# The Impact of Subsampling on Differentially Private Fraud Detection

**Storm van Wassenaar**[1]
**Supervisor: Dr. Zeki Erkin**[1]
[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 21, 2026

Name of the student: Storm van Wassenaar
Final project course: CSE3000 Research Project
Thesis committee: Dr. Zeki Erkin, Dr. Merve Gürel

**Abstract**

Fraud detection is a critical task, but detecting fraud can be challenging due to class imbalance. Furthermore, the availability of the data is also limited, because the data is very sensitive and cannot be shared between financial institutions due to privacy regulations. One way to address this is by applying differential privacy. Differential privacy is a mechanism that applies controlled noise to the data to achieve a certain privacy guarantee. However the added noise may also affects the utility of the data.

Privacy amplification techniques, such as subsampling, have been introduced to increase the privacy guarantee without directly adding additional noise. This paper investigates how privacy amplification by subsampling affects the privacy-utility trade-off in differentially private fraud detection.

To answer this question, an experiment is performed in which logistic regression models are trained using different subsampling rates and privacy budgets.

The results show that subsampling can improve the performance of a model in highly private settings. However, this improvement is primarily in distinguishing between fraudulent and legitimate transactions, rather than detecting more fraudulent transactions. Therefore, whether subsampling is effective depends on the application and the costs of false positives and false negatives.

# 1 Introduction

In 2025, reported fraud losses in America reached $15 billion [1], which is 2 billion higher than the previous year. Fraud detection can be challenging, because the activities can be rare and thus hard to detect. Not only is the activity rare, but the availability of the data is also a problem. Collaboration between financial institutions is not straightforward, because they are dealing with privacy sensitive information and sharing this information is not allowed due to privacy regulations. To tackle this problem privacy-preserving techniques have been proposed, such as homomorphic encryption [2] and differential privacy [3]. While homomorphic encryption enables computations on encrypted data it can be very computational expensive. Differential privacy on the other hand uses a different approach. Differential privacy applies noise to data to achieve a certain privacy guarantee. By adding this noise you are also affecting the quality of your data. The added noise can make the detection of the rare events even harder. To deal with this, multiple privacy amplification techniques have been introduced. These techniques improve the privacy guarantee without directly adding additional noise. This paper focuses on how privacy amplification by subsampling [4] affects the privacy-utility trade-off in differentially private fraud detection. It investigates whether the use of subsampling can increase the utility while maintaining the same privacy guarantee.

The remainder of this paper is structured as follows. Section 2 will explain preliminaries needed for this paper. Section 3 will show related work on this topic. Section 4 describes the methods used for this experiment. Section 5 will analyse the results obtained from the experiments. Section 6 discusses the ethical aspects of the research and Section 7 outlines the limitations of the research. Finally, Section 8 concludes the paper and outlines future work.

# 2 Preliminaries

This section explains the main concepts of Differential Privacy and subsampling that are necessary for this paper.

### 2.1 Differential Privacy

Differential Privacy [3] is a framework for protecting the privacy of individuals in a dataset. It provides a privacy guarantee that the output does not reveal whether an individual was included in the dataset.

Formally, a randomised mechanism $\mathcal{M}$ satisfies ($\varepsilon$, $\delta$)-differential privacy if for neighbouring datasets d and d', where neighbouring datasets differ by a single entry, and for any subset of outputs $S \subseteq Range(\mathcal{M})$ the following holds:

$$\Pr[\mathcal{M}(d) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(d') \in S] + \delta \tag{1}$$

To achieve this, noise is added to the data. The mechanisms used for this are dependent on the variables $\varepsilon$ and $\delta$. $\varepsilon$ is the privacy budget, which determines the level of privacy and thus the amount of noise added to the dataset. A smaller $\varepsilon$ means a higher privacy guarantee and thus also more noise. $\delta$ is introduced by relaxed versions of differential privacy. This variable $\delta$ represents the probability that the privacy guarantee fails to hold. In practice this variable is usually chosen to be very small.

The amount of noise does not only depend on $\varepsilon$, it also depends on the sensitivity of the function. This value is the maximum the output of the function can change when an individual is removed or added. In the case of the Gaussian mechanism the sensitivity is defined as $\Delta_2 f$.

### 2.2 Privacy Amplification

The results of the applied differential privacy can be amplified. This means that the privacy guarantee that was achieved by applying differential privacy can be strengthened. Multiple methods to do this have been proposed, like for example, Shuffling [5] and Subsampling [4]. In both cases, an additional level of uncertainty is introduced. For subsampling each client has a probability to be used for a computation. This means that it is also possible for a client not to be used, which introduces uncertainty that improves the privacy guarantee. Since there are multiple subsampling methods with different effects on the data they also have a different effect on $\varepsilon$ and $\delta$.

However, in this paper, we mostly focus on Poisson subsampling. With Poisson subsampling a data point is included with probability $\gamma$. A data point is included if a randomly drawn number between 0 and 1 is smaller than the chosen $\gamma$. With the following theorem from [4] we can calculate the effective $\varepsilon$ with subsampling:

Let $\mathcal{M}' = \mathcal{M}^{S^{po}_{\gamma}}$. For any $\varepsilon \geq 0$ we have $\delta_{\mathcal{M}'}(\varepsilon') \leq \gamma \delta_{\mathcal{M}}(\varepsilon)$, where $\varepsilon' = \log(1+\gamma(e^{\varepsilon}-1))$.

This theorem states that if you have a randomised mechanism $\mathcal{M}$ and you perform this on a Poisson subsample you will have $\mathcal{M}$'. This shows that when you perform a differentially private mechanism on a Poisson subsample this results in an improved privacy guarantee, since the $\varepsilon'$ is reduced.

## 3 Related Work

Fraud detection with machine learning is well studied. There have been multiple papers analysing different machine learning methods for fraud detection and improving the models with different pre-processing techniques. As done in [6], where they analysed the performance of different machine learning models and also applied preprocessing techniques like

feature selection and SMOTE. However, this does not yet considers the privacy of the sensitive data.

The application of differential privacy on fraud detection has been explored in different settings. [7] evaluates the performance in a more isolated setting. Where multiple machine learning models and differential privacy mechanisms are compared with each other.

Another approach is federated learning. A method commonly used in a federated setting is DP-SGD [8]. This method performs stochastic gradient descent while adding noise to the gradients. In DP-SGD, subsampling is also used, but its impact has not been discussed in detail.

# 4 Methods

This section explains how the experiment was performed. The experiment consisted of the following components:

1. Dataset
2. Subsampling
3. Model training
4. Differential privacy (Perturbation)
5. Evaluation

Each component is explained individually in the following subsections.

## 4.1 Dataset

This experiment uses the "Credit Card Fraud Detection" dataset introduced by the Machine Learning Group at ULB [9] and available on Kaggle [10].

This dataset contains 284,807 transactions, of which 492 are fraudulent, with only 0.172% of transactions being fraudulent. This dataset is constructed based on credit card transactions made in September 2013 by European cardholders.

Due to confidentiality issues, most features have been transformed using PCA. The features are transformed into the features V1 through V28. The only features that have not been transformed are "Time" and "Amount". In this experiment, all features except "Time" are used and no additional preprocessing is applied.

## 4.2 Subsampling

In this step, the theorem of privacy amplification by subsampling [4] is applied. Specifically, Poisson subsampling is used, as described by the following theorem:

Let $\mathcal{M}' = \mathcal{M}^{S^{po}_{\gamma}}$. For any $\varepsilon \geq 0$ we have $\delta_{\mathcal{M}'}(\varepsilon') \leq \gamma\delta_{\mathcal{M}}(\varepsilon)$, where $\varepsilon' = \log(1+\gamma(e^{\varepsilon}-1))$.

During this experiment, different subsampling rates ($\gamma$) are used, specifically $\gamma \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$. Here 1.0 serves as a baseline, since no subsampling is performed and the effective epsilon will not be changed. For each $\gamma$ the effective epsilon ($\varepsilon'$) has been kept constant, which means that the original $\varepsilon$ is adapted to the $\gamma$ that is used. This can easily be calculated by rearranging the previous formula:

$$\varepsilon = \log(\frac{e^{\varepsilon'} - 1}{\gamma} + 1) \tag{2}$$

For example, when $\gamma = 0.5$ and $\varepsilon' = 1.0$, the $\varepsilon$ used during the perturbation step can be computed as:

$$\varepsilon = \log(\frac{e^{1.0} - 1}{0.5} + 1) \approx 1.590 \tag{3}$$

Thus, when a $\gamma$ of 0.5 is used, an $\varepsilon$ of 1.590 in the perturbation step is sufficient to achieve a privacy guarantee of $\varepsilon' = 1.0$.

### 4.3 Model Training

The machine learning model used for this experiment is logistic regression. Logistic regression has been chosen due to its simplicity and its popularity in fraud detection. Although more complex machine learning models may perform better, the goal of this experiment is not to create the most optimal model, but to study the effect of subsampling on the privacy-utility trade-off.

Logistic regression is a supervised machine learning algorithm that can be used for classification. In this case, it predicts the class to which a transaction belongs. The parameters of the model are set based on which combinations minimizes a loss function based on the given training data.

For this experiment, a regularized version of logistic regression is used. Regularizations is used to bound the parameters of the model, which can help with for example overfitting. It is also important for the perturbation step in this experiment. Bounding the parameters also bounds the sensitivity, which is necessary for adding noise in the next step.

For this experiment, the L2-regularized logistic regression implementation provided by the scikit-learn library is used.

### 4.4 Differential privacy (Perturbation)

To achieve $(\varepsilon, \delta)$-Differential Privacy, noise is added. Several perturbation mechanisms have been proposed to achieve this guarantee. Two commonly used mechanisms are objective and output perturbation [11].

- Objective perturbation adds noise to the objective function.
- Output perturbation adds noise to the parameters.

For this experiment, output perturbation is used. This mechanism is chosen to be able to use existing logistic regression implementations. For adding noise to the parameters, the Gaussian mechanism [12] is used. This mechanism looks like:

$$\mathcal{M}(x) = f(x) + \mathcal{N}(0, \sigma^2) \tag{4}$$

$$\sigma \geq \frac{\sqrt{2\ln(\frac{1.25}{\delta})}\Delta_2 f}{\varepsilon} \tag{5}$$

Where $\Delta_2 f$ is the sensitivity, which was bounded in the training step. To get the sensitivity value we can use the following formula

$$\Delta_2 f = \frac{2}{n\lambda} \tag{6}$$

However, the scikit-learn implementation uses a $C$ instead of $\lambda$, so the lambda can be computed as $\lambda = \frac{1}{C}$.

### 4.5 Evaluation

To measure the performance of a model, there exist multiple metrics, but since the dataset is highly imbalanced some of them may not be very informative. For example, accuracy might be misleading, since it is defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

Even with 99% accuracy, the model may still only predict a single class correctly.

Instead, the following metrics are used:

- AUPRC: This metric measures how well fraudulent transactions are detected while minimizing the number of false positives.
- Sensitivity: This metric measures how well fraudulent transaction are correctly predicted.
- Specificity: This metric measures how well legitimate transaction are correctly predicted.

The use of both sensitivity and specificity is important, because both false positives and false negatives can be costly.

## 5 Result analysis

This section presents the results of the experiment described in the previous section. The experiment was performed with 400 repetitions. For each metric, the mean difference between the baseline and the subsampled models has been plotted with their 95% confidence interval. In each plot, the solid lines represent the mean and the dashed lines represent the confidence interval. The notebook used for the experiments is available on GitHub [13].

The analysis begins with the AUPRC results, followed by specificity and sensitivity.

### 5.1 AUPRC

The Area Under the Precision-Recall Curve (AUPRC) is used as the primary performance metric due to class imbalance. A higher AUPRC value indicates that the model is better at identifying fraudulent transactions while maintaining precision.

Figure 1 shows the difference in AUPRC between the baseline and subsampled models. From the figure, it appears that many subsampling rates are outperforming the baseline. However, the confidence intervals show that these differences are sometimes not significant.

For $\gamma = 0.5$, the differences at $\varepsilon = 0.75$ and $\varepsilon = 1.0$ are significant, as their confidence intervals lie entirely above zero. The same holds for $\gamma = 0.9$ at $\varepsilon = 0.75$, $\varepsilon = 1.0$ and $\varepsilon = 10.0$. The last one is barely above zero with a confidence interval of $[0.0002, 0.0027]$.

As $\varepsilon$ increases, the difference between the models becomes more stable. Additionally, larger values of $\gamma$ appear to be closer to zero. However, there are still cases where the confidence intervals overlap, so this is not certain.

A possible explanation is that subsampling reduces the size of the training set, which may negatively affect performance. For smaller values of $\varepsilon$, the loss that gets introduced by differential privacy may have a larger impact than that introduced by subsampling. As a result, some subsampling rates are able to outperform the baseline for lower values of $\varepsilon$.

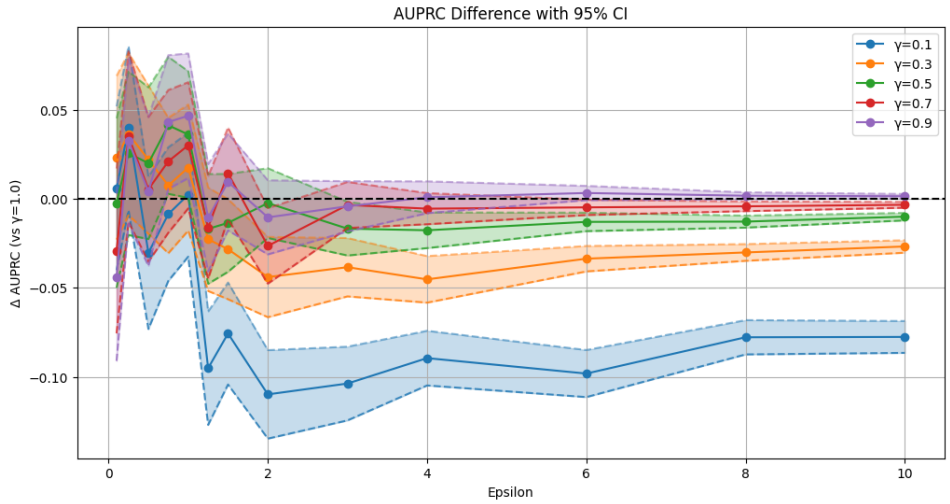


Figure 1: AUPRC Difference with 95% CI $0 \leq \varepsilon \leq 10$

## 5.2 Specificity

Specificity measures the proportion of non-fraudulent transactions that are correctly classified.

Figure 2 shows that the difference is generally higher for small values of $\varepsilon$. For small values of $\varepsilon$, several subsampling rates appear to be above the baseline indicating an improvement of specificity. For $\gamma = 0.1$, $\gamma = 0.3$ and $\gamma = 0.5$ both the mean and its confidence interval lie above zero for the majority of $\varepsilon < 1.0$, indicating an improvement in correctly classifying non-fraudulent transactions.

For $\gamma = 0.9$ it appears to be the only one clearly below zero. For $\varepsilon = 0.1$ the mean and most values are below zero. For higher values of $\varepsilon$, the difference for $\gamma = 0.9$ get closer to zero and sometimes above zero.

Overall, subsampling seems to have a positive impact on the specificity for small privacy budgets. However, for $\gamma = 0.9$ it is a bit different, as the difference is for the majority below zero for $\varepsilon = 0.1$. When $\varepsilon > 1.0$, the effect of subsampling becomes very small and the difference becomes almost zero.

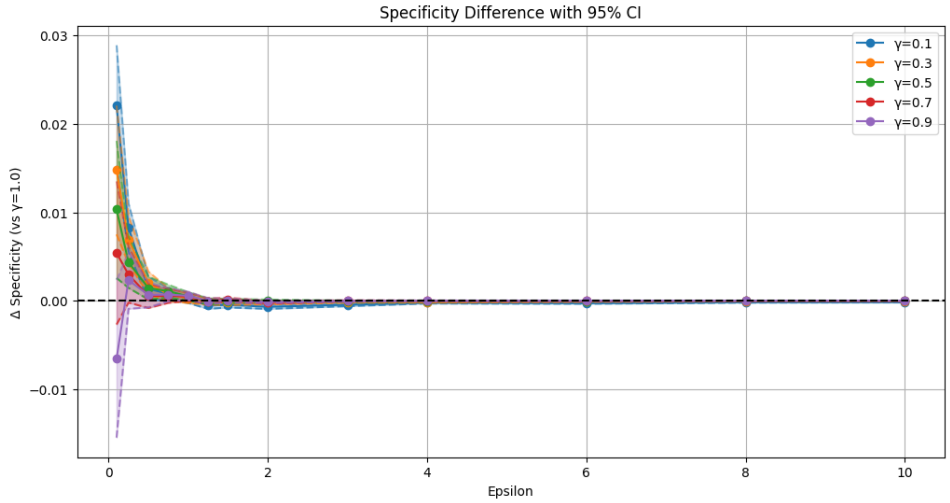


Figure 2: Specificity Difference with 95% CI $0 \leq \varepsilon \leq 10$

### 5.3 Sensitivity

Sensitivity measures the proportion of fraudulent transactions that are correctly classified.

Figure 3 shows the difference in sensitivity between the subsampled models and the baseline model. For most values of $\varepsilon$, the differences for all subsampling rates are below zero, indicating that subsampling negatively affects sensitivity.

Only at $\varepsilon = 0.1$ is the effect less clear. For this value of $\varepsilon$, only $\gamma = 0.1$ has both its mean difference and confidence interval below zero. For $\gamma = 0.5$ and $\gamma = 0.9$, their mean differences are above zero, but their confidence intervals still include zero, so we cannot conclude that there is an improvement. For the remaining subsampling rates, their mean differences are below zero, but their confidence intervals include zero.

Although all subsampling rates are below zero for larger values of $\varepsilon$, the differences for higher values of $\gamma$ are closer to zero, indicating that the negative effect of subsampling on sensitivity is smaller for these values.

Overall, the results suggest that subsampling reduces sensitivity, which means that fewer fraudulent transactions are correctly detected compared to the baseline model.

## 6 Responsible Research

Using a Machine Learning model to predict whether transactions are fraud can have a significant impact on society and economy. While these systems can be used to catch fraud, they are not perfect and may produce false positive and negative cases. A false positive occurs when a non-fraudulent transaction is incorrectly flagged as fraudulent. It depends in which situation the model is used, but in some cases it can therefore be beneficial to let people manually check positive cases to filter these false positives out. Since incorrectly accusing someone of fraud can have a big societal impact. A False negative occurs when a fraudulent transaction is not detected. This comes with a financial loss. However this is hard to double check with manual work.

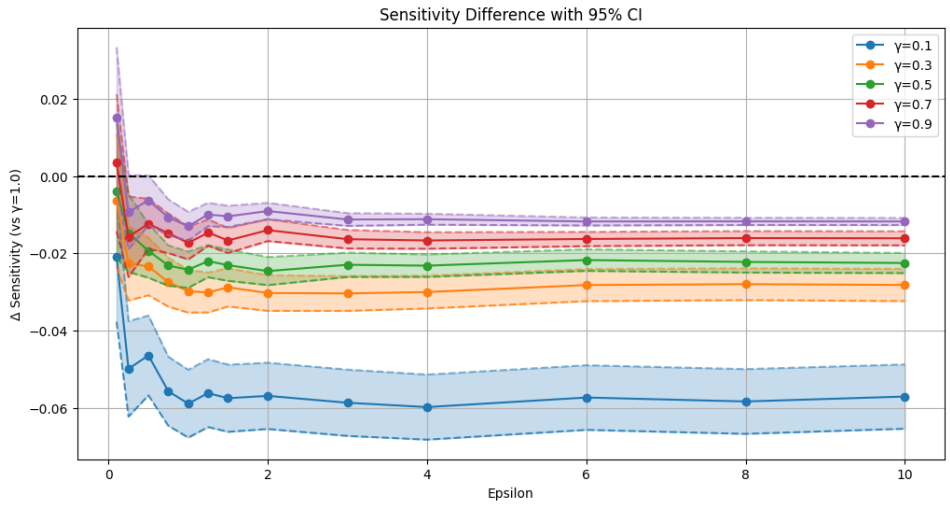


Figure 3: Sensitivity Difference with 95% CI $0 \leq \varepsilon \leq 10$

# 7 Discussion

The results show that subsampling can improve the AUPRC for small values of $\varepsilon$. However, this improvement is not reflected in the sensitivity and specificity. Specificity remains similar to the baseline, while the sensitivity decreased slightly. This suggests that the number of transactions that were correctly labelled as fraud decreased. The improvement in AUPRC would suggests that models are more selective. A possible explanation could be that the noise introduced by differential privacy may have a larger impact on the performance than the training set reduction caused by subsampling. As $\varepsilon$ increases and less noise is added, this effect disappears.

Whether subsampling has a positive effect depends on the application and the costs of false positives and false negatives. For example, when using fraud detection in an automated way, one could argue to minimize the number of false positives, since there is no manual work to filter these out. In such a situation, subsampling may help, since it appears to be better at distinguishing fraudulent and non-fraudulent transactions.

This paper has some limitations. This experiment was performed on a highly imbalanced dataset without any additional techniques to address this. Adding these techniques might change the effect of subsampling. Additionally, this experiment still considered a single institution. To be able to let institutions work together, a more federated learning setting would be more suitable and the effect of subsampling in such a setting may differ. Finally, only logistic regression has been used and the effect of subsampling might be different for other machine learning methods.

# 8 Conclusions and Future Work

This paper investigated how privacy amplification by subsampling affects the privacy-utility trade-off in differentially private fraud detection. To answer this question, an experiment was performed in which multiple logistic regression models were trained and perturbed with

output perturbation. These models were trained with multiple values of $\gamma$ and $\varepsilon$ on a realistic credit card fraud dataset.

The results indicate that subsampling can improve the performance for small privacy budgets. Based on the AUPRC results, subsampling rates of $\gamma = 0.5$ and $\gamma = 0.9$ outperform the baseline for multiple values of $\varepsilon$. However, this improvement is not visible in the other plots. For specificity, the differences are close to zero, while the differences of sensitivity are slightly smaller than zero. This suggests that the models become more selective in predicting fraudulent transactions. Although fewer fraudulent transactions are detected, the model is better at distinguishing between fraudulent and legitimate transactions. Whether subsampling is beneficial depends on the application and the relative costs of false positives and false negatives.

For future research, it could be valuable to investigate the effect of subsampling in combination with class imbalance techniques. These techniques are commonly used in fraud detection and may influence the performance. Additionally, research can be done on the effect of subsampling in a federated learning setting. Furthermore, this paper only focused on logistic regression, however there are more machine learning models suitable for fraud detection and subsampling might have a different effect on these models.

# A Use of Generative AI

For this paper AI was used to improve grammar. Prompts used look like: "{...} Can you improve the grammar?" and "{...} How can I improve the phrasing of this sentence?" The tools used were OpenAI's ChatGPT and Claude. The output from these tools was not blindly copied. Instead they were used to identify mistakes and explore different ways of phrasing a idea.

# References


[1] Federal Trade Commission, "Fraud reports - fraud losses," https://public.tableau.com/shared/CX28495JF, 2025, accessed: 13-06-2026.

[2] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 79:1–79:35, 2018. [Online]. Available: https://doi.org/10.1145/3214303

[3] C. Dwork, "Differential privacy," in *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, ser. Lecture Notes in Computer Science, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds., vol. 4052. Springer, 2006, pp. 1–12. [Online]. Available: https://doi.org/10.1007/11787006_1

[4] B. Balle, G. Barthe, and M. Gaboardi, "Privacy amplification by subsampling: Tight analyses via couplings and divergences," *CoRR*, vol. abs/1807.01647, 2018. [Online]. Available: http://arxiv.org/abs/1807.01647

[5] V. Feldman, A. McMillan, and K. Talwar, "Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling," *CoRR*, vol. abs/2012.12803, 2020. [Online]. Available: https://arxiv.org/abs/2012.12803

[6] D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla, "Credit card fraud detection - machine learning methods," in *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, 2019, pp. 1–5.

[7] K. Fujianto, F. Lokanta, H. Lucky, and D. Suhartono, "Optimizing differential privacy for effective machine learning-based financial fraud detection," in *2025 IEEE International Conference on Data and Software Engineering (ICoDSE)*, 2025, pp. 383–388.

[8] M. Abadi, A. Chu, I. J. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," *CoRR*, vol. abs/1607.00133, 2016. [Online]. Available: http://arxiv.org/abs/1607.00133

[9] A. D. Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in *IEEE Symposium Series on Computational Intelligence, SSCI 2015, Cape Town, South Africa, December 7-10, 2015*. IEEE, 2015, pp. 159–166. [Online]. Available: https://doi.org/10.1109/SSCI.2015.33

[10] Machine Learning Group - ULB, "Credit card fraud detection," https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud, 2016, accessed: 02-06-2026.

[11] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *J. Mach. Learn. Res.*, vol. 12, pp. 1069–1109, Jul. 2011.

[12] M. Yang, T. Guo, T. Zhu, I. Tjuawinata, J. Zhao, and K. Lam, "Local differential privacy and its applications: A comprehensive survey," *Comput. Stand. Interfaces*, vol. 89, p. 103827, 2024. [Online]. Available: https://doi.org/10.1016/j.csi.2023.103827

[13] S. van Wassenaar, "The impact of subsampling on differentially private fraud detection," https://github.com/Stormvw/subsampling_fraud_detection, 2026.