there are different level of trusts. For our application, a trust level 7 is expected. Level 7 means "The AI executes automatically and informs the human". This level is chosen because the AI makes the decision if something is a fraud. And that decision persists until an employee vetoes it. With Level 7 a human stays in the loop of the process. Which is the case in our process.

F. Proposal

This research will use the CRISP-MLQ development methodology [14], which offers a structured framework for an AI project. We will focus on improving an AI model to reduce alert fatigue.

CRISP-MLQ Methodology Application:

- Business and Data Understanding: We will start by identifying the business objectives and data requirements. This step also involves cleaning and organizing the data for analysis.
- Modeling and evaluating: We will apply machine learning techniques to develop and refine the AI model. The model will be evaluated for accuracy and effectiveness, specifically focusing on reducing alert fatigue.
- Deployment and monitoring: The improved model and alert management system will be deployed in real-time. Continuous monitoring ensures the system remains effective and prompts necessary improvements.

This study uses only the first two phases of the CRISP-MLQ methodology. The phases: Business/data understanding and model development. These last phases are for deploying and maintaining the model, which is something we do not cover in this study.

Potential Limitations and Challenges:

- Data Quality: Ensuring high-quality data is important and can be challenging. This is because financial institutions do not share their data, the reason being it contains sensitive information about their clients.
- Model Complexity: Developing a model that reduces alert fatigue without overfitting or making the model too complex requires a delicate balance.
- Resource Constraints: Time and computational resources might limit the extent of model improvements.

II. Background

The primary objective of this research is to use machine learning techniques to reduce alert fatigue in credit card fraud detection systems. This involves understanding various AI techniques for credit card fraud detection, selecting an architecture based on specific requirements, and researching different AI techniques to enhance this architecture. Furthermore, we also need to know how to measure alert fatigue.

A. Introduction to fraud

Fraud is an intentional lie told against another person to obtain an unfair or illegal advantage. Fraud is a lie used with the intention of misleading or damaging another person or party [15]. Fraud may be committed in many forms and contexts. In the world of credit card fraud detection, there are different kinds of frauds [1]. These frauds are online, offline, or via telecommunication.

- Online fraud occurs when a card is used via the internet, phone, or shopping in the absence of the cardholder. For example, I have a picture of the front and back of someone else his credit card. And I use these numbers to pay for a Netflix subscription or send money to my own bank account.
- Offline fraud is committed when the credit card is physically stolen and used in a shop.
- Telecommunication fraud is trying to receive money from a person while creating the illusion that you are someone trusted or selling them something without actually selling them something.

For this research, telecommunication fraud will not be further investigated because this happens when a person misleads another person, which has to do with psychology. When improving an AI model, we need to choose between online or offline fraud. This is because their data will be different. With offline fraud, you could look at the address of the shop, CCTV or signature data. With online fraud, you do not have this kind of data. For this research, we will focus on online fraud. This is because there is a large increase in the amount of online transactions and fraud taken place with online transaction then with offline transactions [16].

B. Credit card transaction process

For this research, the focus will be on online fraud because of the increasing amount of online transactions and online credit card frauds [16].

It is important to analyze the different parts of an online credit card transaction to understand how it works. The operation can be broken down into four crucial parts, each designed to ensure a transaction is conducted safely and effectively. An outline of this is presented below, and an image, 2 that illustrates these steps is depicted below. Keep in mind that this is a simplistic approach of the process and does not show all the steps that are taken in the specific part [17].

1) Customer Initiates Transaction: In this step an user fills in their credit card credentials on the merchant website.
2) Merchant Processes Payment: From the website of a merchant, the details of the transaction are routed to the

payment gateway. Additionally, it sends the information securely to the credit card network like VISA.

3) Credit Card Network Authorization: The network of a credit card, such as VISA, receives a request for a transaction and forwards it for authorization to the issuing bank. It performs checks for available credit and potential fraud.
4) Bank Responds to Network: The issuing bank sends the response through the credit card network to the merchant. Once approved, the transaction is completed and funds transfer from the customer's account to the merchant's account. In case of a denied response, it informs the customer of the failure.

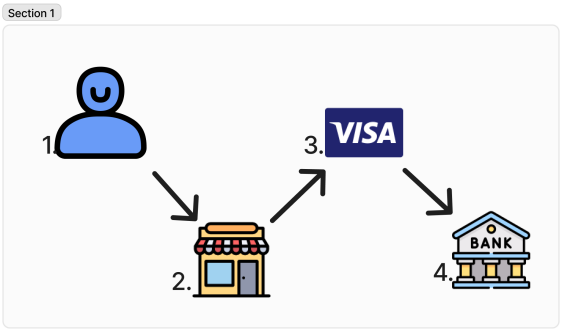


Fig. 2: Credit card process

This process shows 4 different steps and their position in the credit card payment process. However, this does not show the fraud detection part of the system. The place where fraud detection is done may differ from country to country according to law. It is often a country law, that companies must have a fraud detection system [18], [19]. This means that steps 3 and 4 each have their own fraud detection systems. However, these companies can also outsource it to external companies that specialize in fraud detection.

Credit card fraud detection is a complex process involving multiple steps to ensure transactions are legitimate and secure. Various companies, such as ICS, MasterCard, Maestro, Visa, and American Express, have different strategies for detecting fraud. While the fundamental process is similar across them, the methods employed can differ significantly.

Below is an overview of the credit card fraud detection process, illustrated by a simple diagram highlighting the main steps involved [20].

- Transaction Data Collection: Upon payment initiation, detailed data such as the amount, time, IP address, and other related information are collected and submitted to be analyzed.
- AI Model Analysis: An artificial intelligence model is then used to analyze the processed data and identify fraudulent activity. Each company applies various techniques and algorithms to assess risk and the possibility of fraud.
- Original Determination: Based on the processing of the AI model, it classifies the transaction as possibly fraudulent or otherwise as legitimate:
  Transaction no fraud: Transaction will continue.
  Fraud detected: The card is blocked, usually temporarily, and the suspicious transaction is blocked.
- Human Review: If the transaction is labeled as a fraudulent one, then a human analyst reviews the case with the intention of confirming the decision made by the model. In this way, there is ensured both accuracy and feedback to improve the model.
- Final Decision:
  Fraud Not Confirmed: An analyst unblocks the card when a valid transaction is found, and the payment processing proceeds.
  Confirmed Fraud: If fraud is confirmed, the card will stay blocked to prevent additional unauthorized use.

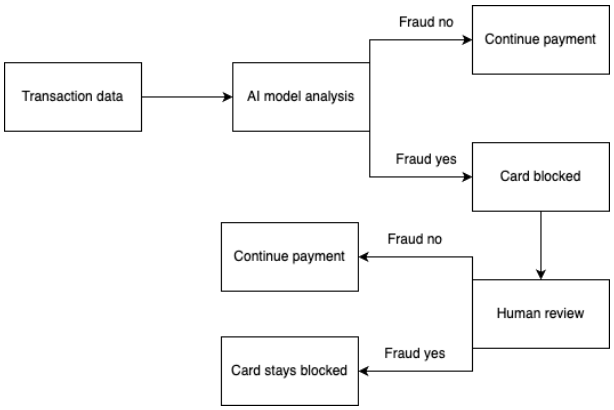


Fig. 3: Fraud detection process

C. Ethical considerations

The implementation of AI models in credit card fraud detection systems raises a number of ethical considerations. Key to protecting the interests of the stakeholders involved, and most importantly the customers, the staff, and credit card companies themselves, is ensuring that the use of AI is responsible and fair.

AI models have to be trained on diverse and representative data sets to make them averted to all forms of bias that could lead to unfairness to other groups. In the case of credit card fraud detection, biases in data can result in a higher false positive rate against certain demographics, leading to unexplainable and financial impacts. It is therefore very important to continually monitor the AI models for such biases that may creep in and correct them.

AI models should be transparent, and the process adopted towards the decision-making is explainable. This is of utmost importance in establishing user trust when some transaction has been marked as fraudulent. It should be possible to explain to customers why their legitimate transactions were denied and not leave them frustrated and untrusting of a credit card company. Explainable AI models will help the teams in fraud detection to understand and verify the decisions of the model so that, in case it is a genuine case, they can catch it quickly, thereby helping the employee's in the process by providing information.

The detection of credit card fraud uses personal and financial information. It is crucial to ensure privacy and the data protection of that information. This can be aligned with guidelines related to data protection such as the General Data Protection Regulation (GDPR) [21] in the EU. Both customer data should be anonymized and encrypted; access to sensitive information is controlled and monitored strictly.

Generally, ethical AI deployment in credit card fraud detection involves multiple aspects that emphasize fairness, transparency, privacy, accountability, and employee well-being. By taking due consideration of these ethical concerns, credit card companies will enhance the effectiveness of their fraud detection systems without losing their customer's and employee's loyalty and confidence.

D. AI techniques in fraud detection

There were no artificial intelligence models in the early days of detecting credit card fraud. Detecting fraud consisted of rule-based systems [4]. For example, if the merchant is on a blacklist, then block the card. Or if the transaction amount is higher than 10.000,- block the card. When AI was introduced to credit card fraud detection different methods and techniques were researched. We separate these methods into 3 sections. Supervised-, Unsupervised learning and Hybrid models.

1) Supervised learning: Supervised learning is particularly valuable in credit card fraud detection due to its effectiveness in classifying transactions as fraudulent or non-fraudulent based on historical data. Fraud detection systems rely on labeled datasets, marking each transaction as 'fraudulent' or 'non-fraudulent'. By learning from these labels, a supervised model can identify existing fraud patterns in transactions. This capability is important for financial institutions as it helps minimize false positive transactions that are incorrectly flagged as fraudulent while accurately identifying true fraudulent activities, thereby protecting both the consumers and the financial institutions.

Some notable models are two papers published in 1997 [22] [23]. These papers introduced a method called database mining neural network. This method was widely used by different researchers. The paper proposed an improvement on the rule based system by finding changes in user behaviour. Furthermore this paper also developed a GUI to use, so a person can train their own neural network without the need of coding. The neural network was trained on synthetic data which consisted of four columns: Category of purchase, dollar amount, hours between previous purchase of the same category and fraud. There were 2 flaws of the research which are the low amount of features and not being tested in the real world. The low amount of features do not give the model a well enough understanding of a user spending pattern. But these studies introduced us to the field of user behaviour and spending patterns. Where if someone does something outside the expected pattern, it can be seen as fraud. Initially, this was pretty straightforward and the system already alerted if you were on holiday. This was because it did not fit the pattern of buying coffee/lunch every working day. Later, new models were developed and fine-tuned to achieve better performance.

Some other supervised models that are widely used in the past few years are naive bayes [6], Random forest [7], Support vector machine, logistic regression or decision trees. These models have a few advantages, they are often easy to interpret. This allows us to understand how the model reaches its conclusions, which is useful for tasks where explainability is important.

However, supervised learning models also have disadvantages. They, need labeled data for training. However, getting labeled data is expensive and time-consuming. Also, not all non-fraud cases are really non-fraud. It is possible that someone has had a fraudulent transaction on their credit card but was too embarrassed or neglected the fraud. And did not tell the credit card company about it. Furthermore, the model's performance can change significantly when real-world data changes. This is called concept drift [24]. Lastly, the real-world data is mostly imbalanced. There are more real transactions than fraudulent transactions resulting in the supervised learning model becoming biased. There are techniques to overcome this, these techniques are called sampling techniques and will be talked about in the next chapter.

2) Unsupervised learning: Unsupervised learning is used to identify unusual patterns or outlier transactions that may show fraudulent activity. However, domain knowledge requires determining whether something is an unusual or anomalous transaction. An advantage of unsupervised learning is that it can detect new forms of fraud faster. But this comes at the cost of a thorough analysis of the outliers. Supervised learning uses labeled data which takes some time to create. In this time, the new form of fraud can slip through the system because the system is not trained for the new form of fraud. Unsupervised learning does not suffer from this. In fact, unsupervised learning can be retrained much more frequently to recognize new forms of fraud.

Techniques like clustering [25], OCSVM [26], auto encoders [27] or generative adversarial networks [28]. Mainly look at patterns. Some patterns that can be looked at are time-based, location, or transaction patterns.

The advantage of using unsupervised learning is mainly that it can spot new forms of fraud. So concept drift is less likely to happen. Another advantage is that it does not need labeled data for training. The disadvantage is that it has a harder time reducing false positives or the ROC-AUC in comparison to supervised learning.

3) Hybrid models: While both supervised and unsupervised learning offer advantages in credit card fraud detection, a hybrid approach can use the strengths of both. Supervised models are good at identifying known fraudulent patterns but struggle with entirely new fraud schemes. On the other hand, unsupervised models can detect anomalies but often generate many false positives. Combining these techniques allows a hybrid model to learn from labeled data for known fraud while using unsupervised techniques to flag suspicious outliers.

Some hybrid models are [29], [30] or [31]. We are going to analyze further the hybrid model of (F. Carcillo, et al) [29]. This model combines unsupervised and supervised learning. During the first part of the process, unsupervised techniques like clustering and anomaly-detection algorithms identify transactions that differ from normal patterns without depending on labeled data. The labeled anomalies, along with historical data, are fed into supervised learning algorithms that classify future transactions into fraudulent or legitimate classes based on learned patterns. This combines the strong points of both methodologies in such a way that the model discovers new fraud patterns and is adaptive to changes in behavior, thereby boosting robustness and, in effect, overall detection accuracy.

The advantage of a hybrid model is that it can detect new patterns without the need for labeling. The disadvantage is that it is complex and not explainable.

E. Measuring alert fatigue

Alert fatigue, also known as crying wolf syndrome, is researched mainly in the medical and cybersecurity fields [10], [11] [32]. It is proved that alert fatigue is hard to measure within a person due to its individual specificity. Therefore, it is important to determine when alert fatigue occurs in people. It should be noted that alert fatigue usually develops when the alerts are overly repetitive, especially if a majority of the alerts are known to be false. This might build up frustration and fatigue as individuals start to take such alarms for granted. This brings us to the conclusion that we can measure alert fatigue by measuring the false positive ratio. If the false positive ratio is high, it means the employees are more likely to show alert fatigue. The lower the false positive the less likely it is that employees start showing alert fatigue.

F. State-of-the-art

This research aims to reduce alert fatigue in credit card fraud detection by improving the model to maintain accuracy while reducing false positives. This study employs sophisticated supervised learning models due to their higher probability of reducing false positives.

The research of F. Carcillo, et al [29] shows an approach to use an anomaly detector. This detector creates scores that are used in the supervised learning model. The unsupervised model also takes into account different granularity. This means it detects outliers for global clusters and local clusters.

Another anomaly detection model is that of (Malik, et al) [33]. Their model uses a deep learning technique called a feed-forward neural network. Transactions go through the network, and anomalies are identified by analyzing various features. The model is designed to consider both common and less common spending patterns to catch fraud. Furthermore, in this research, they fine-tune the model to get the best accuracy. Anomaly detection is closely related to pattern recognition which is a well-known technique in credit card fraud detection. [34]

Because of requirements numbers 4,5 and 6, we will not be pursuing these models. The winning submission from the Kaggle competition [35] does have a publicly available dataset and model. This model has an ROC-AUC of 0.945. The confusion matrix can be seen here in table I.

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP): 141444	False Negative (FN): 1091
Actual Negative	False Positive (FP): 1914	True Negative (TN): 2730

TABLE I: Confusion matrix

As can be seen in the confusion matrix, the false positive rate is 41%. This means for every 100 alerts, there are 41 false alerts. Furthermore, this model is well documented and comes from an online Kaggle competition. Because this model follows the requirement: 4,5 and 6 we will be using this AI architecture to reduce alert fatigue by improving it and thus reducing false positives.

III. METHOD

This section elaborates on the steps and techniques used in the research to reduce alert fatigue associated with credit card fraud detection. It includes model requirements, the baseline

requirement, and criteria; this is followed by descriptive data analysis, model design, and model performance evaluation. This study applies advanced feature engineering, regularization techniques, and sampling methods in handling data imbalance. Requirement analysis was conducted based on the standard methodology of MoSCoW prioritization.

A. Requirements and Criteria

A requirement list ensures that the results and process are up to standard. A requirement list for different categories is created in table II. The requirements are created using insights from other researchers, domain knowledge and school. MoSCoW is used in the requirements table. MoSCoW is a prioritization technique to help understand the importance of different requirements. Moscow stands for Must have, Should have, Could have, and Won't have. In Table III, we can see the requirements of the AI model. This table covers the goal, input/output data, performance, and training data.

The product requirements are shown in Table II. These requirements guide and help decision-making in the design phase. Furthermore, they are also meant to create expectations from the AI model. The requirements are divided by category and as stated earlier a prioritization technique called MoSCoW. For the categories, we have: Model, Dataset, Juridical, Ethical, and Organisational. These categories are chosen because of certain learning goals from school. In table III, we can see the different model requirements that are expected. With a short description. Here we can see the expected goal of the AI model and what kind of input and output we should expect.

B. Baseline Model Source

The goal of this research is to reduce alert fatigue. To reduce alert fatigue we need to have a baseline model. This model can be developed and trained by us, or we can utilize a sophisticated AI model that has demonstrated good results. Because of the time constraints of this research, we have chosen to use an already existing AI model. The model that will be improved in this research comes from a Kaggle competition [36]. This competition is hosted by IEEE-Computer intelligence society(IEE-CIS). IEEE-CIS is an institute of engineers who are focused on design, application, and development of computational intelligence. The competition where this institute is the host, is trying to develop the best AI model in credit card fraud detection. They are partnering with VESTA. A credit card fraud detection company for e-commerce. VESTA provides a real-world dataset that has a wide variety of features. Submissions are evaluated by the ROC-AUC. Participants also have a large incentive to make the best working model because of the reward. The prize pool consists of 20.000 euros. The competition lasted three months.

For this research we will improve the AI model from the competition's winners. This is because everything is well documented, and the dataset is publicly available per requirements 4,5 and 6. The winner of this competition is a team of 2 people. Who had an AUC-ROC of 0.945 [37] and a false positive rate of 41%, which was calculated locally. The confusion matrix can be seen in this table I

As can be seen in the confusion matrix, the false positive rate is 41%. This is calculated by getting the sum of the True negative and False positive. Divide this sum with the FP, and then multiply by 100. So for every 100 alerts, there are 41 false alerts.

When the model was trained locally, the AUC-ROC was plotted as shown in figure 4. It is clear that the model is overfitting. This could be because there are too many features, the model is too elaborate, or insufficient training data. In the submission, I see no reference to any overfitting of the model. Although it is unlikely the authors did not notice the overfitting, there is no mention of it in their submission.

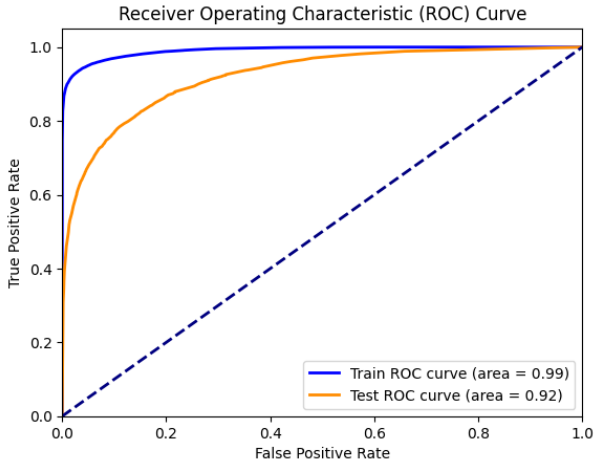


Fig. 4: Overfitting

C. Data Description

The biggest hurdle was finding available data. A lot of financial institutions do not share their data because it contains personal information of their clients. Nonetheless, one dataset that is used widely for this case is a European dataset [38] that consists of 39 features. The con with this dataset is that there is no information about the features. However, as discussed in the previous section, the Kaggle competition followed requirements 4,5 and 6. This means that the dataset also had to be publicly available. The dataset that the Kaggle competition used comes from a company called VESTA, which is a company that specializes in fraud detection from credit cards in online transactions. VESTA is a company that is very relevant to this research. Because this data is publicly available. Anyone can access it and train their own model on it. Furthermore, there are way more features(434), and there is more information about the different features available in contrary to the European dataset. The VESTA data is broken

ID	Description	Accountability	Category	MoSCoW
1	The metric is going to be confusion matrix and the ROC-AUC	The RQ is to reduce alert fatigue. Alert Fatigue can be measured by looking at false positives. And the recall and precision	Model	Must have
2	An improvement of an already existing advanced architecture.	Reducing alert fatigue is a complicated process. Because of time and the purpose of the study, an existing model is already being improved.	Model	Must have
3	The response time should be under 0.5 seconds	As a credit card company, you want to block the transaction before it has been completed. So this check has to be done in the process of validating the payment.	Model	Must have
4	The advanced architecture has to be publicly available	My research has to be replicatable for anyone that reads the research	Model	Must have
5	The data has to be publicly available	My research has to be replicatable	Dataset	Must have
6	The advanced architecture has to have existing code and be well explained	This is because there will no be a lot off time in creating the code off the model and having to understand it.	Model	Should have
7	The AUC-ROC needs to be above 0.92, and the false positive rate under 41%	When these metrics are acquired, there is an improvement in the advanced model	Model	Must have
8	Responsible AI	AI models should be transparent and explainable so that decisions can be understood and justified.	Ethical	Should have
9	Fair treatment	The model should provide fair and unbiased results, without discrimination based on gender, race, age, etc.	Ethical	Must have
10	Audit trail	There should be an audit trail for all transactions and decisions made by the model to ensure accountability.	Juridicial and organisational	Should ahve
11	Incident response plan	There should be a plan for dealing with incidents where the AI makes incorrect decisions.	Organisational	Should have
12	Alert system	The system should generate alerts for suspicious transactions and present them clearly to fraud analysts.	Functional	Must have
13	Quality criteria	There has to be an evaluation of different quality criteria: performance and model complexity.	Organisational, Juridical, Ethical and Model	Should have

TABLE II: Requirement list

Requirement	Description
Goal	The aim of the model is to create an alert when there is a fraud on a credit card
Input data	Transaction and identity data
Output data	Fraud, Boolean yes or no
Performance	Lower False positives and same ROC-AUC as a chosen advanced model.
Training data requirements	Credit card data for online transactions that is publicly available

TABLE III: AI Requirements Table

down into two categories. The two categories are transaction and identity.

1) Transaction: This category consists of 393 features. Some data you could find include bank, card type, time delta, country, or product code.

2) Identity: In this category, there are 41 features. Not much is known about what is in this category. This is because a feature anonymization technique has been used. This is done to make the data not traceable to a place or person. Several techniques can be used for this e.g. PCA or auto encoders. The reason why financial institutions do this is because they do not want people to be able to trace the data back to a customer of them. The technique that has been used for this dataset is not clear.

In table IV, we can see the different features and their datatype. This table will also be referred to later on in the research. As you may notice in table IV, there is no user ID(UID). There is only a possibility to track something through the transaction ID. It is unclear why VESTA does not provide a user ID. A possibility could be that they neither get a user ID from the financial institutions. Because they are an external company. Or that they deliberately removed it so it is harder to understand what the patterns of the users are. This way the transactions can not be traced back to a customer. Having a user ID is quite important for different feature engineering techniques. This is because most feature engineering techniques are based on creating a pattern for a user [39].

Feature name	Type
ID01 - ID14	Number
ID15 - ID16	Categorical
ID17 - ID32	Number
ID33 - ID34	Categorical
ID35 - ID38	Boolean
Device type	Categorical
Device info	Text

TABLE IV: Identity features

D. Model Architecture

To understand what can be improved about the AI model from the Kaggle submission [35]. We need to analyze the process which can be seen in figure 5. This figure does not show the different parameters for the models or the feature engineering techniques used.

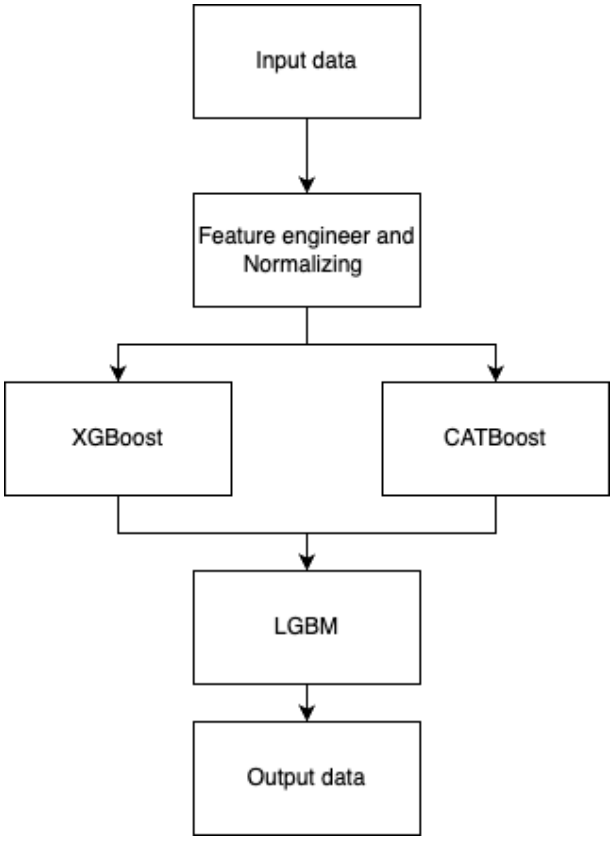


Fig. 5: Architectuur

1) Process of the AI model: The first step after retrieving the input data is to normalize, feature engineer, and feature select the data. After that the model XGBoost and CATBoost are trained with the data. The outputs from these two models are fed into the LGBM model. This LGBM model gives an output if something is a fraud yes or no. The techniques used for feature selection are listed below.

- forward feature selection (using single or groups of features)
- recursive feature elimination (using single or groups of features)
- permutation importance
- adversarial validation
- correlation analysis
- time consistency
- client consistency
- train/test distribution analysis

This research will not explain all the feature selection tech-

niques because this is unnecessary, and we will not improve the implemented feature selection techniques. But two notable feature selection techniques are:

- Forward Feature Selection [40]: This is an iterative process that initiates with no features and adds one feature at a time; the feature added in every step maximizes improvement for the model. This procedure continues until the added features reach saturation, beyond which they do not significantly improve the model any further.
- Permutation Importance [41]: It quantifies decreases in model accuracy when a particular feature is shuffled. The method quite clearly explains which features are important and which ones are not. By disrupting the relationship between the feature and the target variable, the impact on the model's accuracy can be observed, revealing the true significance of each feature.

Because of the amount of different sophisticated feature selection techniques. I will not be pursuing other feature selection techniques. Looking at how the architecture was created, we noticed that no sampling techniques were used. Looking at the data and whether it is balanced, we can see that only 3.6% are fraud cases. This means that there is a possibility that overfitting will occur in the AI model. Which is something we see in figure 4. We can use sampling techniques to improve the balance in the dataset and thus reduce overfitting [42]. Different sampling techniques are explored in section E.

The authors of the AI model also use different feature engineering techniques. Their method consists of thinking about possible good combinations and testing whether they work. If the ROC-AUC goes up, they keep the feature. If it goes down, they remove it. However, the authors did not research existing feature engineering techniques that could provide better results. Different feature engineering techniques are talked about in section E

Model	ROC-AUC	False positive rate
XGBoost	0.932	21%
CATBoost	0.94	22%
LGBM/Architecture	0.928	41%

TABLE V: Performance architecture

In table V, we can see the performance of the first two models. Ran locally instead of taking the values from the Kaggle competition. And looking at the third row, we can see the performance of the LGBM model, which is also the architecture's performance. The LGBM is also the architecture performance because the LGBM only works with the outputs from the previous two models, which are the XGBoost and CATBoost models. We can see that when the goal is to reduce alert fatigue, the XGBoost model works best. However, the incentive from the Kaggle competition was to achieve the highest ROC-AUC. So, the authors created an architecture with a higher false positive rate and a higher ROC-AUC, which gave them the winning architecture. It is also important to note that the LGBM/Architecture has a lower ROC-AUC then the other two models and the submission. That is because these are the metrics when the models were trained locally. In the result section, we give an explanation to how this is possible

Looking at the false positive rate we can see that out of every 100 alerts there are 41 false alerts. For this research, the ROC-AUC 0.928 and false positive rate of 41% will be the baseline model that we need to beat.

When looking at figure 6 where we can see in red the fraud cases and in blue the non-fraud cases. 3.6% of the dataset is fraud, and the rest is non-fraud which means there is an imbalanced dataset. This research shows that when we used an oversampling technique in an imbalanced dataset we can reduce the false positives [43]. As stated earlier, the AI architecture from the Kaggle competition has not done that. There is no explanation as to why the authors did not implement an oversampling technique.

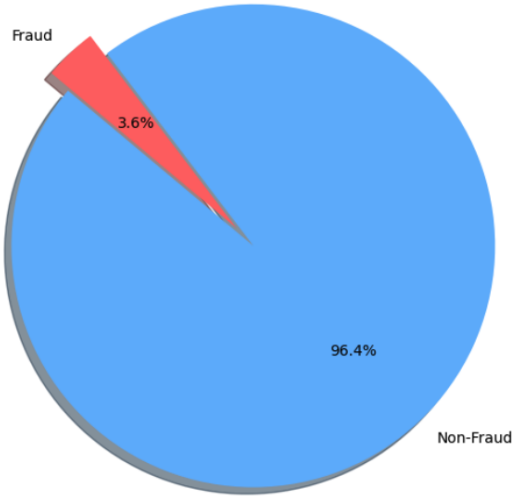


Fig. 6: Amount of fraud

E. Improvement Techniques

1) Sampling: Sampling falls under the architecture's preprocessing steps. From the previous section, we saw that only 3.6% of the dataset is fraudulent. In the architecture, no sampling techniques were used or tested. But as seen in the research from (Priscilla, et al) [43], there is a reduction in false positives when implementing sampling techniques.

The research from (Kotsiantis, et al) [44] comprehensively explains two techniques to battle imbalanced datasets. Those techniques are called oversampling and undersampling.

- Oversampling: This involves randomly duplicating minority data points. To create a more balanced dataset. While this can potentially improve the performance of the model. It also has a few disadvantages. When duplicating data points, there is an increased risk of

overfitting. This is because the data points are too closely related to certain data points. To mitigate this, we can use different methods like SMOTE [45] or ROSE [46], which generate new minority data points by adding variation to the data points.

- Undersampling: This is a technique where data points are randomly selected and removed from the dataset. This only happens to the majority of the class. While this method helps to reduce bias towards a majority class. It can also have a negative impact, which is that it can remove valuable information. Furthermore, it can also distort the sample distribution, making it less representative of the actual non-fraud transactions.

Research from (Priscilla, et al) [43] has concluded that SMOTE [45] provides no improvement over a boosting model. The research from (Priscilla, et al) [43] uses the same dataset as our dataset. However, future research states that other techniques are yet to be investigated e.g. ROSE [46].

SMOTE is created to synthetically generate minority instances that are randomly selected by k-nearest neighbors. A disadvantage is that it can lead to overfitting. Which is already a problem with our model as seen in figure 4. ROSE also generates minority instances but does this with a bootstrap-smoothed approach. An advantage is that it creates more diverse samples. But it is prone to create a lot of extra noise because of the diversification.

- Select $y^* = Y_j$ with probability π_j.
- Select $(x_i, y_i) \in T_n$, such that $y_i = y^*$, with probability $\frac{1}{n_j}$.
- Sample x^* from $K_{H_j}(\cdot, x_i)$, with K_{H_j} a probability distribution centered at x_i and covariance matrix H_j.

In this list, we can see the different steps ROSE performs. The first step ROSE performs is to calculate a class probability. The class probability of the fraudulent feature is then calculated for our case, and the probability of fraud "Yes" is 0.035. Because it is lower than the probability of fraudulent feature "No". The probability of 0.035 is chosen. Then, a transaction has to be chosen, which is chosen randomly from the dataset where fraud must be "Yes". Generate a new sample by looking at the vector space of the transaction in question. Then, use the covariance matrix H to create the new data point. The covariance matrix H gives variation to the values so that the same data point is not always created.

The final step of the ROSE process is what gives ROSE a better edge than SMOTE. SMOTE follows almost the same steps except at the end. ROSE uses a covariance matrix H to generate new data points with a variation. SMOTE uses linear interpolation, which has a hard time capturing the underlying distribution of the minority class, especially with a high-dimensional dataset [47] which our dataset is.

2) Regularization: As stated earlier the AUC-ROC in figure 4 shows overfitting. We will implement sampling methods that could lead to more overfitting [48]. So, to mitigate the overfitting, we can implement different techniques like regularization or k-folds. Our architecture consists of XGBoost and CATBoost, which are boosting models. According to this research when dealing with boosting models, it is best to use regularization techniques [49].

We can choose between different regularization techniques. The most used regularization techniques are L1 and L2 [50].

- L1: This regularization technique adds a penalty equally to the value of the coefficients to the loss function. What this means is that it creates more sparsity in the model. It is particularly useful when only a few features are expected to be significant [51].
- L2: This regularization technique adds a penalty equal to the square of the coefficients. This means that in contrast to L1, this technique ensures most of the features contribute equally to the model. It is especially effective when many features are important for the model.

Because not all 393 features are expected to be equally significant, we will use L1 regularization.

3) Feature engineering: Feature engineering for credit card fraud detection involves creating new features from the data to improve model precision. Our goal is to reduce the false positives of an AI architecture that consists of boosting models. In the research from (punmiya, et al) [52], they show that when using feature engineering with boosting models, it can reduce the false positive. In the Kaggle competition, some feature engineering is done in the AI architecture by the authors. But as read in the author's submission. Their strategy was to understand what would potentially benefit the model. Create the new feature. If there is an improvement, keep the feature; otherwise, remove the feature and create a new one. Keep repeating this process until you are out of ideas. There is a lot of research done in feature engineering, which they did not use, for example the research from (Bahnsen, et al) [39]. These feature engineering techniques are quite sophisticated and have improved performance if you choose the correct one.

Most feature engineering techniques consist of aggregating transactions to create a spending pattern. In the research of (Bahnsen, et al) [39], they expand the transactional behavior by analyzing the periodic behavior of the time of a transaction using the von Mises distribution. The math will be explained later on in this section.

To apply feature engineering according to the research of [39]. We need to have an UID for clients. Which, as stated before, the dataset does not have. This is important for feature engineering because most feature engineering strategies are built on user behavior, as talked about in Table IV.

First, we tried to manually create an UID and went through the transaction table. In table IV, you can see which features it contains. As you can see, many features are anonymous, which was also the intention of VESTA. There are also a few columns that we can already conclude we can take out. For example, the features that show which web browser the transaction was made on. This may be useful for our fraud detection model but not for giving a customer an UID. After all, I can use my own credit card on different computers. There is also a column for screen size, which can also be removed.

During further manual research, it turns out that it is quite difficult to create an UID manually. So, we need to start looking for other solutions. On the Kaggle competition page, the author's submission added UID. Their technique consisted of combining different columns. Each transaction is recorded with a timestamp called TransactionDT, which represents the number of seconds from a reference point. They converted this to days instead of seconds. They did this because days is a more intuitive and manageable time unit. 'card1' and 'addr1' are key features representing user-specific information. These were combined to form a base identifier. As the third step, they calculate the relative day difference. This is done so that all the transactions on a certain day, within a certain amount of days on a certain address, are grouped. And seen as one UID. As the authors stated. It is still possible that multiple users are grouped as the same UID. However, their argument for this is that the AI model can take care of that.

So now we have a UID for a credit card. And we can apply the feature engineering of the research from (Bahnsen, et al) [39]. This study explains 2 different methods. These methods are called Agg1 and Agg2. For this research, we only implement Agg1. The formula for Agg1 is seen in equation 1.

$$S_{\text{agg}} \equiv \text{TRX}_{\text{agg}}(S, i, t_p) = \{\ x^l_{\text{amt}} \mid x^l_{\text{id}} = x^i_{\text{id}} \wedge \text{hours}(x^i_{\text{time}}, x^l_{\text{time}}) < t_p \ \}_{l=1}^{N} \tag{1}$$

Let's look at the first part of the formula.

$$S_{\text{agg}} \equiv \text{TRX}_{\text{agg}}(S, i, t_p) \tag{2}$$

For this function, S is the full transaction dataset. i is the index of a specific transaction in S. The time window in hours is t_p

The second part of the formula is quite elaborate. Because this is the part where the relevant transactions are extracted. $x_{\text{i}}^{\text{amt}}$ is the amount of a transaction for i. x_{i}^{id} is the ID for a card for transaction i. $x_{\text{i}}^{\text{time}}$ is the time of a transaction

When calculating, it gives you a variable S_{agg}. With this, two features are created with these small equations.

$$x^i_{a1} = |S_{\text{agg}}| \tag{3}$$

and

$$x^i_{a2} = \sum_{x_{\text{amt}} \in S_{\text{agg}}} x_{\text{amt}} \tag{4}$$

These two features are created and will be inserted into the dataset. In equation 3 it shows how many transactions a user has made in a certain amount of time. In equation 4, we can see the monetary amount of transactions in a certain time frame. The research says you should experiment with the hours 24, 60, or 168. So we will experiment with these three variables.

F. Evaluation Metrics

We can use the same metric employed in the Kaggle competition to evaluate the model which is the ROC-AUC. The ROC-AUC effectively shows precision and recall. The ROC-AUC is often used to improve recall or precision [53]. This metric shows how well a model is performing but not necessarily what the rate of the false positive is. So, we can use the confusion matrix to see the percentage of false positives and see if this is reduced by applying different techniques. So, we are going to use ROC-AUC to see if there is an improvement compared to the already existing architecture and evaluate the overall performance of the model. We will also use the confusion matrix to see if our goal has been achieved by reducing false positives.

Furthermore, it is also important to evaluate the models individually to track whether the architecture is improving. This is also done with the same metrics: ROC-AUC and the confusion matrix. Why an F1 score, for example, is not used is because the dataset is imbalanced. A high value of true negatives can inflate the precision value. And give us a wrong representation of the performance of the model.

IV. Results

This section presents the findings from techniques implemented in the AI models to reduce alert fatigue in fraud detection. It begins by discussing the baseline performance of the AI model, followed by an evaluation of various improvement techniques, such as sampling and feature engineering. These techniques are compared using metrics like ROC-AUC and confusion matrix to determine their effectiveness.

A. Metrics Used for Evaluation

The performance of the models is assessed with the help of two metrics: ROC-AUC and the confusion matrix. These two metrics, taken together, provide an overall view of a model's accuracy in distinguishing fraudulent from non-fraudulent transactions. ROC-AUC (Receiver Operating Characteristic - Area Under Curve) assesses the trade-off between the rate of true and false positives. ROC-AUC will be a measure of whether the model is functioning. The higher the ROC-AUC score, the more likely the model can identify fraudulent transactions without much bias toward false positives. Confusion Matrix will provide insights into the model's performance by showing the distribution of true positives, true negatives, false positives, and false negatives in detail. It helps to understand how good the model is at classifying the transactions and where it may make an error. Specifically, we have put more effort into lessening the rate of false positives, as this was essential when lowering alert fatigue among the employees while, at the same time, remaining effective in the detection of fraud. Using these evaluation metrics, we showed evidence of quantifiable improvement in the baseline model and the effectiveness of the applied techniques.

B. Baseline model results

The data was split into a train and test set where the percentage of fraud is the same in the test set as in the train set. The baseline model achieved an ROC-AUC score of 0.928, and a confusion matrix with a false positive rate of 41%. The ROC-AUC score shows that 93% of the model's classification ability could be seen in distinguishing fraudulent from non-fraudulent transactions. There is a sidenote and that is that there is a gap between the performance of the baseline model in this research and the original Kaggle competition baseline model which has a AUC-ROC of 0.945. This can be attributed to some factors:

- Dataset seed: The kaggle competition did not have a dataset split seed which resulted in a having a different split then the kaggle competition.
- Computational Resources: The contestants had access to more resources and time. Which means they could train their model extensively.

C. Implemented techniques

The techniques to improve the performance of the AI model were tested separately.

1) Sampling: As discussed previously, we will implement an oversampling technique. A study clearly showed that SMOTE would not improve our model and reduce the false positive. However, future research has explained that ROSE could potentially improve the model's performance.

So, we implemented ROSE as an oversampling technique for the dataset. ROSE has been implemented to create a dataset with twice the number of fraud cases as can be seen in figure 7. This means instead of 3.6%, we now have 6.9% of fraud cases. After that, the models were trained again on their original parameters.

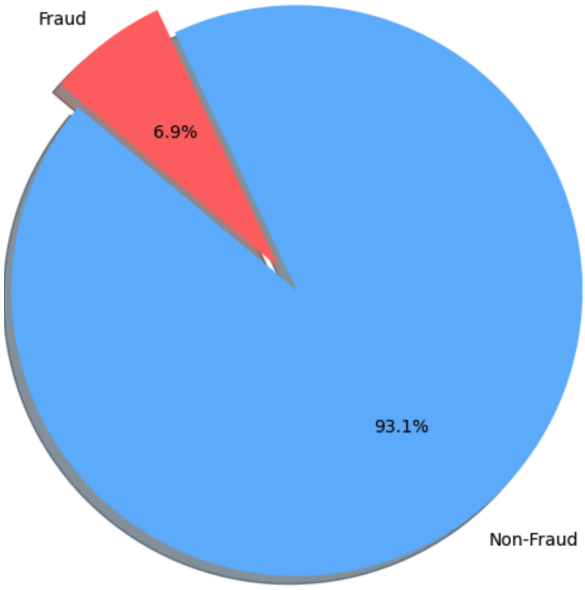
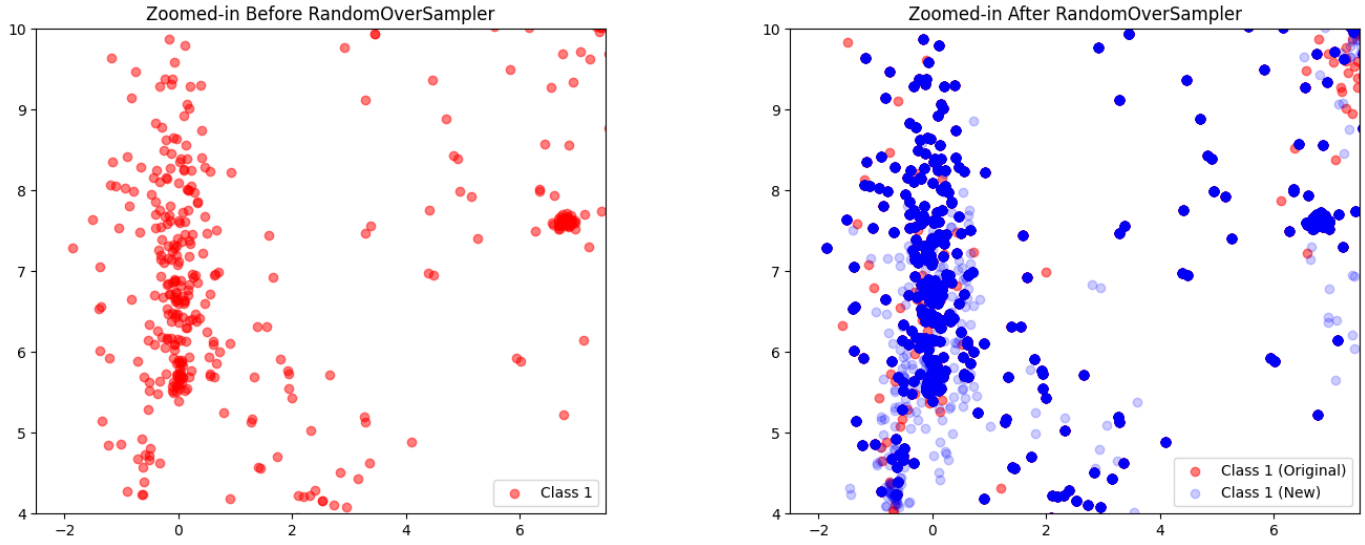


Fig. 7: Oversampling dataset

To see ROSE's impact on the data points, we created a plot of a before and after. In figure 8a, we can see a selected part of the data points plotted with the use of PCA, which is a feature reduction technique. This way, we can plot the data points on a 2d graph. In figure 8b we can see the same vector space but with the oversampling datapoints. As can be seen in the figures, a lot of extra points were created. Furthermore, we can see some randomizing in the data points. Some points are more opaque than other points. The less opaque a data point is, the more data points are in the same location. This does not necessarily mean that ROSE created 2 identical points. But this means that the features selected by the PCA have the same values. The data points that ROSE added to the dataset are making the existing clusters more dense. This can help AI learn better patterns. Furthermore, some data points have little randomization.

2) Feature engineering: As discussed in the previous chapter, we will use the feature engineering technique from [39]. From VESTA we have acquired two datasets. A train and a test dataset. These datasets did not have an UID. So we created our UID as mentioned before and with this UID we could create the new feature. The new feature looks at a user's past transactions. We tried it with 24hours, 60hours an 168hours in the past as mentioned in the paper of (Bahnsen, et al) [39].

3) Comparison of the techniques : Three different methods are considered: regularization, oversampling, and feature engineering. Table VI shows the test results of 6 iterations since all the parts of the AI model were tested separately. In the table, we can see iteration one. This is the baseline. Every part of the architecture has their own



(a) Zoomed-in Before RandomOverSampler

(b) Zoomed-in After RandomOverSampler

Fig. 8: Visualization of credit card fraud data before and after applying RandomOverSampler. Class 1 data points are highlighted in red, with newly added points shown in blue with lower opacity.

Iteration	Model	AUC-ROC	False positive rate	Hours	Regularization	Sampling
1	XGBoost	0.932	21	-	No	No
1	CATBOOST	0.940	22	-	No	No
1	LGBM/Architecture	0.928	41	-	No	No
2	XGBoost	0.938	22	-	Yes	No
2	CATBOOST	0.943	20	-	Yes	No
2	LGBM/Architecture	0.932	40	-	Yes	No
3	XGBoost	0.921	17	24	Yes	No
3	CATBOOST	0.843	21	24	Yes	No
3	LGBM/Architecture	0.922	53	24	Yes	No
4	**XGBoost**	**0.935**	**15**	**60**	**Yes**	**No**
4	CATBOOST	0.871	21	60	Yes	No
4	LGBM/Architecture	0.949	39	60	Yes	No
5	XGBoost	0.954	18	168	Yes	No
5	CATBOOST	0.872	20	168	Yes	No
5	LGBM/Architecture	0.921	37	168	Yes	No
6	XGBoost	0.964	18	-	Yes	Yes
6	CATBOOST	0.931	18	-	Yes	Yes
6	LGBM/Architecture	0.935	40	-	Yes	Yes

TABLE VI: Labbook

measurement of the ROC-AUC and false positive rate. The third and fourth columns represent the metric scores, and the fifth column is the parameter used in the feature engineering technique. Columns six and seven detail whether regularization or sampling is used.

Iteration one is considered the baseline; however, the LGBM/Architecture's false positive rate performed as one of the worst. Of course, we do have to consider that the authors did not use this metric but only the ROC-AUC. When adding the different techniques, we saw no significant decrease in the LGBM/Architecture false positives. But when we start looking at the XGboost model or CATBoost model separately. We can see some different improvements. These two models showed a lower false positive rate in every iteration compared to the LGBM/Architecture. While sampling in iteration six showed an improvement in false positives, it does show a lower ROC-AUC. However, the best working model is the XGBoost model in iteration four, which is also bolted. This model has a ROC-AUC of 0.935 and a false positive rate of 15%. This model shows a decreased false positive rate and an improved ROC-AUC. The ROC-AUC can be seen in figure 9 and the confusion matrix can be seen in table VII. The parameters used for the XGBoost model come from the research of (Priscilla, et al) [43]. With a regularization of 0,5. We could not implement gridsearch to find the best parameters as this took too long. The expected time when the gridsearch was complete was 56 hours.

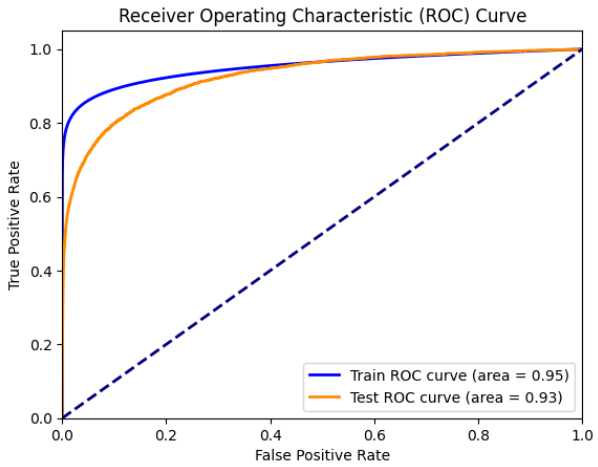


Fig. 9: XGBoost ROC-AUC

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP): 142216	False Negative (FN): 319
Actual Negative	False Positive (FP): 3051	True Negative (TN): 2049

TABLE VII: Confusion matrix XGBoost model

D. Requirements

Not all the requirements have been discussed or even researched. The most important requirements were 1, 2, 4, 5, 6, 7, and 13. One of the key features of requirement one is using the confusion matrix and the ROC-AUC as key metrics. The confusion matrix is useful for visualizing the performance of an algorithm in showing true positives, true negatives, false positives, and false negatives—with a particular focus on minimizing the false positive rate. The metric, ROC-AUC, helps understand the performance of the overall model. Requirement two is an improvement to an existing advanced architecture instead of developing from scratch; it was taken as time-efficient and reproducible. Applying a strong baseline model maximized not only the use of resources but also the use of time to target reducing alert fatigue while maintaining the overall accuracy of the baseline. Requirement 7 sets the performance benchmarks equal or above 0.93 in ROC-AUC and reduces the false positive rate to less than 41%, such that measurable improvement is realized. At the same time, the performance is maintained. Finally, requirement 13 underlined the fact that different quality criteria have to be evaluated for performance and model complexity, such that improvements were not only on raw performance but also in practical applicability and ethical considerations. Explainability guaranteed that the decisions taken by the model were transparent and justifiable; on the other hand, model complexity ensured the model was scalable and maintainable. We made the model less complex by instead of using three different AI models to check for fraud. Financial institutions can now use one model for detecting fraud. And that one model performs better than the previously made architecture. By reducing the complexity, we made it easier to understand and maintain.

Therefore, considering the different technical and ethical aspects, we find AI suitable for credit card fraud detection. However, the implementation should be carefully managed to ensure its benefits are maximized and potential risks are minimized. Like having bias towards a certain group.

V. Discussion

The first objective of this study aimed to reduce alert fatigue in the credit card fraud detection system by improving the already existing AI model. Our approach was based on the CRISP-MLQ methodology that structured the research process in terms of business understanding, data understanding, modeling and evaluation.

A. Summary of Findings

We noted performance improvement in the AI model after implementing and testing techniques like sampling and feature engineering. The baseline model from a winning Kaggle competition resulted in an initial ROC-AUC of 0.928, with a false positive rate of 41%. The false positive rate was reduced and the ROC-AUC improved in some iterations. The most significant results were seen when we implemented feature

engineering and regularization. The best model iteration, which used feature engineering with a 60-hour parameter and regularization, realized a ROC-AUC of 0.935, with a false positive rate reduced by 15%.

B. Implications

This reduction in the number of false positives has great implications for credit card companies and their customers. Through this reduction, more transactions will be marked genuine, increasing customer satisfaction. Further, a lower false positive rate will increase the effectiveness of fraud detection teams, which can now focus on really suspicious transactions, thus reducing alert fatigue and being more productive. This further justifies the improvement in the performance of the AI model through advanced techniques, such as ROSE, coupled with sophisticated feature engineering to solve the problem at hand. This finding is important for AI research in credit card fraud because it offers practical solutions for reducing the false positive rate.

C. Limitations

However, this study has several limitations. First and foremost, not having a user ID in the dataset significantly diminished the potential for feature engineering further, most of which relies on determining the patterns of user behavior. In addition, the time that we had for this research and the computational resources might have limited the training and optimization of the models. The research also narrowly centered on a specific subset of techniques and did not investigate other potential methods, such as deep learning models or hybrid approaches that merge supervised and unsupervised learning. Or the use of different regularization techniques like L2. Finally, the dataset used a normalization technique. Which meant that the data was anonymous. However, this also means that the bias was not tested or researched. It could be that certain features regarding race, gender, or location are the most prominent features in the AI model

D. Future Research

Future research will have to be explored in several respects to build upon the results and findings of this study. It would be helpful first if it were possible to access a dataset including user IDs, as this data would permit the usage of more advanced feature engineering techniques. Furthermore, the use of the false positive metric should be further studied and could potentially lead to using a different metric. This is because alert fatigue is something that happens in the human brain and every human has their own threshold for showing alert fatigue. Knowing where the average threshold is or if there are other better metrics for measuring alert fatigue would lead to a better approach for solving alert fatigue. Additionally, further research should focus on making models more explainable to make AI-driven fraud detection systems transparent and trusted. For instance, it could be a study aimed at optimizing strategies for reducing alert fatigue related to prioritization and customization of alerts, as this also helps reduce alert fatigue. Testing such strategies with financial institutions in a live environment will likely give many insights, leading to practical recommendations for improvement in alert management systems. Lastly the biggest limitation is that bias could not have been measured. This is because of the anonymization of the data. It is important to make sure an AI model does not contain bias. So further research into seeing if there is a bias and solving it would benefit the model ethics.

E. Conclusion

The study represents the potential impact of AI techniques on alert fatigue in the credit card fraud detection system. While model accuracy increases and false-positive decreases, this boosts the fraud detection team's efficiency and customer satisfaction. With some limitations and several areas to further explore, the findings set a good ground for future research and practical applications in the financial sector. The subquestions stated in the introduction are both answered. Using the false positive rate, you can measure alert fatigue with an AI model. You can also reduce alert fatigue by reducing the false positive rate. Techniques that reduce false positives include regularization and creating a spending pattern for a client.

REFERENCES

[1] K. Chaudhary, J. Yadav, and B. Mallick, "A review of fraud detection techniques: Credit card," *International Journal of Computer Applications*, vol. 45, no. 1, pp. 39–44, 2012.

[2] D. Wang, B. Chen, and J. Chen, "Credit card fraud detection strategies with consumer incentives," *Omega*, vol. 88, pp. 179–195, 2019.

[3] J. S. Ancker, A. Edwards, S. Nosal, D. Hauser, E. Mauer, R. Kaushal, and W. the HITEC Investigators, "Effects of workload, work complexity, and repeated alerts on alert fatigue in a clinical decision support system," *BMC medical informatics and decision making*, vol. 17, pp. 1–9, 2017.

[4] S. Ghosh and D. L. Reilly, "Credit card fraud detection with a neural-network," in *System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on*, vol. 3. IEEE, 1994, pp. 621–630.

[5] F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan, and M. Ahmed, "Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms," *IEEE Access*, vol. 10, pp. 39 700–39 715, 2022.

[6] M. F. A. Gadi, X. Wang, and A. P. do Lago, "Credit card fraud detection with artificial immune system," in *International conference on artificial immune systems*. Springer, 2008, pp. 119–131.

[7] C. Whitrow, D. J. Hand, P. Juszczak, D. Weston, and N. M. Adams, "Transaction aggregation as a strategy for credit card fraud detection," *Data mining and knowledge discovery*, vol. 18, pp. 30–55, 2009.

[8] Y. Sahin and E. Duman, "Detecting credit card fraud by decision trees and support vector machines," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, 2011, pp. 1–6.

[9] F. Itoo, Meenakshi, and S. Singh, "Comparison and analysis of logistic regression, naïve bayes and knn machine learning algorithms for credit card fraud detection," *International Journal of Information Technology*, vol. 13, no. 4, pp. 1503–1511, 2021.

[10] A. S. Kesselheim, K. Cresswell, S. Phansalkar, D. W. Bates, and A. Sheikh, "Clinical decision support systems could be modified to reduce 'alert fatigue'while still minimizing the risk of litigation," *Health affairs*, vol. 30, no. 12, pp. 2310–2317, 2011.

[11] S. L. Kane-Gill, M. F. O'Connor, J. M. Rothschild, N. M. Selby, B. McLean, C. P. Bonafide, M. M. Cvach, X. Hu, A. Konkani, M. M. Pelter *et al.*, "Technologic distractions (part 1): summary of approaches to manage alert quantity with intent to reduce alert fatigue and suggestions for alert fatigue metrics," *Critical care medicine*, vol. 45, no. 9, pp. 1481–1488, 2017.

[12] A. Srivastava, A. Kundu, S. Sural, and A. Majumdar, "Credit card fraud detection using hidden markov model," *IEEE Transactions on dependable and secure computing*, vol. 5, no. 1, pp. 37–48, 2008.

[13] A. Kore, *Designing Human Centric AI Experiences*. Springer, 2022.

[14] "The crispmlq method." [Online]. Available: https://ml-ops.org/content/crisp-ml

[15] "What is fraud." [Online]. Available: https://www.acfe.com/fraud-resources/fraud-101-what-is-fraud

[16] P. Dolfen, L. Einav, P. J. Klenow, B. Klopack, J. D. Levin, L. Levin, and W. Best, "Assessing the gains from e-commerce," *American Economic Journal: Macroeconomics*, vol. 15, no. 1, pp. 342–370, 2023.

[17] "The process of the credit card payment." [Online]. Available: https://www.uschamber.com/co/run/finance/guide-to-credit-card-processing#:~:text=The%20credit%20card%20network%20sends,the%20credit%20card%20processing%20company

[18] "Laws of credit card fraud in the eu." [Online]. Available: https://www.dnb.nl/en/reliable-financial-sector/combating-money-laundering-and-fraud/

[19] "Laws of credit card fraud in the usa." [Online]. Available: https://www.law.cornell.edu/wex/credit_card_fraud

[20] "The process of credit card fraud." [Online]. Available: https://cybeready.com/category/comprehensive-guide-to-fraud-detection-management-and-analysis

[21] P. Voigt and A. Von dem Bussche, "The eu general data protection regulation (gdpr)," *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, vol. 10, no. 3152676, pp. 10–5555, 2017.

[22] T. Fawcett and F. Provost, "Adaptive fraud detection," *Data mining and knowledge discovery*, vol. 1, no. 3, pp. 291–316, 1997.

[23] E. Aleskerov, B. Freisleben, and B. Rao, "Cardwatch: A neural network based database mining system for credit card fraud detection," in *Proceedings of the IEEE/IAFE 1997 computational intelligence for financial engineering (CIFEr)*. IEEE, 1997, pp. 220–226.

[24] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: a realistic modeling and a novel learning strategy," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 8, pp. 3784–3797, 2017.

[25] R. J. Bolton and D. J. Hand, "Statistical fraud detection: A review," *Statistical science*, vol. 17, no. 3, pp. 235–255, 2002.

[26] B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," *Advances in neural information processing systems*, vol. 12, 1999.

[27] A. Pumsirirat and Y. Liu, "Credit card fraud detection using deep learning based on auto-encoder and restricted boltzmann machine," *International Journal of advanced computer science and applications*, vol. 9, no. 1, 2018.

[28] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection," *Information Sciences*, vol. 479, pp. 448–455, 2019.

[29] F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, "Combining unsupervised and supervised learning in credit card fraud detection," *Information sciences*, vol. 557, pp. 317–331, 2021.

[30] R. Sarno, R. D. Dewandono, T. Ahmad, M. F. Naufal, and F. Sinaga, "Hybrid association rule learning and process mining for fraud detection." *IAENG International Journal of Computer Science*, vol. 42, no. 2, 2015.

[31] S. Kamaruddin and V. Ravi, "Credit card fraud detection using big data analytics: use of psoaann based one-class classification," in *Proceedings of the international conference on informatics and analytics*, 2016, pp. 1–8.

[32] M. I. Hussain, T. L. Reynolds, and K. Zheng, "Medication safety alert fatigue may be reduced via interaction design and clinical role tailoring: a systematic review," *Journal of the American Medical Informatics Association*, vol. 26, no. 10, pp. 1141–1149, 2019.

[33] E. F. Malik, K. W. Khaw, B. Belaton, W. P. Wong, and X. Chew, "Credit card fraud detection using a new hybrid machine learning architecture," *Mathematics*, vol. 10, no. 9, p. 1480, 2022.

[34] C. Jiang, J. Song, G. Liu, L. Zheng, and W. Luan, "Credit card fraud detection: A novel approach using aggregation strategy and feedback mechanism," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3637–3647, 2018.

[35] Cdeotte, "Xgb fraud with magic - [0.9600]," Feb 2020. [Online]. Available: https://www.kaggle.com/cdeotte/xgb-fraud-with-magic-0-9600

[36] "Kaggle competition," 2019. [Online]. Available: https://www.kaggle.com/competitions/ieee-fraud-detection/overview

[37] "Kaggle score page," 2019. [Online]. Available: https://www.kaggle.com/competitions/ieee-fraud-detection/leaderboard

[38] "A european credit card dataset," 2013. [Online]. Available: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

[39] A. C. Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, "Feature engineering strategies for credit card fraud detection," *Expert Systems with Applications*, vol. 51, pp. 134–142, 2016.

[40] H. Meyer, C. Reudenbach, T. Hengl, M. Katurji, and T. Nauss, "Improving performance of spatio-temporal machine learning models using forward feature selection and target-oriented validation," *Environmental Modelling & Software*, vol. 101, pp. 1–9, 2018.

[41] A. Altmann, L. Toloşi, O. Sander, and T. Lengauer, "Permutation importance: a corrected feature importance measure," *Bioinformatics*, vol. 26, no. 10, pp. 1340–1347, 2010.

[42] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[43] 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)*. IEEE, 2020, pp. 1309–1315.

[44] S. Kotsiantis, D. Kanellopoulos, P. Pintelas *et al.*, "Handling imbalanced datasets: A review," *GESTS international transactions on computer science and engineering*, vol. 30, no. 1, pp. 25–36, 2006.

[45] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, "Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary," *Journal of artificial intelligence research*, vol. 61, pp. 863–905, 2018.

[46] N. Lunardon, G. Menardi, and N. Torelli, "Rose: a package for binary imbalanced learning." *R journal*, vol. 6, no. 1, 2014.

[47] R. Blagus and L. Lusa, "Smote for high-dimensional class-imbalanced data," *BMC bioinformatics*, vol. 14, pp. 1–16, 2013.

[48] B. Das, N. C. Krishnan, and D. J. Cook, "Racog and wracog: Two probabilistic oversampling techniques," *IEEE transactions on knowledge and data engineering*, vol. 27, no. 1, pp. 222–234, 2014.

[49] H. Liu, K. Roeder, and L. Wasserman, "Stability approach to regularization selection (stars) for high dimensional graphical models," *Advances in neural information processing systems*, vol. 23, 2010.

[50] Z. Xu, X. Chang, F. Xu, and H. Zhang, "$l\_\{1/2\}$ regularization: A thresholding representation theory and a fast solver," *IEEE Transactions on neural networks and learning systems*, vol. 23, no. 7, pp. 1013–1027, 2012.

[51] D. Vidaurre, C. Bielza, and P. Larranaga, "A survey of l1 regression," *International Statistical Review*, vol. 81, no. 3, pp. 361–387, 2013.

[52] R. Punmiya and S. Choe, "Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing," *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 2326–2329, 2019.

[53] Q. Zou, S. Xie, Z. Lin, M. Wu, and Y. Ju, "Finding the best classification threshold in imbalanced classification," *Big Data Research*, vol. 5, pp. 2–8, 2016.