Informatics

Privacy-Preserving Collaborative Anomaly Detection

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Data Science

eingereicht von

Dipl.-Ing. Ghaith Arfaoui
Matrikelnummer 01435404

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Privatdoz. Mag.rer.soc.oec. Dipl.-Ing. Dr.techn. Edgar Weippl
Mitwirkung: Univ.Lektor Mag.rer.soc.oec. Dipl.-Ing. Rudolf Mayer

Wien, 25. Jänner 2023

Ghaith Arfaoui

Edgar Weippl

TU WIEN Informatics

Privacy-Preserving Collaborative Anomaly Detection

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Data Science

by

Dipl.-Ing. Ghaith Arfaoui
Registration Number 01435404

to the Faculty of Informatics

at the TU Wien

Advisor: Privatdoz. Mag.rer.soc.oec. Dipl.-Ing. Dr.techn. Edgar Weippl
Assistance: Univ.Lektor Mag.rer.soc.oec. Dipl.-Ing. Rudolf Mayer

Vienna, 25th January, 2023

Ghaith Arfaoui

Edgar Weippl

Erklärung zur Verfassung der Arbeit

Dipl.-Ing. Ghaith Arfaoui

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 25. Jänner 2023

Ghaith Arfaoui

Acknowledgements

First, I would like to thank Privatdoz. Mag.rer.soc.oec. Dipl.-Ing. Dr.techn. Edgar Weippl for providing me with the opportunity to work on this thesis under his supervision.

I would also like to express my deepest appreciation to Univ.Lektor Mag.rer.soc.oec. Dipl.-Ing. Rudolf Mayer who guided me throughout the whole process. Without his continuous help and all the insightful discussions we had, this work would not have been possible. It was a real pleasure working with him.

Lastly, I would like to thank my family, for their continuous encouragement. I'm greatly indebted to my parents for their unwavering support and for being always there, at every step of the way.

Abstract

Modern machine learning applications deal with various types of data, which might originate from different entities and/or spread across different geographical locations. In particular, anomaly detection, relies on combining diverse data from different sources to generalize well. However, transferring data to a centralized location for further processing is not always possible due to data protection, usage, and ownership restrictions. This results in data silos that cannot be merged and represent a serious impediment for most anomaly detection tasks dealing with sensitive data.

Federated learning provides a privacy-preserving solution that allows training machine learning models across multiple distributed clients holding local data without exchanging them. While preserving privacy, such solution is usually associated with loss in the predictive performance compared to centralized training.

In this thesis, we provide a comprehensive evaluation of federated learning when applied to anomaly detection for different label availability scenarios. We investigate the effect of federated learning on the predictive performance for various applications. For this purpose, federated models are compared to models trained using only the locally available data, to models trained on centrally aggregated data, and to centralized models trained on aggregated synthetic data generated by each client individually. We also investigate the effect of amount and distribution of data locally available at each client on the predictive performance.

We show that, federated learning is able to provide good predictive performance compared to other settings for most cases of label availability. Unlike synthetic data-based learning, which seems to highly depend on the type of training data, it consistently provides good predictive performance across different data sets. In addition, federated learning is able to perform well under different data distribution scenarios.

Contents

CHAPTER 1

Introduction

1.1 Motivation

The amount of data available worldwide has significantly increased over the last decade due to the increase in the number of devices that generate such information. The global amount of data generated, copied and consumed is expected to reach 97 zettabytes in the year 2022 [RD22]. This resulted in a tremendous increase in the number and size of available data sets generated from multiple sources such as financial transactions, web and telecommunication logs, and health records [TBJS20]. Analyzing large amounts of data has therefore became a necessity and has been providing businesses and enterprises with valuable discoveries and insights [HKP12]. For instance, financial institutions may discover hidden patterns and correlations within transactions that are used to reduce fraud, hospitals keep track of their patient records to spot trends and anomalies, and computer networks can be monitored for the purpose of intrusion detection. The set of techniques and tools that help transforming data into useful information and knowledge is known as *data mining*. In particular, when analyzing real-world data sets, there has been an interest in detecting instances that stand out as being dissimilar to others. This is done by means of *anomaly detection*, defined by [HKP12] as the process of detecting data instances with behaviors that are very different from what is expected. It is extensively used in a wide variety of applications such as fraud detection, intrusion detection, in medical data and military surveillance [CBK09]. Anomaly detection is of high importance since anomalies in data usually translate to critical actionable information in various applications.

Many current machine learning methods require large amounts of data that might be collected from various sources with different access and usage requirements. This raises serious privacy concerns related to the risk of leakage of private or confidential data. It is therefore important to employ privacy-preserving machine learning solutions. One increasingly used solution is federated learning. It is a distributed learning setup where multiple instances of the same model type/architecture are trained on private, local data sets and then aggregated to a global model, which in turn is sent back to local nodes for further training, or usage. Privacy is therefore enhanced since only models are shared while data remain on the client side. Another approach is to perform privacy preservation on the data level, where synthetic data are generated using a model built from the original data. Synthetic data generation, unlike other techniques like data anonymization, generates a new artificial data set that protects sensitive information from being disclosed while retaining statistical properties that are similar to the real data [RBB+20]. It is therefore considered that trying to obtain the original data by means of reverse engineering is unlikely.

In particular, various anomaly detection applications involve sensitive data that might include personal or confidential information (e.g. patient records from hospitals and health institutions). This raises serious concerns about the usage of such data and introduces a need to use privacy-preserving machine learning techniques.

1.2 Problem definition

Anomaly detection is usually considered as a challenging task due to the relatively high data imbalance where only small number of anomalies are available compared to the non-anomalous data instances [MMH17]. In addition, for some applications, labeled data might be difficult and/or expensive to obtain and therefore only semi-supervised or unsupervised methods can be used. This makes it very challenging to train models that result in high predictive performance on unseen data.

While federated learning improves privacy, it might negatively affect the anomaly detection model performance. In [DKP+20], the authors performed a comparative study between centralized and federated learning scenarios for a classification task and concluded that the centralized approach outperforms the federated one for non independent and identically distributed (non-i.i.d.) data. A similar behavior is observed when comparing classification models trained on real and synthetic data, where the model trained on the full real data provides higher accuracy [RBB+20]. We assume that data synthesis will have greater effect when applied to data located at individual nodes rather than on larger amount of data at the centralized server. This suggests that applying privacy-preserving techniques imposes trade-offs between various factors such as improved privacy, increased communication costs, and lower predictive performance of the resulted model.

Given an architecture consisting of multiple nodes containing variable amounts of similar data, we define four learning settings:

- Centralized learning: data from the individual nodes are transferred to a central server where they are aggregated and used to train a centralized model. This serves as a glass-ceiling baseline, and scores from such experiments will be used as a reference for the other training scenarios.

- Local training: models are trained at the client nodes, using only the locally available data.

- Federated learning: separate models are locally trained where the data resides. Single model updates are then communicated to generate a global model, potentially repeated for multiple rounds.

- Learning from synthetic data: the local original data at the node is used to generate synthetic data. Synthetic data from each node are then aggregated at a central server and used to train a single model.

The above-discussed machine learning settings are shown in Figure 1.1.

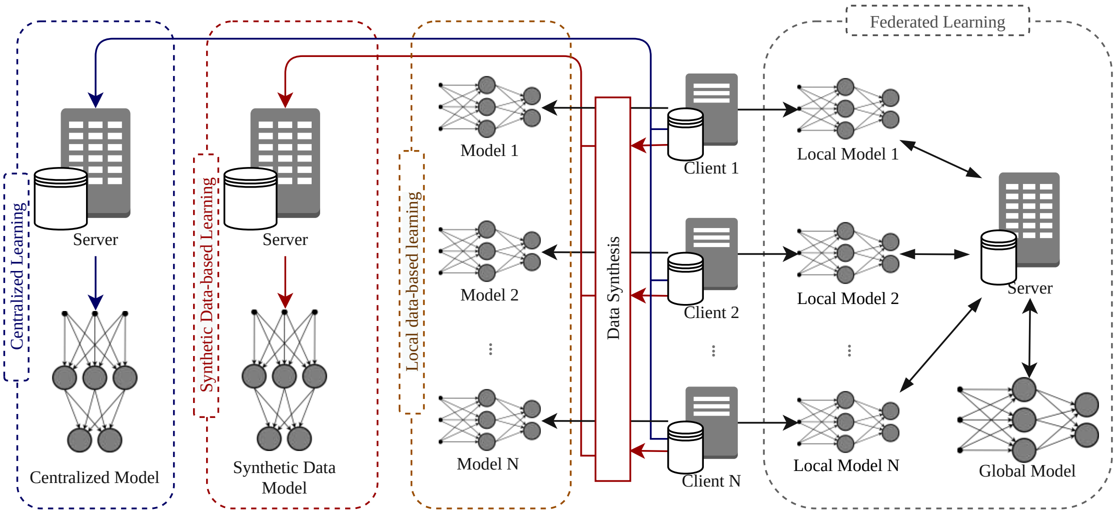


Figure 1.1: Different Learning Settings

Evaluating the predictive performance resulting from each learning setting will provide valuable information on the effect of privacy preserving on anomaly detection methods.

The idea of independent and identically distributed (i.i.d.) data is often adopted when training machine learning models. However, many federated learning scenarios are characterized with a certain statistical heterogeneity where data are generated in a non-i.i.d. manner across the network [LSTS20]. As part of the experiments, the effect of data heterogeneity is evaluated by considering both i.i.d. and non-i.i.d. scenarios.

1.3 Research questions

Based on the problems defined above, this thesis will provide an in-depth evaluation of existing anomaly detection methods and their applicability when trained in a collaborative way. The focus will be on tabular data and machine learning models covering the different training scenarios: supervised, weakly-supervised, semi-supervised and unsupervised, and will be evaluated using existing benchmark data sets.

In the next section, we define the research questions that will be addressed in this thesis. Three main research questions are defined:

RQ 1) What is the effect of collaborative learning when applied to anomaly detection models compared to centralized learning?

First, we address the effect of federated learning on the overall performance of the different models. In particular, we are interested in comparing the performance of the collaborative models to the centralized model to better understand to what extent the performance is affected. This question is further broken down into two sub-questions:

RQ 1.1) To what extent does training such models in a federated manner affect the overall predictive performance?

The predictive performance of anomaly detection models is normally evaluated in terms of metrics such as precision and recall. In order to analyze the effect of collaborative learning on the overall model performance, such metrics are measured for both centralized and federated scenarios. The centralized scenario is considered as a baseline that represent the results that can be obtained without the need for distributed learning and is used to evaluate the robustness the global federated model.

RQ 1.2) Which models are more suited to be used in a federated setting for more effective anomaly detection?

Based on the results obtained from the performance evaluation, we are interested in identifying which models are well-suited for federated anomaly detection.

RQ 2) What impact does data heterogeneity have on the anomaly detection models in federated learning?

One of the most important characteristics of federated learning is that we usually deal with statistically heterogeneous data. This results in violating the common assumption of independent and identically distributed (i.i.d.) data. Thus, we are interested in evaluating the impact of data distribution on the overall federated model performance. For this purpose, we define the following sub-questions:

RQ 2.1) To what degree do the amount and distribution of data available at local nodes influence the global model performance?

As part of the evaluation, both i.i.d. and non-i.i.d. scenarios are simulated. The non-i.i.d. scenario is of particular interest where various methods are used to achieve data heterogeneity over the different clients. We are interested in the effect of data distribution on the predictive performance of the federated model for the different training scenarios.

RQ 2.2) To what extent does applying resampling techniques to local data at individual nodes affect the global model predictive performance?

Various resampling techniques including oversampling and undersampling are applied to the data at the local nodes in the supervised scenario. Here, we are interested in evaluating the effect of resampling on the predictive performance and how models trained on data with resampling perform compared to the ones without resampling.

RQ 3) How do models trained in federated manner perform when compared to central models trained using synthetic data locally generated at client nodes?

Another way of preserving privacy is generating synthetic data at local nodes and then transferring them to the central server. In this thesis, we are interested in comparing the performance of federated learning models to models trained on aggregated synthetic data generated at local nodes.

1.4 Structure of the thesis

The remainder of this thesis is structured as follows. Literature about different aspects of anomaly detection, state-of-the-art methods and their corresponding applications is provided in Chapter 2. Chapter 3 provides an introduction to privacy-preserving machine learning and the related challenges. In particular, settings that are relevant to this thesis, data synthesis and federated learning, are discussed. Related working combining federated learning and anomaly detection for tabular data is discussed as well.

Chapter 4 provides a detailed description of the experiments, including the involved design decisions. We start by describing the selected data sets and algorithms. We then define a way to simulate different data distribution scenarios and the performance metrics used in the evaluation phase. Finally, for each learning setting defined in Section 1.2, the learning process is described in details.

Chapter 5 presents the results for each of the learning settings. It also provides a detailed discussion of the results.

Finally, Chapter 6 summarizes the contribution of this thesis and the main conclusions. The research questions are also revisited and findings related to each of them are presented. Finally, future work and potential improvements are discussed.

Tables including detailed results for all experiments performed as part of this thesis, can be found in Appendix A.

CHAPTER 2

Anomaly Detection

Anomaly detection is an important task in data mining and machine learning, and has seen applications in a wide range of domains. An anomaly, also known as outlier, is an instance of a data set that exhibits some abnormality or some kind of out of the way behavior [RSMMA19]. In other words, an anomalous data instance is a data object that significantly deviates from what is considered to be normal data. There may be various reasons for such anomalies to be present in the data. Malicious activities such as credit card fraud or cyber-intrusions, or also system failures represent a major type of anomalies that can significantly affect the functionality of critical systems. Thus, being able to effectively detect anomalies is of high importance and in most of the cases leads to critical actionable information.

In this section, we discuss various aspects of anomaly detection and their related challenges. In addition, we provide an overview about the state-of-the-art methods, their applicability in real-life scenarios and their limitations. Finally, we discuss applications of anomaly detection with a focus on fraud detection, intrusion detection, and anomaly detection in health care.

2.1 Aspects of anomaly detection

In multiple scientific and engineering fields, the generated data represent the state of a system whose processes follow various rules and principles [MMH17]. Such data can then be used to formulate hypotheses about the underlying processes that describe the *normal* states of the system. However, most of such systems may also exhibit states that deviate from the normal behavior and, therefore, the resulting data are different from the ones observed previously. The task of discovering such variations in the observed data is known as anomaly detection.

An example of a 2-dimensional data set with anomalies is illustrated in Figure 2.1. The data consist of two normal regions C_1 and C_2, where data points representing the normal behavior are located. All other points that are far from these two regions are considered as anomalies. This includes regions O_1 and O_4 and individual data points o_2 and o_3.

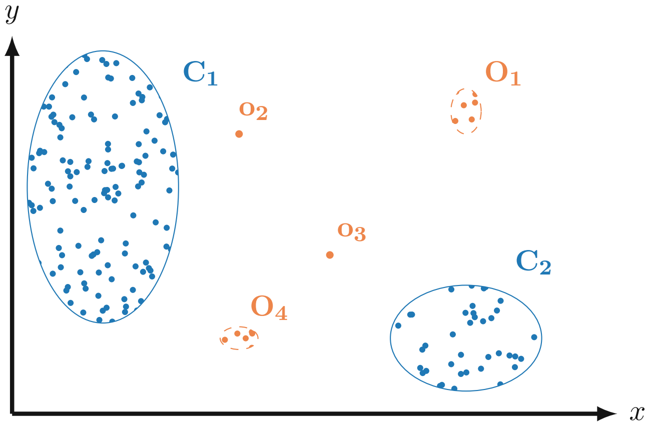


Figure 2.1: An example of a two-dimensional data set with anomalies [based on figure from [CBK09]]

2.1.1 Types of anomalies

A key aspect of solving any anomaly detection problem is identifying the nature of the target anomalies. Some of the previous work [AA19, MMH17] propose to group anomalies into three categories: point anomalies, contextual anomalies, and pattern anomalies.

Point anomalies are individual data instances that are considered as anomalous with respect to normal data points. While point anomalies could be caused by certain random errors or other systematic errors (e.g. faulty sensor), they may reflect abnormalities that represent a deviation from what is considered normal.

In Figure 2.1, point anomalies are represented by sets O_1 and O_4 that contain multiple anomalous points and by individual data instances o_2 and o_3 as well. Point anomalies are the focus of majority of research on anomaly detection [CBK09] and will be the focus of this thesis as well.

Contextual anomalies are defined as data instances that are anomalous in a specific context, but not otherwise [CBK09]. The context should be defined in the data set and has to be part of the problem formulation. For instance, in time series, time defines the position of a data point in a given data set and therefore can be used as a contextual attribute.

Pattern-based anomalies, also known as collective anomalies, are defined as a collection of related data instances that deviate from their historical counterparts [MMH17]. Single data instances within a collective anomaly may not be anomalies by themselves, but their combined occurrence as a collection is anomalous.

2.1.2 Data labels

In a given data set, labels indicate whether a data instance is normal or anomalous. In most cases, a human expert is required to manually annotate the data. For some applications, such labels are expensive and difficult to obtain. In addition, the anomalous behavior is usually dynamic: situations where new types of anomalies, for which there is no available training data, might occur.

Based on the availability of labels, and as illustrated in Figure 2.2, four training scenarios can be defined:

1. Supervised Anomaly Detection: labeled training and test data are available for data instances representing both normal and anomaly class. In this scenario, a (binary) classifier can be trained to distinguish between normal and anomalous data instances. In most cases, the available data set is highly imbalanced and therefore not all classification algorithms are suitable for this task. For instance, decision trees are known to be unable to deal with imbalanced data [CBK09], while other methods such as Support vector machine (SVM) and neural-netword based methods provide better predictive performance [CBK09].

2. Weakly-supervised Anomaly Detection: large amounts of normal data instances are provided while only a very limited number of anomalous data are available. While anomaly detection is already considered as a severe case of a class imbalance problem [BKW20], in this scenario, class imbalance is even more extreme.

3. Semi-supervised Anomaly Detection: training data consists only of normal data instances (or sometimes also only anomalous data instances), while test data include labeled instances for both classes. Many model types can be trained on data representing normal behavior and then used to identify anomalies in test data. Some other techniques rely on anomalous instances for training, and can thus not be used.

4. Unsupervised Anomaly Detection: labels are not available for any of the classes in training data. Several semi-supervised techniques can be adapted to operate in unsupervised mode and trained on a sample of the unlabeled data set [CBK09]. In this case, the assumption that the test set contains only few anomalies should hold.

2.1.3 Evaluation metrics

The main goal of anomaly detection is to determine whether a given data instance is anomalous or not. In other words, for a data point x we assign the class *normal* or

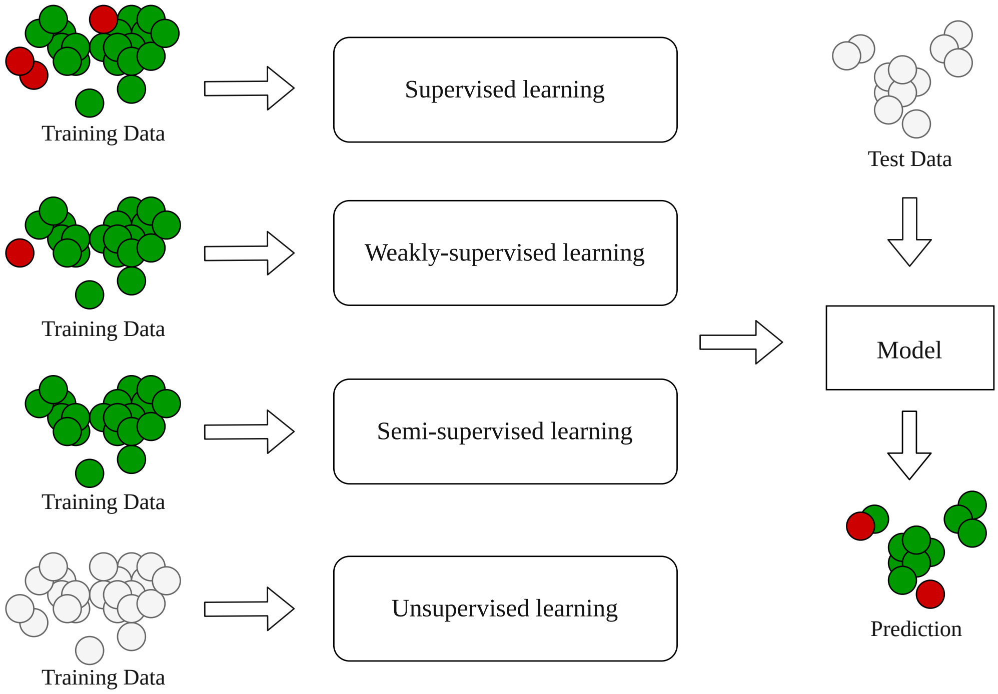


Figure 2.2: Comparison of different learning scenarios

anomalous (usually encoded as 0 and 1). In analogy to the output of binary classification, a *positive* output corresponds to *anomalous* data instance while *negative* output suggests a *normal* data instance.

The case where the data instance is anomalous and the class *anomalous* is assigned to it is known as a True Positive (TP). On the other hand, if the data instance is normal but is predicted as *anomalous*, this is a False Positive (FP). Equivalently for a normal data instance, we have a True Negative (TN) if the model assigns the class *normal* to it and False Negative (FN) if the model predicts it as *anomalous*. These values can be shown in what is known as a “confusion matrix”, shown in Table 2.1.

In statistical hypothesis testing, false positive and false negative are related to the concepts of type I error and type II error. Type I error refers to the mistaken rejection of an actually true null hypothesis. In this case, we reject what turns out to be true for the favor of something that is false and we therefore have a “false positive finding”. Type II error refers to the acceptance of null hypothesis that is actually false. In this case, we accept what is false and reject what is true and therefore have a “false negative finding”.

In order to understand how a specific model performs, we usually rely on various metrics:

- Accuracy represents the ratio of correctly predicted observations. It describes how the model performs across the different classes.

- Precision or Positive Predictive Value (PPV) describes how many of the positive predictions turned out to be actually positive. In the context of anomaly detection, it provides the proportion of data points correctly identified to be anomalous among all data points that have been classified as anomaly.

- Negative Predictive Value (NPV) depicts how many of the negative predictions turned out to be actually negative. For anomaly detection, it gives the proportion of data points that have been correctly identified to be normal among all data points that have been classified as normal.

- Recall or True Positive Rate (TPR) or Sensitivity describes how many of the data points that are actually positive did the model correctly predict. For anomaly detection, it is the proportion of anomalous data points that have been correctly classified as anomalous.

- Fall-Out or False Positive Rate (FPR) depicts how many of the actually negative data points are incorrectly predicted as positive. In the context of anomaly detection, it can be seen as the false alarm ratio where it represents the portion of the data points that have been falsely classified as anomalous among all data points that are actually anomalous.

- Miss Rate or False Negative Rate (FNR) describes how many of the data points that are actually positive did the model incorrectly predict as negative. For anomaly detection, this represents the proportion of anomalous data points that have been falsely classified as normal.

- Specificity describes how many of the data points that are actually false did the model correctly predict. In the context of anomaly detection, it is the proportion of normal data points that have been correctly classified as normal.

The different formulas for calculating such metrics can be seen in Table 2.1.

Accuracy is often considered to be the most intuitive metric and is widely used by researchers to select models [DDRF^{+}22]. However, accuracy is considered to be an overly optimistic estimation of the ability of the classifier over the majority class [CJ20] and is therefore sensitive to imbalanced data. Since anomaly detection tasks deal with highly imbalanced data, this metric does not seem to be of interest.

The other metrics mentioned above do not utilise all confusion matrix elements. For instance, recall only focuses on positive data instances, while specificity only considers the negative ones. For highly imbalances data sets, the goal is to improve recall without negatively impacting precision. However, these objectives are usually conflicting since when trying to increase the true positive rate for the minority class, the number of false positives also often increases – resulting in a lower precision [HC13]. The F-measure, the weighted harmonic mean of precision and recall, captures both properties and is usually considered a better choice for such tasks. A general formulation of F-score, known as F_β

Table 2.1: Confusion matrix representation

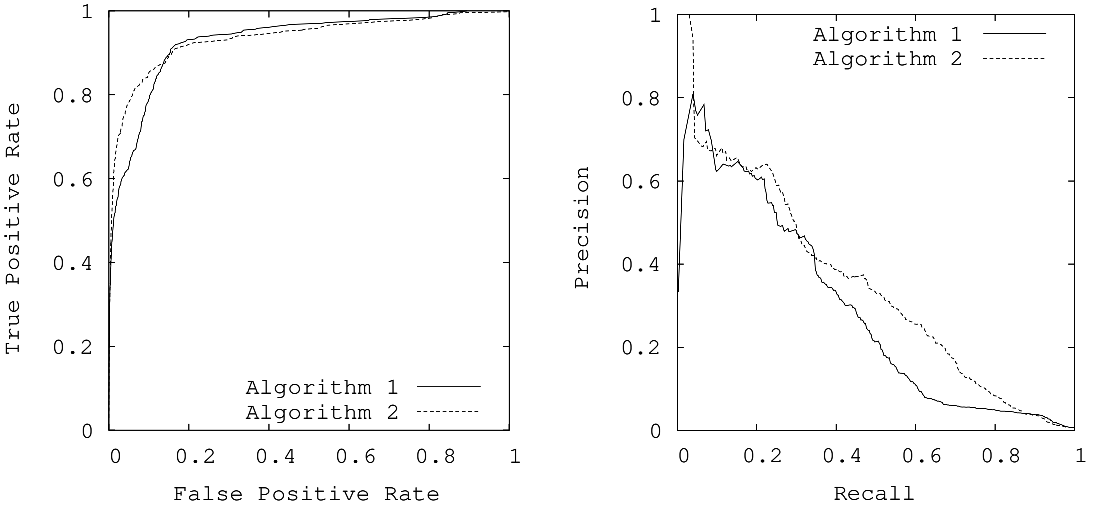
		Actual		
		Positive (1)	Negative (0)	
Prediction	Positive (1)	**True Positive** **TP**	**False Positive** **FP** **(Type I error)**	**Precision** or Positive Predictive Value (PPV) $\frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}}$
	Negative (0)	**False Negative** **FN** **(Type II error)**	**True Negative** **TN**	Negative Predictive Value (NPV) $\frac{\mathbf{TN}}{\mathbf{TN} + \mathbf{FN}}$
		Recall or Sensitivity or True Positive Rate (**TPR**) $\frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}}$	False Positive Rate (**FPR**) or Fall-Out $\frac{\mathbf{FP}}{\mathbf{TN} + \mathbf{FP}}$	**Accuracy** $\frac{\mathbf{TP} + \mathbf{TN}}{\mathbf{TP} + \mathbf{FP} + \mathbf{FN} + \mathbf{TN}}$
		False Negative Rate (**FNR**) or Miss Rate $\frac{\mathbf{FN}}{\mathbf{FN} + \mathbf{TP}}$	True Negative Rate (**TNR**) or Specificity $\frac{\mathbf{TN}}{\mathbf{TN} + \mathbf{FP}}$	

where β is chosen such that the recall is considered β times as important as precision, is defined in the following way:

$$F_\beta = (1 + \beta^2) \times \frac{precision \times recall}{(\beta^2 \times precision) + recall}$$

In particular, a widely used version of F_β is the F_1 score, which is defined as follow:

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} = \frac{2 \times TP}{2 \times TP + FP + FN}$$



(a) ROC curve

(b) Precision-Recall (PR) curve

Figure 2.3: Example of ROC and PR curves for two different algorithms [DG06]

Instead of a simple positive or negative prediction, multiple classification algorithms and most of the anomaly detection methods provide a numeric score for an instance to be classified in the positive class [FGG+18]. In the case of anomaly detection, an algorithm then decides on an instance to be normal or anomalous by applying a threshold to this score (i.e. a discretisation of the score). The choice of this threshold controls the trade-off between positive and negative errors [FGG+18]. Choosing a high threshold results in lower FPR and high FNR since the classifier becomes very restrictive in classifying an instance as positive. Inversely, selecting a lower threshold decreases the FNR and increases FPR as the classifier becomes more lenient in classifying instances as positive [FGG+18].

When evaluating algorithms that output probabilities of class values, the previously defined measures are not optimal since they require converting the probabilities to class labels by selecting a certain threshold. [PFK98] recommends using the Receiver Operator Characteristic (ROC) curves for evaluating binary decision problems. The ROC curve is created by plotting the TPR against the FPR as shown in Figure 2.3a. It shows how the number of correctly classified positive examples varies with the number of incorrectly classified negative examples with different thresholds.

However, in the case of large skew in the class distribution, ROC curves usually represent an overly optimistic view of the model's performance [DG06]. When addressing tasks with large data imbalance, Precision-Recall (PR) curve has been considered a good alternative to ROC curve [RBJ89, MS99, SR15]. Such curve shows the trade-off between precision and recall for different thresholds, as shown in Figure 2.3b.

The Precision-Recall curve might expose differences that are not easily noticeable in ROC curves. The example curves shown in Figure 2.3 shows the results from a model trained on highly-skewed cancer detection data [DBD+05]. Looking at the ROC curves (Figure 2.3a), where the goal is to be on the upper-left-hand corner, it can be concluded that both algorithms 1 and 2 are almost optimal. However, looking at the PR curves (Figure 2.3b), where the goal is to be in the upper-right-hand corner, it can be seen that there is still a large room for improvement. This difference in views is mainly related to the highly imbalanced nature of the data set, where the number of negative examples largely exceeds the number of positive examples by several orders of magnitude. Thus, a large change in the number of false positives will result in small change in the false positive rate used in the ROC curve [DG06]. ROC curves depicts the behavior of an algorithm independently of the class distribution or error cost and therefore they decouple the classification performance from such factors [PF97]. On the other hand, the PR curve better captures the changes in the number of false positives by considering the false positives instead of false negatives.

While visually analyzing the ROC and PR graphs might be helpful, it usually does not represent a convenient way of choosing an algorithm over another. In fact, it can only be concluded that an algorithm is better than another when it clearly dominates the other algorithm over the entire performance space. Hence, the need for an index that summarizes both ROC and PR curves arises. Such metrics are the Area under the ROC curve (ROC AUC) and Area under the PR curve (PR AUC). As the name suggests, these metrics calculate the area under the ROC or PR curve, to give a single score for a classification model across all threshold values.

The ROC AUC can be defined as the probability that the scores provided by a classifier will rank a randomly selected positive instance higher than a randomly selected negative one [FGG+18]. Given that the ROC AUC of random guessing is 0.5 and the ideal ROC AUC score is 1, a classifier that provides an ROC AUC score higher than 0.5 is considered to be useful [FGG+18].

Unlike ROC AUC, the PR AUC value does not have a probabilistic interpretation [FGG+18]. The PR AUC of the random classifier depends on the number of instances belonging to the positive class and its expected value is close to the proportion of positive instances [FGG+18].

2.2 State-of-the-art algorithms

Various anomaly detection algorithms are available for the different learning scenarios discussed in Section 2.1.2. The selection of such methods depends on various factors such as the application, data types, anomaly type and data labels availability [AMM+21]. In this section, the methods that are used in the experiments, as well as some other widely known algorithms are described. Based on existing taxonomies in the literature [GR12, GU16, ADE20, AMM+21], the different methods are categorized into five groups as shown in Figure 2.4.

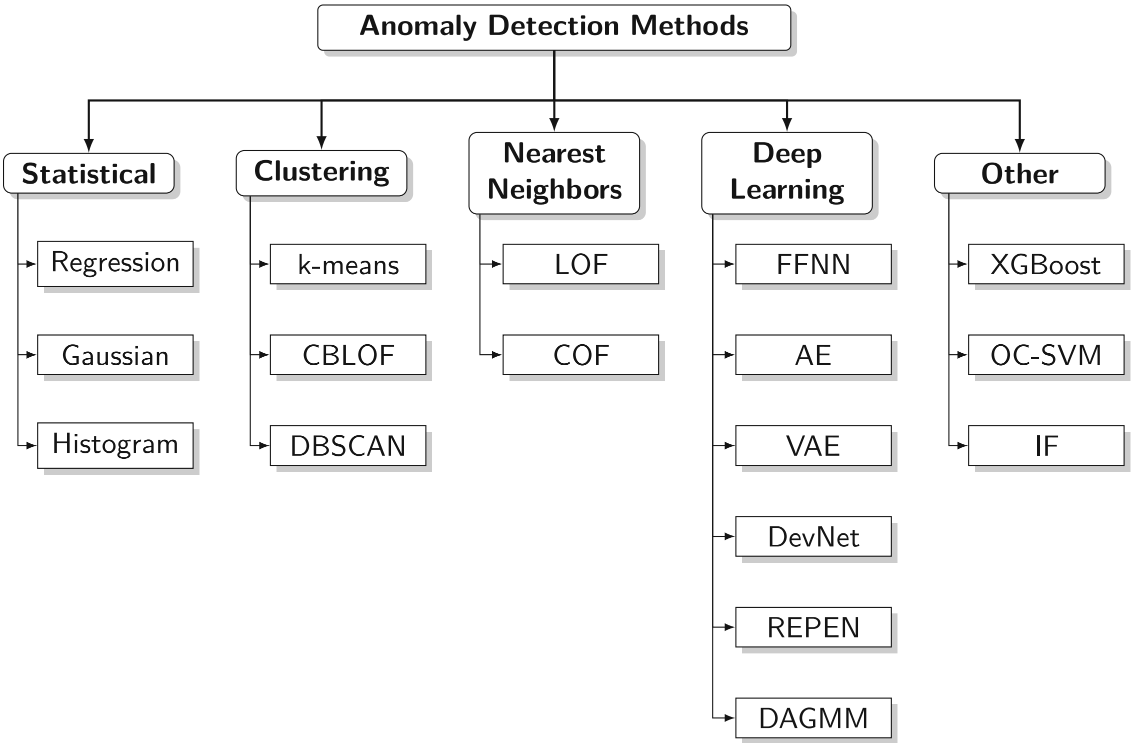


Figure 2.4: Taxonomy of the selected anomaly detection methods [Based on [GU16, ADE20, GR12, AMM$^+$21]]

2.2.1 Statistical methods

Statistical anomaly detection methods are based on the assumption that normal data points are located in high probability regions of a stochastic model while anomalous instances occur in low probability regions [CBK09]. These methods can be further subdivided into parametric techniques and non-parametric techniques.

Parametric techniques assume that data are generated by a parametric distribution with parameters Θ, and probability density function $f(x, \Theta)$ for an observation x.

Gaussian model-based methods are widely-used parametric techniques, where the data are assumed to be generated from a Gaussian (also called normal) distribution and the parameters are estimated using the Maximum Likelihood Estimates (MLE) [CBK09]. The anomaly score for a data instance is defined as its distance to the estimated mean.

Given the normal distribution curve, it is possible to approximate the proportion of data that falls within certain intervals. This is given by the Empirical Rule, also known as the 68-95-99.7 rule [Hub19]. This rule is illustrated in Figure 2.5 and states that:

- Approximately 68% of the data points will fall within no more than one standard deviation from the mean.

- Approximately 95% of the data points will fall within two standard deviations from the mean.
- Approximately 99.7% of the data points will fall within three standard deviations from the mean.

One of the simple outlier detection methods, known as the “standard deviation method”, is to consider all data points that are more than 3σ distance away from the mean μ as outliers. However, this is method may not be able to detect outliers, because outliers highly affect the standard deviation. In fact, extreme outliers magnify the standard deviation and therefore result in a broader detection range $[-3\sigma, 3\sigma]$. This results in failing to detect “less extreme” data instances.

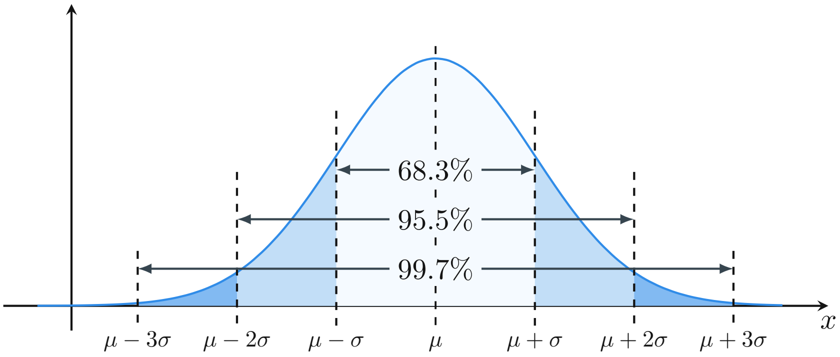


Figure 2.5: Gaussian distribution curve with the three-sigma rule

Another method that is commonly used to identify outliers uses a graphical representation known as boxplot [Tuk77]. The data is represented in a boxplot with the smallest non-outlier (or non-anomaly) observation, lower quartile ($Q1$), median, upper quartile ($Q3$) and largest non-outlier (or non-anomaly) data point as can be seen in Figure 2.6. The range defined by $Q3 - Q1$ is known as the Inter Quartile Range (IQR). Any data points that lie below the minimum or above the maximum are considered as anomalies. Since 99.3% of data points are located between the minimum and the maximum values, this method is equivalent to the previously described Empirical Rule method.

Regression models represent another type of parametric techniques for anomaly detection. They can be used for anomaly detection where a model is first fitted on the data and the anomaly score is determined based on the residuals. A residual is the part of an instance that cannot be explained by the model and its magnitude can be used as the anomaly score.

Logistic regression, also known as logit regression, is commonly used in anomaly detection applications to estimate the probability that an instance belong to the normal or anomalous class. Similar to linear regression, logistic regression computes a weighted sum of the input features with a bias term, but instead of providing the result directly,

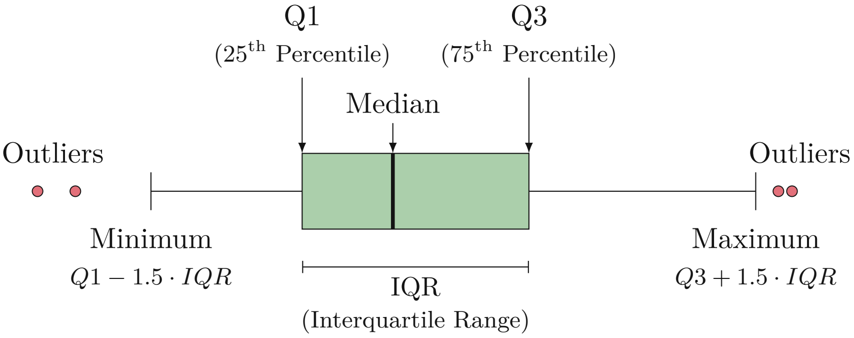


Figure 2.6: Box plot representation

it applies logit function on the output to obtain a probability value. The estimated probability for a given instance $\mathbf{x}$ is given by: $\hat{p} = h_\theta(\mathbf{x}) = \sigma(\theta^T \cdot \mathbf{x})$, where θ are the model parameters, h_θ is the hypothesis function and $\sigma = 1/\left(a + exp(-t)\right)$ is the logistic function. Given the estimated probability, it is possible to make prediction on the class membership using the following equation:

$$\hat{y} = \begin{cases} 0 \text{ if } \hat{p} < 0.5 \\ 1 \text{ if } \hat{p} \geq 0.5 \end{cases}$$

When training a logistic regression model, the goal is to define the parameter vector θ so that the model outputs high probabilities for normal instances and low probabilities for anomalies. For this purpose, a cost function known as the log loss is used. Given m data points the log loss can be defined as follow [Gér19]:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \cdot log\left(\hat{p}^{(i)}\right) + \left(1 - y^{(i)}\right) \cdot log\left(1 - \hat{p}^{(i)}\right) \right]$$

The log loss is a convex function, and therefore optimization algorithms (such as gradient descent) can be used to find its global minimum. The partial derivative for the j^{th} model parameter θ_j is given by the following equation:

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} \left(\sigma\left(\theta^T \cdot \mathbf{x}^{(i)}\right) - y^{(i)} \right) \cdot x_j^{(i)}$$

The gradient vector containing all partial derivatives is passed to the stochastic gradient descent algorithm. Batch size is a hyperparamter of the gradient descent algorithm that controls the number of training samples to be used before the model's parameters are updated. It is possible to use stochastic gradient descent, where only a single data instance is processed at a time, mini-batch gradient descent where more than one training instance known as mini-batch is used at every step and batch gradient descent where at each step the full training set is processed.

Another type of statistical anomaly detection methods are non-parametric models, which do not use a predefined model structure, but instead determine it from the the given data [CBK09]. Such models make fewer assumptions on the data compared to parametric techniques [CBK09]. One of the methods that is popular in intrusion and fraud detection communities is known as histogram-based anomaly detection. In the case of univariate data, a histogram is first built using the provided data, and the value of each test data instance is checked if it falls in any of the bins of the histogram. If it does not fall in any of the bins, it is considered to be anomalous. Furthermore, an anomaly score can be defined based on the height of the bin into which the data instance falls. Such method requires defining the size of the bins to construct the histogram. This is challenging, since defining small bins will result in test instances falling in empty or rare bins while large bins may result in anomalous instances falling in frequent bins and therefore results in high false negative rate [CBK09].

In the case of multi-variate data, histograms are constructed for each attribute, and anomaly scores for a test instance are then obtained for each of them and aggregated to define the overall anomaly detection score.

2.2.2 Clustering-based methods

Cluster analysis uses the information found in the data to create groups of data points, where within a single group instances are similar (or related) to one another, while being different (or unrelated) to the instances in other groups [TSKK19]. Clustering-based anomaly detection methods rely on the assumption that normal instances appear close to each other and therefore can be grouped into clusters. On the other hand, anomalies do not fit well in any of the normal clusters, or appear in small clusters that are away from normal clusters [TSKK19].

K-means [Llo82] is one of the most widely used algorithms for clustering-based anomaly detection. As described in Algorithm 2.1, the algorithm repeatedly assigns each data point to the nearest cluster and recomputes the centroid of each cluster until a convergence condition is met.

The distance of a data point to its cluster centroid represents how strongly it belongs to it. Therefore, instances that are distant from their respective centroids are assumed to be anomalies. [TSKK19] defines two different ways of calculating the anomaly score for a given data point:

(a) The distance between the data point and its closest centroid

(b) The relative distance between the data point and its closest centroid: the ratio of the point's distance from the centroid to the median distance of all points in the cluster from the centroid.

A limitation of the clustering-based approaches is the fact that the quality of clusters corresponding to normal classes is highly affected by the presence of outliers. In addition, the number of selected clusters highly affect the model performance.

Algorithm 2.1: K-means clustering

```
Require: Data set $\mathcal{D}$, Number of cluster ($k$), Termination threshold ($\theta$);
Randomly choose $k$ elements of $\mathcal{D}$ as the initial set of Centroids: $C = \{c_1, ..., c_k\}$;
Repeat
    Assign each data point $p \in \mathcal{D}$ to its closest Centroid minimizing the distance
      $d_E(p, c_j)$;
    Update the value of each Centroid $c_j$ to the new mean value of all data points
      assigned to it;
Until The number of points assigned in the current iteration is lower than $\theta$;
```

Another clustering-based anomaly detection method named Cluster-based Local Outlier Factor (CBLOF) (also known as *FindCBLOF*) is proposed by [HXD03]. For each data point, this method assigns an anomaly score that captures the size of the cluster to which the instance belongs and the distance between the data point and its cluster centroid. First, a clustering algorithm is used to assign each data point to a single cluster. [HXD03] uses the Squeeze algorithm [HXD02], but any other algorithm (e.g. k-means) may be used. The clusters are then ranked according to their sizes from large to small. The clusters holding 90% of the data are defined as large clusters while the ones holding the remaining 10% are called small clusters. The outlier score of a data point is then calculated in the following way:

- If the data instance belongs to a large cluster, the anomaly score is the distance to the centroid multiplied by the number of data points in that cluster.

- If the data point belongs to a small cluster, the anomaly score is the distance to the centroid of the closest large cluster multiplied by the number of data of the cluster to which the data point belongs.

Another popular clustering method that is being used for anomaly detection is Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [EKSX96]. This method groups data points that are closely packed together and considers data points that lie in low-density regions as outliers. In contrast to k-mean clustering, this method does not require the number of clusters to be defined. [TSKK19]

2.2.3 Nearest neighbor-based methods

Nearest neighbor-based (also known as proximity-based) anomaly detection methods are based on the assumption that normal data points occur in dense neighborhoods while anomalous instances occur far from their closest neighbors [CBK09]. Proximity-based approaches can be grouped into two categories [CBK09]:

- Methods that define the anomaly score as the distance of a data point to its k$^{\text{th}}$ nearest neighbor.
- Methods that use the relative neighborhood density of each data instance to calculate its corresponding anomaly score (e.g. Local Outlier Factor (LOF), Connectivity-based Outlier Factor (COF)).

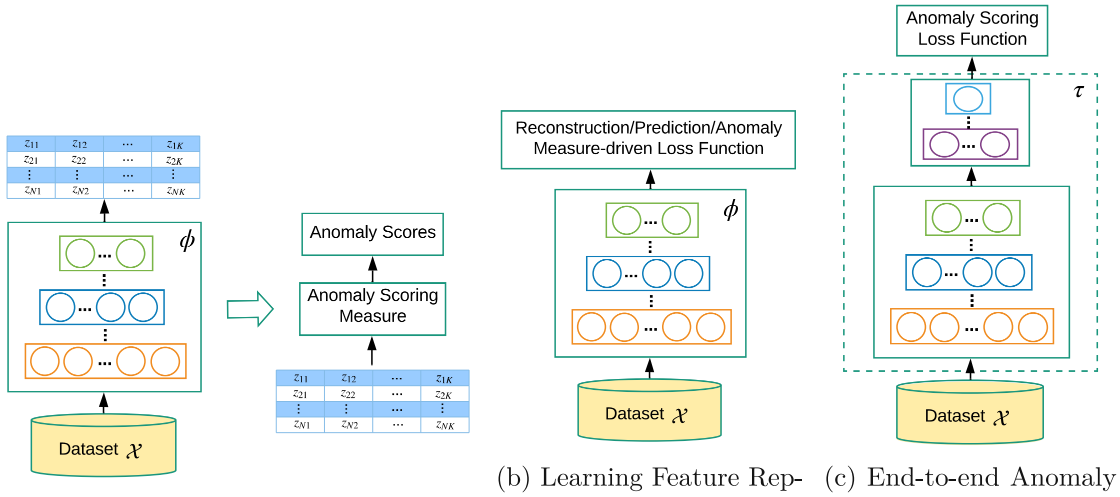
One of the simplest ways to define the nearest neighbor-based anomaly score of a data point x is to consider the distance to its k^{th} nearest neighbor $dist(x, k)$. A normal data instance should have multiple other instances close to it and therefore a low distance value, while an anomalous data point has a high distance value since it is quite distant from its k$^{\text{th}}$ neighbor. For this approach, the anomaly score is very sensitive to the value of k where for too small values, anomalous instances located close to each others might have low anomaly score. Alternatively, a more robust approach is to take the average distance to the first k-nearest neighbors as anomaly score. This formulation of the anomaly score has been widely used in multiple applications [TSKK19].

Another proximity-based method that relies on the relative density of the data point to calculate the anomaly score is the Local Outlier Factor (LOF) [BKNS00]. For each data point, it measures the LOF score defined as the local deviation of a data point with respect to its k nearest neighbor. It uses the local reachability density of a given data point and compares it to the local reachability density of all its k nearest neighbors. A data instance with large LOF value is declared as anomalous while an instance with low LOF is assumed to be normal.

Even though LOF performs well in multiple applications, its effectiveness is highly affected if the density of an outlier is close to densities of its neighbors [MMH17]. Therefore, a variation of LOF known as Connectivity-based Outlier Factor (COF) [TCFC02] is introduced. It operates in a similar way to LOF but computes the neighborhood in an incremental manner. For a given instance, the nearest data point is first added to the neighborhood set. The next instance to be added to the set is defined as the one having the minimum distance to the existing neighborhood set among all remaining instances [CBK09]. The distance between an instance and a set of instances is given by single linkage where it is equal to the minimum distance between such instance and any instance within the set. The neighborhood grows in a gradual way until its size reaches the value k.

2.2.4 Deep-learning based methods

Over the last few years, deep learning has shown great capabilities to learn expressive representations of complex data and has been widely applied in multiple applications [GBC16]. When applied to anomaly detection, it is usually referred to as *deep*


z_{11} z_{12} ⋯ z_{1K}
z_{21} z_{22} ⋯ z_{2K}
z_{N1} z_{N2} ⋯ z_{NK}
ϕ
Dataset $\mathcal{X}$
Anomaly Scores
Anomaly Scoring Measure
Reconstruction/Prediction/Anomaly Measure-driven Loss Function
Anomaly Scoring Loss Function
τ

(a) Deep Learning for Feature Extraction

(b) Learning Feature Representations of Normality

(c) End-to-end Anomaly Score Learning

Figure 2.7: Conceptions of the main deep anomaly detection approaches as defined by [PSCvdH22]

anomaly detection, where the goal is to learn feature representations or anomaly scores using neural networks to spot anomalies [PSCvdH22]. Multiple deep anomaly detection methods have been introduced and show a performance increase compared to traditional anomaly detection methods (e.g. LOF or CBLOF) for various applications [PSCvdH22, CC19, TBJS20, BKL$^+$21]. These methods provide a way to learn useful representations specifically tailored to the anomaly detection task and allow end-to-end optimization of a custom anomaly score, thus providing significant improvement over traditional methods.

Based on a literature review, [PSCvdH22] groups state-of-the-art deep anomaly detection methods into three major groups as shown in Figure 2.7.

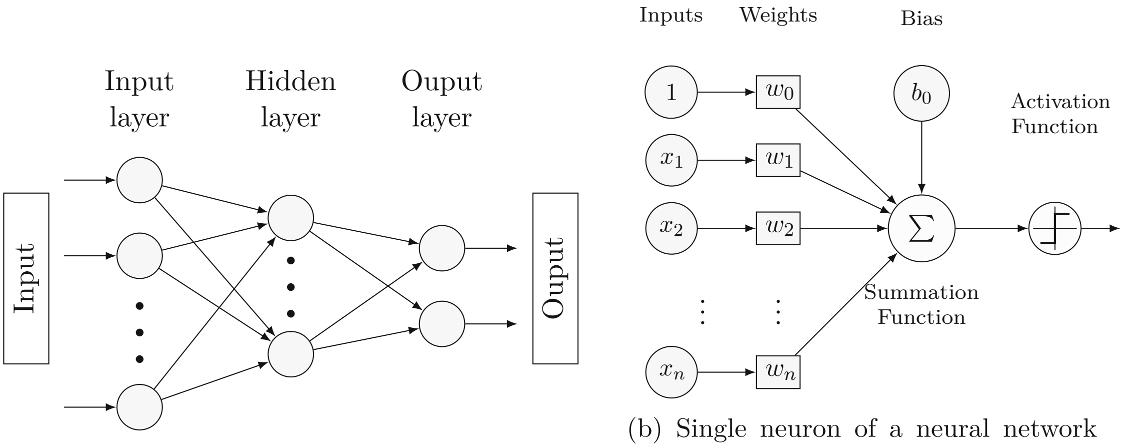
In the first group, deep learning methods are used for feature extraction and the anomaly detection task is performed separately based on the resulting features, using traditional methods like OC-SVM [XRY$^+$15]. This is illustrated in Figure 2.7a. The second category consists of deep learning methods aiming to learn expressive representations of normal instances (Figure 2.7b). It includes methods that rely on existing shallow anomaly measures such as distance- or clustering-based measures to learn representations. The third category includes methods that learn anomaly scores via neural networks in a end-to-end manner (Figure 2.7c).

Feedforward neural networks

Feedforward Neural Network represents one of the most widely used deep learning methods [GBC16], with the Multilayer Perceptron (MLP) being the most prominent mode. Given a certain input x and a target y, a feedforward network defines a mapping

$y = f(x; \theta)$ and learns the values of the parameters θ that provide the best function approximation [GBC16]. In such a model, information flow through the function being evaluated from x through some intermediate computations defining f to get the output y.

A feedforward neural network consists of a certain number of neuron-like processing units, also referred to as nodes, organized in layers. Units in a layer are connected with units from other layers. Intermediate layers that are located between the input and the output are known as hidden layers. An example of a feedforward neural network with a single hidden layer can be seen in Figure 2.8a.



(a) Example of a Feedforward Neural Network

(b) Single neuron of a neural network (Perceptron)

Figure 2.8: Graphical representation of Feedforward Neural Network

Figure 2.8b shows the operations performed at the basic unit of computation in a neural network, often referred to as *neuron* or *node*. The input from the previous layer is fed to the current neuron, where weights reflect the strength of the connection between nodes. The products of the inputs and their corresponding weights are then summed up across all connections from the previous layer with an additional bias term. The obtained sum is then passed to an activation function, which introduces non-linearity to the output of the neuron.

The training process, aiming to find the best set of parameters for the model, contains two main phases: feedforward and backpropagation. In the feedforward phase, the output is calculated given the input values and compared to the ground truth, estimating the error. In the backpropagation phase, the weight and bias values are adjusted until the output error is below a predetermined threshold, or another stopping criterion is met.

Neural networks have been widely used in various anomaly detection applications [CBK09]. A basic neural network for anomaly detection is trained in a supervised manner where during the learning phase, the model is trained to distinguish between normal and anomalous classes by predicting an anomaly score.

Autoencoders

Another popular neural network-based anomaly detection method is known as *Autoencoder* [Sch15]. An Autoencoder is a type of feedforward neural network where both the number of input and output neurons equal the number of original attributes [TSKK19]. The general architecture of an Autoencoder includes two main steps: *encoding* and *decoding*, as shown in Figure 2.9. In the encoding part, a data instance $\mathbf{x}$ is transformed to a lower-dimensional representation $\mathbf{z}$. This is performed via a number of nonlinear transformations performed by the encoder, a set of one or more encoding layers. The encoding layers are defined in a way where the number of neurons at a given layer is always lower than the number of neurons at the previous one. The learned representation $\mathbf{x}$ is then mapped back to its original space of attributes using the decoder, a set of decoding layers with increasing number of neurons. The output of the decoder is a reconstruction of $\mathbf{x}$ denoted by $\hat{\mathbf{x}}$.

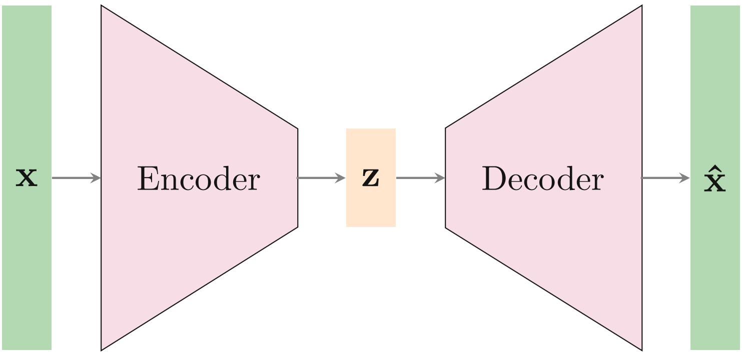


Figure 2.9: General architecture of an Autoencoder

Training an Autoencoder usually involves providing it with an input data that include only normal instances with the goal of learning complex and non-linear representations of the normal class.

The difference between $\hat{\mathbf{x}}$ and $\mathbf{x}$, known as the reconstruction error, can then be used as anomaly score [TSKK19], as the assumption is that the reconstruction error is smaller on normal inputs, whose characteristics the Autoencoder learned to represent.

Autoencoder, as a reconstruction-based anomaly detection method, provides a generic way for modeling the normal behavior, without the need for many assumptions on the data distribution [TSKK19]. It is not affected by the used attributes, since attributes that do not have relationships to the other attributes are ignored in the encoding step [TSKK19]. On the other hand, the Autoencoder performance can be affected by large number of attributes and the inherent curse of dimensionality, since the reconstruction error is computed based on the distance between $\hat{\mathbf{x}}$ and $\mathbf{x}$ in the original space of attributes.

Deep Distance-based Anomaly Detection (REPEN)

A deep learning-based method that is designed to deal with anomaly detection tasks with very limited number of labeled anomalies is known as REPEN [PCCL18]. It provides a transformation of high-dimensional data into a low-dimensional space to allow an easier and more efficient distance-based outlier detection [PCCL18]. Given a set of $\mathcal{N}$ data points $\mathcal{X} = \{x_1, x_2, ...x_N\}$ with $x_i \in \mathbb{R}^D$ and an arbitrary distance-based anomaly scoring function $\Phi(\mathcal{X}) \longrightarrow \mathbb{R}$ that uses distances in a random data subsample to define such score, the goal is to learn a representation function $f(\mathcal{X}) \longrightarrow \mathbb{R}^M$ with $M \ll D$ where for an outlier x_i and an inlier x_j, $\Phi(f(x_i)) > \Phi(f(x_j))$ [PCCL18]. This allows learning a feature representation where anomalies have a larger nearest neighbor distance in a random data subsample than normal data instances.

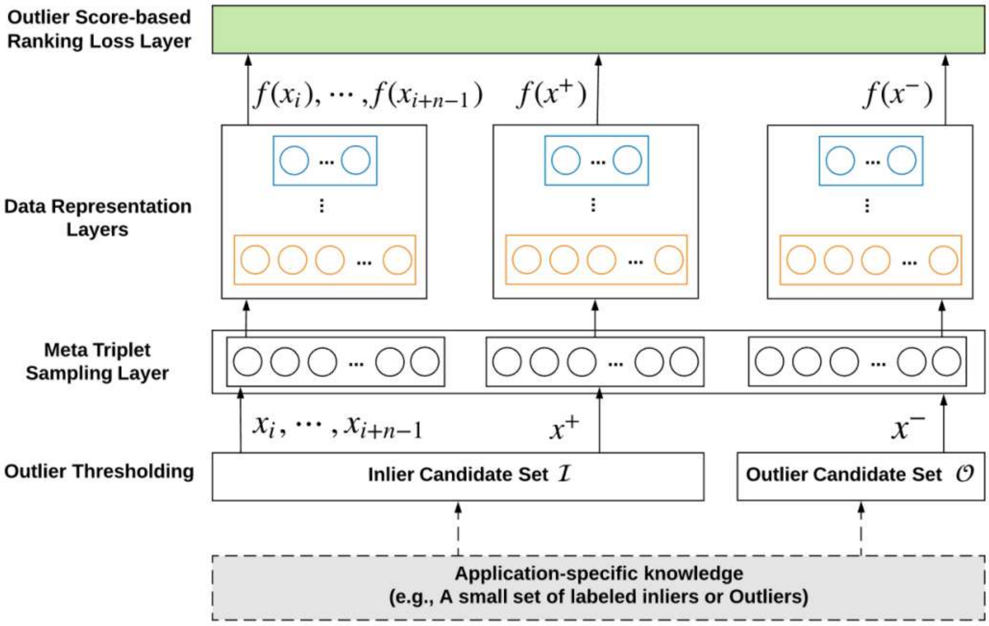

Outlier Score-based Ranking Loss Layer
$f(x_i), \cdots, f(x_{i+n-1})$
$f(x^+)$
$f(x^-)$
Data Representation Layers
Meta Triplet Sampling Layer
$x_i, \cdots, x_{i+n-1}$
x^+
x^-
Outlier Thresholding
Inlier Candidate Set $\mathcal{I}$
Outlier Candidate Set $\mathcal{O}$
Application-specific knowledge
(e.g., A small set of labeled inliers or Outliers)

Figure 2.10: Illustration of the framework of REPEN [PCCL18]

An illustration of the REPEN framework can be seen in Figure 2.10. It first performs outlier thresholding to split the data into inlier and outlier candidates. Then, meta triplet samples $T = (< x_i, ..., x_{i+n-1} >, x^+, x^-)$ are generated by randomly selecting n objects from the inlier set as query set $\mathcal{Q}$, one object from the inlier set and one object from the outlier set. This triplet is then passed to data representation layers. The data representation is further learned by a function f that is composed of one or multiple hidden layers. The optimization is performed based on an outlier score-based ranking loss defined as: $L\left(\Phi\left(f_\theta(x^+)|\mathcal{Q}\right), \Phi\left(f_\theta(x^-)|\mathcal{Q}\right)\right)$ where Θ is a random distance-based scoring function and L is a loss function.

REPEN can work in both supervised and unsupervised setups [PCCL18]. It is flexible to incorporate information about labeled inliers and outliers by including the corresponding data points to the candidate sets.

Deviation Network

Deviation Network (DevNet) [PSvdH19] is a deep learning-based method that focuses on learning end-to-end anomaly score prediction. DevNet is designed to deal with limited labeled anomalies and leverage them to directly learn the anomaly score. Given $N + K$ data points $\mathcal{X} = \{x_1, x_2, ...x_N, x_{N+1}, ..., x_{N+K}\}$ with $x_i \in \mathbb{R}^D$ and $K \ll N$. The set of unlabeled data is defined as $\mathcal{U} = \{x_1, x_2, ..., x_N\}$ while the set of labeled instances has a very small size and is defined as $\mathcal{K} = \{x_{N+1}, x_{N+2}, ..., x_K\}$. The goal is to learn an anomaly scoring function $\Phi(\mathcal{X}) \longrightarrow \mathbb{R}$, where given an anomalous data point x_i and a normal data point x_j: $\Phi(x_i) > \Phi(x_j)$.

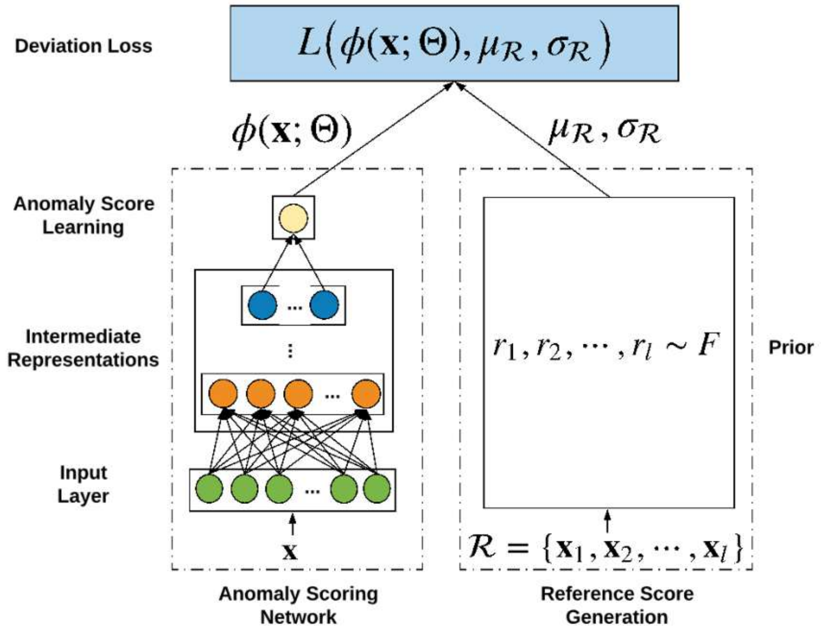

Deviation Loss
$L\left(\phi(\mathbf{x};\Theta), \mu_{\mathcal{R}}, \sigma_{\mathcal{R}}\right)$
$\phi(\mathbf{x};\Theta)$
$\mu_{\mathcal{R}}, \sigma_{\mathcal{R}}$
Anomaly Score Learning
Intermediate Representations
Input Layer
$\mathbf{x}$
$r_1, r_2, \cdots, r_l \sim F$
Prior
$\mathcal{R} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_l\}$
Anomaly Scoring Network
Reference Score Generation

Figure 2.11: Illustration of the framework of DevNet [PSvdH19]

The DevNet framework can be seen in Figure 2.11. First, the input is passed through an anomaly scoring network to obtain an anomaly score for each input. To guide the learning of such scores, a generator is used to provide a reference score defined as the mean of the anomaly scores for a set of l randomly selected normal objects denoted as $\mu_{\mathcal{R}}$ (and $\sigma_{\mathcal{R}}$ the corresponding standard deviation). The reference score is determined by a prior probability F. The loss function L takes as input $\Phi(x)$, $\mu_{\mathcal{R}}$ and $\sigma_{\mathcal{R}}$ to guide the optimization. The loss function aims to reach anomaly scores that significantly differ from $\mu_{\mathcal{R}}$ in the upper tail for anomalies and are as close as possible to $\mu_{\mathcal{R}}$ for normal data points. The loss function is defined as: $L\left(\Phi(x;\theta), \mu_{\mathcal{R}}, \sigma_{\mathcal{R}}\right) = (1-y)|dev(x)|+y.max(0, a-dev(x))$ where $dev(x) = \left(\Phi(x,\theta) - \mu_{\mathcal{R}}\right)/\sigma_{\mathcal{R}}$

Deep Autoencoding Guassian Mixture

Deep Autoencoding Gaussian Mixture Model (DAGMM) is another unsupervised deep learning anomaly detection method [PTE+20]. It consists of two main components: a compression network and an estimation network. First, dimensionality reduction of the input is performed through the compression network using an Autoencoder. The compression network provides both a low dimensional representation given by the Autoencoder and the features derived from the construction error which are in turn passed to the estimation network. The estimation network performs density estimation under the framework for GMM [Han82]. An illustration of a DAGMM model is shown in Figure 2.12.

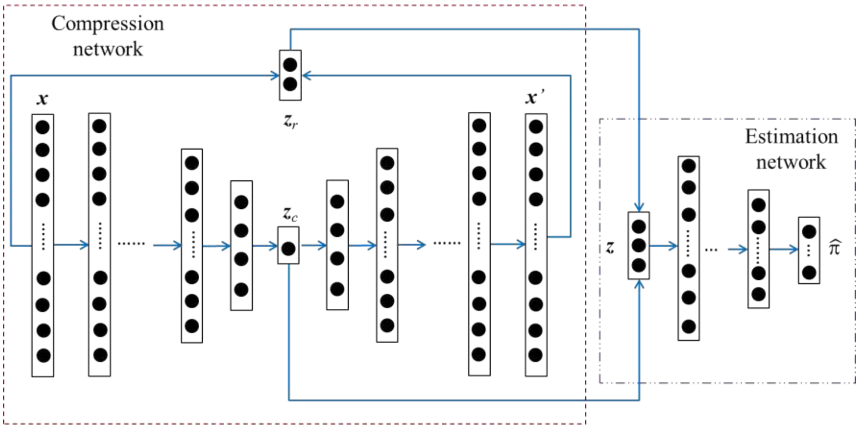


Figure 2.12: Overview of the DAGMM architecture [PTE+20]

DAGMM gives an end-to-end training where the estimation network provides a membership prediction so that the parameters in GMM can be estimated without alternating procedures such as Expectation–Maximization (EM) [PTE+20]. While Autoencoder learn by minimizing the reconstruction error that serve as anomaly score, DAGMM simultaneously minimize the reconstruction error from the compression network and sample energy from the estimation network. By conducting experiments on several public benchmark data sets, [PTE+20] shows that DAGMM provides significant improvement of up to 14% in F1 score over other techniques such as OC-SVM.

2.2.5 Other methods

One-Class Methods

In addition to the previously discussed methods, one-class classification can be used for anomaly detection. Instead of learning the distribution of the normal class, one-class approaches model the boundary of the normal class. One-Class Support Vector Machine (OC-SVM) [SWS+99] is a popular semi-supervised algorithm for outlier detection. It aims to separate the set of normal data points from the origin [GGAH14]. It works by penalizing any points that are not well separated from the origin while aiming to maximize the distance between the origin and the hyperplane separating normal and anomalous data points [GGAH14]. At the end of the optimization, the hyperplane can be used as a decision boundary to separate normal instances from anomalous instances.

Ensemble Methods

Another popular category of methods that is widely used for various data mining and machine learning applications is known as ensemble analysis. It consists of combining the outputs of multiple, often weak, algorithms to generate a unified output [Agg13]. When applied to anomaly or outlier detection, it is referred to as *outlier ensembles* [Agg13]. One popular unsupervised ensemble technique is known as Isolation Forest (IF) [LTZ08].

Similar to decision trees, an isolation tree is constructed in top-down fashion. At the start, the root node of the tree is assumed to contain all data points. The algorithm starts by choosing a random attribute and split the data based on a randomly selected threshold within the value range of the attribute resulting in two children nodes. This process is recursively repeated until each node contains a single data instance. Data points in dense regions require a much higher number of splits compared to sparse regions and therefore the generated tree is usually unbalanced. The anomaly score of data instance in a given tree is therefore defined based on this observation and is equal to the depth of its corresponding node in the tree. This can be seen in Figure 2.13, where light green nodes represent common normal data instances, dark green nodes are less common normal instances and red nodes are anomalies [JLLK20]. An isolation forest is an ensemble of multiple isolation trees. The final anomaly score is obtained by averaging scores from all trees.

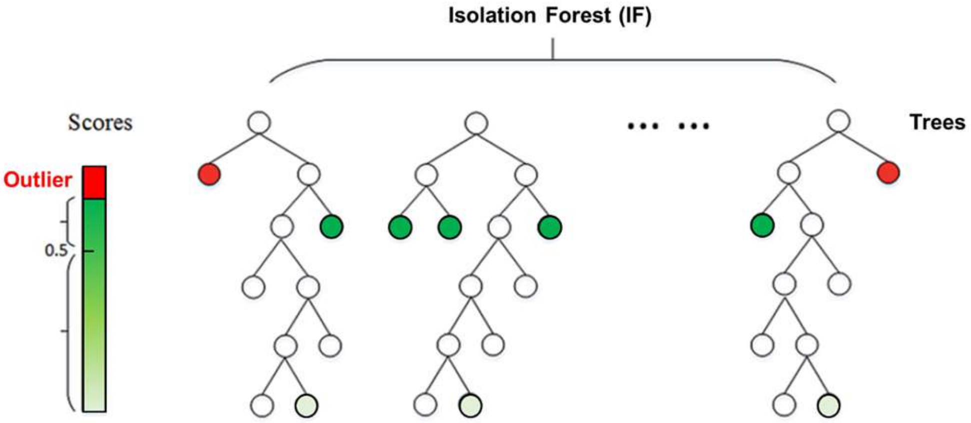


Figure 2.13: Illustration of an isolation forest [JLLK20]

Another popular type of ensemble models are boosting algorithms. Such methods work by combining a set of weak learners, a classifier that performs better than random guessing, to build a strong learner in order to improve the predictive performance. Decision trees are usually used as weak learners [Rok19].

One of the earliest boosting implementations that show wide adoption is known as Adaptive Boosting (AdaBoost) [Sch99]. AdaBoost assigns weights to data points, setting higher weights on instances that are difficult to classify by the already trained classifiers. It then sequentially adds new learners, which will implicitly focus on the more difficult patterns due to the weights. This results in weights associated with difficult samples keep increasing until the ensemble obtains an algorithm that can correctly predict them. Inference is then performed using majority voting of the weak learners, each weighted by its corresponding accuracies.

Another method that builds on top of AdaBoost is known as Gradient Boosting [Fri02]. It works by sequentially adding predictors to an ensemble where each one corrects the error of its predecessor. However, instead of changing the weights like AdaBoost, Gradient Boosting trains on the residual errors of the previous learner.

An illustration of how the Gradient Boosting algorithm works can be seen in Figure 2.14. A new tree is added at a time, which is equivalent to learning a new function $f_i^{(}X, \theta_i)$ to fit the residual of the previous prediction. During prediction, for a given data instance, there will be a corresponding leaf node in each tree and the value of such node corresponds to a score. Summing up all scores from all trees provides the prediction value of the provided instance.

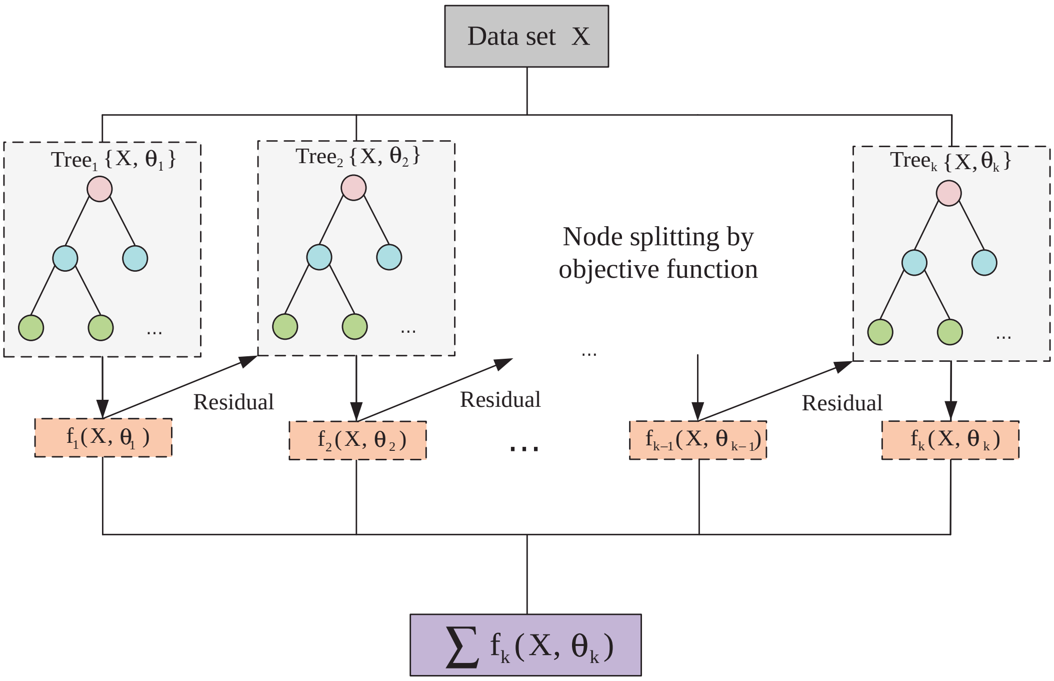


Figure 2.14: Illustration of GB [GZW$^+$20]

Extreme Gradient Boosting (XGBoost) [CG16] is a popular implementation of the gradient boosting method. It provides a few improvements that yield more accurate approximations compared to Gradient Boosting. In addition, unlike Gradient Boosting, it uses second-order gradients of the loss function and includes regularization that enhances the model generalization capabilities.

2.3 Applications

With a wide range of applications, anomaly detection has seen global adoption across multiple domains. In this section, examples of application areas where anomaly detection plays a major role are presented. The focus will be on application areas that are relevant to this thesis.

One of the major issues in the financial sector is the unauthorized access and usage of credit or debit cards. With the increasing number of transactions and activities, fraud is becoming more difficult to identify and therefore requires more sophisticated solutions [Gee15]. The involved data usually comprise of user and transaction specific records such as user ID, amount, or time between consecutive activities [CBK09]. Fraudulent transactions are often associated with high payments, high rate of purchase, or unusual purchases. Anomaly detection methods have been used in two different ways: by-owner and by-operation [CBK09]. In the by-owner setting, each credit card user is profiled based on their usage history and any new transaction is compared to the user's profile. In the by-operation approach, anomalies are detected within a set of transactions taking place at a specific geographic location.

Another popular application of anomaly detection is intrusion detection in network systems. Intrusion is the result of an attack launched by outside hackers in order to obtain unauthorized access to a network, to subsequently disrupt the its functionality or to steal sensitive data [CBK09]. To detect, prevent or mitigate these attacks, intrusion detection solutions monitor network traffic and detect suspicious activities. These systems usually involve an anomaly detection technique that uses network traffic data. The data is usually high dimensional and characterized with a temporal aspect (e.g. a sequence of network connections representing a session), even though most techniques and applications ignore that aspect [CBK09].

Anomaly detection also plays an important role in health care, where important application areas include disease diagnosis and monitoring of patients [MMH17]. Disease diagnostic usually makes use of medical records to detect abnormalities in a patient's behavior or vitals. Anomalies within the data might reflect an abnormal health condition – but can also be due to measurement errors or instrument inaccuracies. Medical records usually include patient's related information such as age, weight, and blood group [CBK09]. The data corresponding to healthy individuals are usually known, and therefore semi-supervised approaches are usually used for medical-related applications.

The data involved in the discussed applications are usually characterized with their highly distributed nature. For example, for spotting diseases from patient's data, health records from multiple health care providers are required to ensure high diversity and representativeness allowing the model to generalize well. Similarly, the data needed for fraud detection are usually located across different entities (e.g. different bank branches, or insurance companies) holding different types of data. Being able to use such data from various sources in a distributed manner is of great importance and provides value for all individual entities. It helps sharing knowledge and achieving anomaly detection solutions with a high degree of accuracy.

At the same time, the data needed for the discussed applications contain highly sensitive information that usually cannot be easily shared between different entities. It is therefore important to employ privacy-preserving learning methods.

CHAPTER 3

Collaborative Learning

Most modern machine learning applications require large sized data sets. Such data sets are usually hard to obtain and thus frequently, only a limited amount of data is available instead. In multiple settings, high quality labeled data that are a result of high efforts from domain experts is only available within an organization or a specific geographical location. In addition, different organizations may hold different types of data. For instance, certain health care providers might deal with diseases that others do not or networks in some organizations are being attacked by a new type of intrusion that others need to learn about. The transfer of such data is usually not possible due to data confidentiality and ownership restrictions especially in the case of sensitive operational data. This results in data fragments that cannot be easily merged.

Having data silos represents a serious impediment to training accurate machine learning models and therefore there is a real need for solutions that allow making use of distributed data sets without the need to collect them in a centralized location.

In this section, we start by discussing machine learning processes and the different levels at which privacy may be preserved. Then, popular privacy-preserving collaborative learning solutions are presented and discussed. Special attention is given to methods that are applicable to training anomaly detection models.

3.1 Privacy-preserving machine learning

Alan Westin [Wes68] was one of the first to define information privacy as "the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to other". [MV17] evaluated multiple definitions, and concluded that the main idea of information privacy is to have control over the collection and handling of one's personal data.

Recent data breaches and privacy violation incidents have significantly increased the concerns of violating privacy while using personal and sensitive data [CLY17, SZA+20]. This motivated the development and adoption of techniques for preserving privacy in machine learning systems [XBJ21]. This in turn resulted in the emergence of Privacy-Preserving Machine Learning (PPML) systems that are machine learning systems equipped with defense measures for protecting user privacy and data security [LTJZ20].

[XBJ21] suggests the process model of a typical machine learning pipeline shown in Figure 3.1, with different processes, data owners, and third-party provided resources. The process model involves four stages. Data preparation is the first stage, where data are collected and preprocessed. Data are generated and owned by an entity having the role of a *data producer*, and then passed to the modeling phase performed at a certain *computational facilities*. The computational facilities may be either a trusted third-party or totally owned by the data producer or the data consumer. If the data producer uses their own computational facilities, the model can be either trained and evaluated locally (scenario *T1* in Figure 3.1) or trained in a collaborative or distributed manner (scenario *T2*). Otherwise, the data have to be sent to a third party owning the computational facilities (scenario *T3*). The model training and evaluation stage involves training a machine learning model and evaluating its performance. The next stage is model deployment, where the model is either provisioned to the *model consumer* or deployed at third-party facilities. Once deployed, the model may be used to obtain predictions on new data. Once the model is deployed and is available for usage, a *model consumer* owning new data instances is expecting to make inferences using the model.

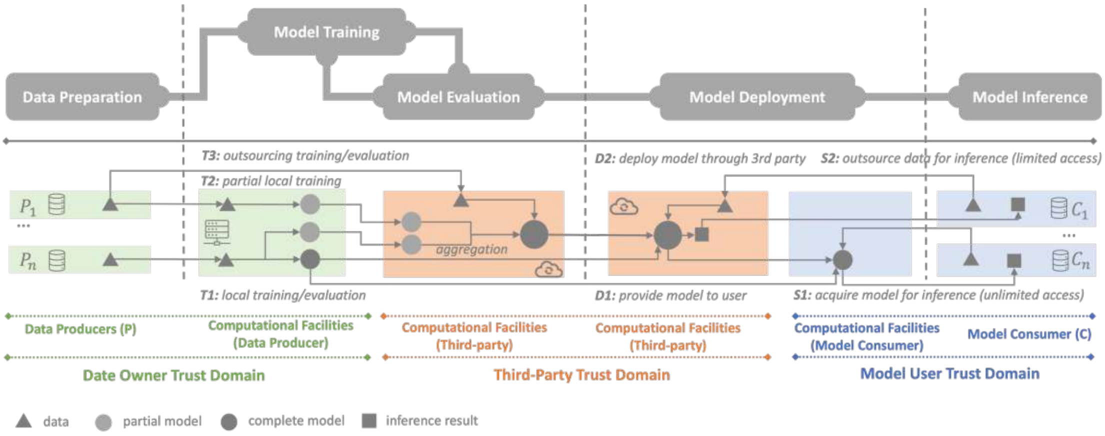

Model Training
Data Preparation
Model Evaluation
Model Deployment
Model Inference
T3: outsourcing training/evaluation
T2: partial local training
D2: deploy model through 3rd party
S2: outsource data for inference (limited access)
P_1
P_n
aggregation
C_1
C_n
T1: local training/evaluation
D1: provide model to user
S1: acquire model for inference (unlimited access)
Data Producers (P)
Computational Facilities (Data Producer)
Computational Facilities (Third-party)
Computational Facilities (Third-party)
Computational Facilities (Model Consumer)
Model Consumer (C)
Date Owner Trust Domain
Third-Party Trust Domain
Model User Trust Domain
data
partial model
complete model
inference result

Figure 3.1: Illustration of the different processes in a machine learning pipeline and the corresponding trust domains [XBJ21]

Based on the four phases of a typical machine learning pipeline, [XBJ21] suggests that at each of them privacy may be preserved, thus leading to:

(a) Privacy-preserving data preparation
(b) Privacy-preserving model training

(c) Privacy-preserving model deployment
(d) Privacy-preserving inference

This thesis focuses on the first two stages, and therefore only methods related to these are discussed.

3.2 Data synthesis

According to the defined PPML stages at which privacy can be preserved, data preparation is the first one. It consists of multiple transformations applied to the raw data so that they are more suitable for further usage. Various methods for preserving privacy at the data level exist. One of the most popular approaches is to perform anonymization using techniques, such as k-anonymity and the related l-diversity or t-closeness [XBJ21]. Other approaches are based on differential privacy techniques, where a noise is injected in the data set formally ensuring that information is preserved and cannot be leaked [Dwo08]. These techniques rely on modifying properties of the data – which may not be desired for some applications. In addition, some of these methods are prone to de-anonymization attacks [BDR18]. An alternative that became popular recently is to generate synthetic or artificial (yet representative) data, based on the original data. This is known as *synthetic data generation* and by preserving the properties and distribution of the original data, it facilitates sharing it while enhancing privacy [CRS20].

Synthetic data are generated using a set of functions and algorithms and both size (number of records) and quality (error characteristics) can be controlled [CP09]. This helps not only in generating synthetic data that are similar to the original data, but also to produce specifically tailored data that might be helpful in certain applications [CP09].

One of the first efforts to apply data synthesis for privacy-related reasons goes back to 1993 [Don93]. This work developed a method to synthesize census response data, preserving anonymity of the households. Various methods from machine learning have been successfully applied to synthetic data generation. [Rei05] was the first to use CART to generate partially synthetic data that is helpful for data sets with certain missing information or when the data quality is not sufficient for a given application. A tree is first fitted on the attribute that has structural missingness using the whole data. Each incomplete instance is then assigned to a leaf in the tree and imputed by sampling from donors within the same leaf. The donor pool can be controlled by further growing the tree given a specific minimum leaf size. Over the last years, several statistical methods have emerged: Synthetic Data Vault (SDV) [PWV16], Synthpop [NRD16], DataSynthesizer [PSH17], simPop [TMKD17], and Synthia [MN21]. In addition, there are a few commercial tools such as Mostly AI[1] and Syntho[2].

Synthpop in particular is known to be able to reproduce the main features of the data set without the need for exploratory analysis [RND18]. It starts by fitting a model that

[1] https://mostly.ai/
[2] https://www.syntho.ai/

describes the input data and then uses it to generate new data instances. In the official implementation of Synthpop[3], the CART model is used by default[BFOS17].

If the CART method is used, a classification or regression tree following the binary recursive partition procedure is fitted. To generate a synthetic data instance, Synthpop replaces some or all observed values of a given instance at a specific node by values from a randomly drawn donor from the node members.

Recently, multiple deep learning-based methods have been introduced for data synthesis. The most prominent ones are Variational Autoencoder (VAE) [KW22] and Generative Adversarial Networks (GAN) [GPM+14]. A Variational Autoencoder is an Autoencoder that uses variational inference to regularize the encoding distribution and prevent overfitting. In contrast to Autoencoder, VAE provides a way to describe data instances in latent space by providing a probability distribution instead of a single value in the bottleneck layer. A GAN model consists of two neural networks: a generator that generates data and a discriminator that validates the authenticity of the generated data. Once the networks are sufficiently trained, they will be able to generate synthetic instances that mimic the real data.

[LM22] provides an evaluation of various data synthesis methods when applied to data for training anomaly detection models. State-of-the-art methods including SDV, Synthpop, and DataSynthesizer are applied to credit card fraud and medical data to generate synthetic data that are in turn used to train multiple anomaly detection methods such as Autoencoder, GMM, and Isolation Forest. Results indicate that Synthpop outperforms other methods, and it was concluded that it represents a good choice to employ for anomaly detection tasks.

3.3 Federated learning

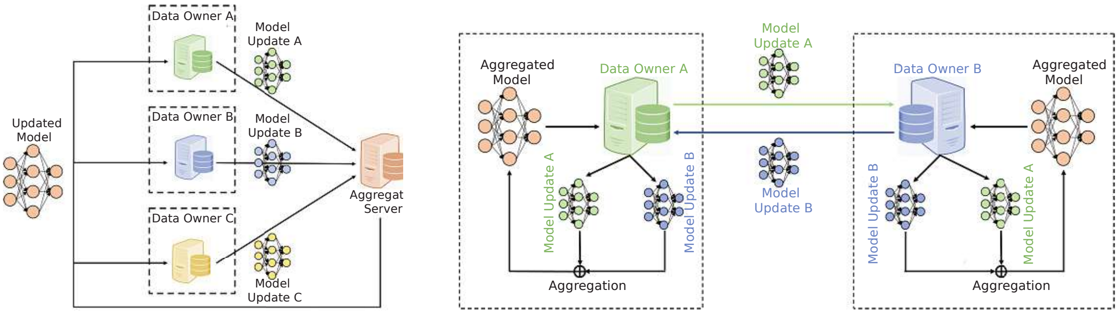
As discussed above, having data silos represent a serious impediment to developing large-scale machine learning solutions. But instead of collecting data from various sources in a centralized location, it would be better to seek a solution that allows making use of the distributed data without the need to transfer them. An idea would be to train a local model where the data reside and use such models to reach a consensus for a common, global model. This represents the idea behind what is called *federated machine learning*, which was introduced by in 2016 by [MMR+17]. It was initially introduced for an edge-server architecture, to periodically update a collective text auto-complete language model on mobile phones, by taking the average of all parameters across the local models. This has become one of the widely used aggregation methods and is known as Federated averaging (FedAvg). This way, the data reside at the edge, and only the model is communicated to the centralized location. The models are aggregated into a global model, and then sent back to all edge devices to be used during inference. The models are encrypted during transfer for confidentiality reasons.

[3] https://www.synthpop.org.uk/

[KMA+21] defines federated learning as “a machine learning setting where multiple entities (clients) collaborate in solving a machine learning problem, under the coordination of a central server or service provider”. To better characterize federated learning, [LTJZ20] suggests that the following requirements are met:

(a) There are at least two parties (or nodes), each holding a certain amount of data and that are interested in jointly training a machine learning model
(b) As part of the training process, the data never leave the client
(c) The models are transferable under a certain encryption scheme.
(d) The predictive performance of the resulting global model is a good approximation of the model trained on the aggregation of all data at a centralized location.

A federated learning setup may or may not involve an aggregation server. If a central aggregation server is available (Figure 3.2a), an initial model (e.g. randomly initialized) is first sent to the local data owners. For models like Isolation Forest and XGBoost, the process starts by fitting a model on the local data and no initial model is required. At each client, the model gets trained using the locally available data and the resulting model parameters are sent back to the centralized server. The coordinator server aggregates the parameters (using e.g. FedAvg) and sends the global model back to the nodes. This process is repeated until a certain stopping criterion is met (e.g. the maximum number of iteration is reached). In another setting that does not involve a coordinator (Figure 3.2b), clients communicate directly between each others without the help of third party.



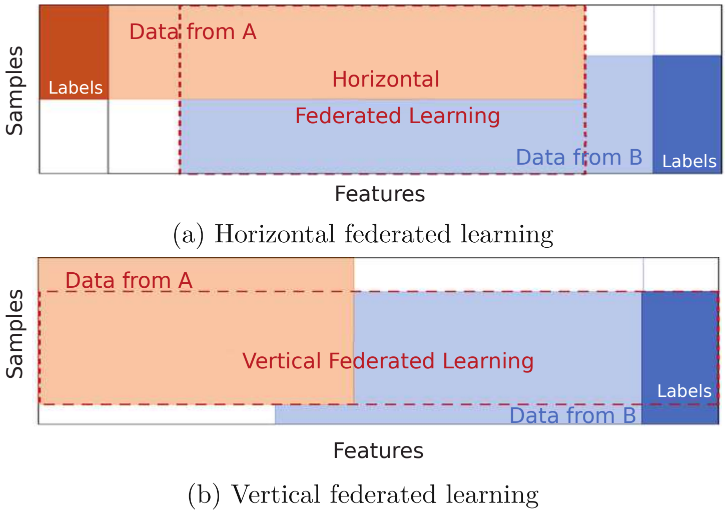
(a) Client-server architecture

(b) Peer-to-peer architecture

Figure 3.2: Federated learning architectures examples [LTJZ20]

3.3.1 Types of federated learning

Federated learning can be grouped into different categories based on how the data are partitioned across the clients. When data owners share overlapping data features, but they hold different data instances, this is referred to as horizontal federated learning [LTJZ20]. It is also known as sample-partitioned or example-partitioned federated learning [KMA+21]. An illustration of this scenario can be seen in Figure 3.3a. On the other hand, one talks about vertical federated learning when the clients share overlapping data samples, but with different features, as can be seen in Figure 3.3b. It is also known as feature-partitioned federated learning [KMA+21].



(a) Horizontal federated learning

(b) Vertical federated learning

Figure 3.3: Main federated learning categories [LTJZ20]

The horizontal federated learning scenario is more commonly encountered in real-life applications [LTJZ20]. An example would be bank branches that are located in different geographical regions and therefore have different user groups, which are described by the same characteristics. This thesis focuses only on horizontal federated learning and therefore, only topics that are relevant to such scenario are discussed in the next sections.

3.3.2 Model aggregation methods

When federated learning was introduced in [MMR+17], models are trained using Federated averaging (FedAvg). The pseudocode for this algorithm is presented in Algorithm 3.1.

FedAvg allows partial participation of the clients in the training, by only selecting a fraction of them in each round (reflected in Algorithm 3.1 by $m = C \times K$). A single communication round corresponds to the process of the clients synchronizing with the server (by e.g. uploading the local model weights in the case of FedAvg) and the global model being updated. At each communication round, it is assumed that both the local and global models have the same structure and their parameters can therefore be directly averaged and updated.

3.3.3 Effect of data distribution

As part of the federated learning process, stochastic gradient descent (SGD) is used at each step local training step to minimize the local empirical risk [ZLL+18]. In order to ensure that the SGD is an unbiased estimate of the full gradient, it is necessary to ensure an independent and identically distributed (i.i.d.) training data [Bot10]. However, it

Algorithm 3.1: Federated Averaging (FedAvg) - Adapted from [ZXLJ21]

```
Input:
$K$: the number of clients
$B$: the size of the local mini-batch
$T$: the total number of communication rounds
$E$: the number of local training epochs
$\eta$: the learning rate

Server:
Initialize global model parameters $\theta_0$
for each communication round $t$ in $\{1, 2, ..., T\}$ do
    Select $m = C \times K$ clients with $C \in [0, 1]$
    for each client $k$ in $\{1, 2, ..., m\}$ do
        Download $\theta_t$ to Client $k$
        Do Client $k$ update and receive $\theta_k$
    end
    Update global model $\theta_t = \sum_{k=1}^{m} \frac{n_k}{n} \cdot \theta^k$
end

Client $k$ update:
Replace local model $\theta^k \leftarrow \theta_t$
for local epoch in $\{1, ..., E\}$ do
    for batch $b \in [1, B]$ do
        $\theta^k \leftarrow \theta^k - \eta \cdot \nabla L_k(\theta^k, b)$
    end
end
return $\theta^k$
```

is not always possible to assume having i.i.d. data at each client within a distributed setup [ZLL+18]. In fact, in most applications the data are non-i.i.d., since they might be related to a specific user, particular geographic location, or given time window.

Within a federated learning setting, given a data set consisting of x features and y labels, we distinguish two levels of sampling: sampling a client i from the distribution of available clients $\mathcal{Q}$ and drawing an instance (x, y) from the client's local data distribution $\mathcal{P}_i(x, y)$. Non-i.i.d. data is usually associated with different data distributions ($\mathcal{P}_i$ and $\mathcal{P}_j$) for different clients i and j. It is also possible to have changes in the distributions $\mathcal{P}_i$ and $\mathcal{Q}$ which introduces another level of statistical heterogeneity.

To proceed, it can be observed that using the chain rule, the joint probability distribution $\mathcal{P}_i(x, y)$ can be rewritten as $\mathcal{P}_i(y|x)\mathcal{P}_i(x)$ and $\mathcal{P}_i(x|y)\mathcal{P}_i(y)$. In other words, the probability of observing the feature x and the label y is equal to the probability of observing y given that x occurred multiplied by the probability of observing x.

[KMA+21] defines five main aspects in which data may deviate from being i.i.d. i.e. $\mathcal{P}_i \neq \mathcal{P}_j$ for two different clients i and j:

(a) Feature distribution skew or covariate shift: The marginal distribution $\mathcal{P}_i(x)$ can vary within the clients even if $\mathcal{P}_i(y|x) = \mathcal{P}_j(y|x)$ for all clients i and j. This may be the case e.g. when data is generated from different users accomplishing the same task in different ways.

(b) Label distribution skew or prior probability shift: the marginal distribution $\mathcal{P}_i(y)$ can vary within the clients even if $\mathcal{P}_i(x|y) = \mathcal{P}_j(x|y)$ for all clients i and j – which means that the labels of the target variables are unevenly distributed across clients. For example, certain target values might be tied to a certain geographic location, and therefore are specific to a limited number of clients.

(c) Same label but different features or concept drift: $\mathcal{P}_i(x|y)$ vary across clients while $\mathcal{P}_i(y) = \mathcal{P}_j(y)$ for all clients i and j. This means that a given label y may be the result of very different features x over multiple clients. For instance, images of the same objects may vary drastically for different weather conditions, e.g. might be partially covered with snow.

(d) Same features but different label or concept shift: $\mathcal{P}_i(y|x)$ vary across clients while $\mathcal{P}_i(x) = \mathcal{P}_j(x)$ for all clients i and j. A typical example of this is sentiment analysis, where the same text might reflect different sentiments among different groups of people.

(e) Quantity skew or imbalance: The amount of data locally available might vary drastically across clients.

Data from real-world applications might involve one or multiple aspects of being non-i.i.d. and it is essential to account of their effects when running federated learning experiments.

3.3.4 Federated anomaly detection

Since its introduction, federated learning has been considered in multiple application domains [LTJZ20]. These include telecommunication and edge computing [LHD+20, XWW+21], security [THX22, MKP+22, ASA+21], finance [YZY+19, LTJZ20], sales [WXDL22], and health care [BCM+18, LMX+19, XGS+21].

While anomaly detection has been always considered to be one of the most challenging tasks in data mining [Gab10], combining it with federated learning adds multiple layers of complexity and results in an even more challenging task. While federated learning shows high adoption in various applications using different types of data, its usage for anomaly detection remained limited with little research performed in that area. In this section, we highlight some of the available work on federated anomaly detection with focus on tabular data.

Finance

Financial institutions are usually restricted by government regulations, especially when it comes to privacy and protection of personal data. Adopting cutting-edge technology such as machine learning or cloud services while complying with such regulations may be a real challenge. Let's take the example of a bank or an insurance company that aims to build a fraud detection system. Relying on data located within a specific branch or geographical location is usually not enough to build a sufficiently accurate system. Usually, there is a need for more diverse data that allows the model to generalize well. This generally requires accessing data from sparse geographical locations, which is in many of the cases not possible due to legal restrictions.

Federated learning represents a promising solution to this problem, as it allows to train models across multiple parties without the need to share the data. For instance, [YZY$^+$19] evaluates the application of federated learning for the task of credit card fraud detection and shows that it's possible to obtain good model performance without the need to share the data.

[CM22] provides a comparative study between centralized and federated learning for anomaly detection on financial and medical data sets. The considered methods involved different anomaly detection methods (MLP, GMM, and Isolation Forest) covering the different label availability scenarios: supervised, semi-supervised, and unsupervised. The data is split into 15 and 30 subsets in order to simulate a distributed scenario with variable number of clients. Results show that while federated MLP provides good precision, recall, and F2 score that are comparable to the centralized ones, the other methods do not achieve similar performance.

Intrusion detection

With the increasing dependency on digital systems, the number of devices and private networks is steadily increasing and becoming more interconnected. This resulted in the spread of cyber crime, especially that is becoming more financially lucrative. Therefore, more data are being collected to build reliable intrusion detection systems. However, it is not possible to communicate data from single users to a centralized location due to privacy and security reasons. The introduction of federated learning provided a solution to make use of machine learning techniques on distributed data without threatening the privacy and security of users.

[SH21] provides an implementation of stacked Long Short-Term Memory (LSTM) networks in a federated setup for anomaly detection on IoT sensor data. The considered data consist of sensors event logs and energy use values distributed across 180 devices. The proposed solution shows better performance in detecting both collective and contextual anomalies compared to other baseline methods like centralized logistic regression and federated logistic regression. In addition, the model shows fast convergence and robustness to changes under different configurations such as changing the number of LSTM layers.

[PZJL22] evaluates a federated anomaly detection solution for detecting anomalies in network traffic. The data consists of cellular traffic records that are extracted from packet level and session level. An LSTM-based Autoencoder is then trained in a collaborative manner. The resulted model shows better performance compared to other reconstruction-based anomaly detection method.

[HBL+21] suggests a federated architecture to detect cyberattacks in time series data for an industrial system. A VAE-LSTM model is trained across multiple manufacturing locations at the edge. The system is evaluated on a specific use case involving time-series data from a gas pipeline factory and also on other data sets involving different industrial applications. The suggested system shows a significant improvement in bandwidth efficiency while achieving high detection performance.

Health care

Another domain that showed high adoption of federated learning is the medical field [LTJZ20]. Driven by the goal to minimize human error and reduce labor costs, medical institutions have been increasingly using machine learning for various tasks. However, one major impediment to a broader and more efficient use of such technologies have always been the difficulty to collect large amount of data. Due to the highly sensitive nature of medical records, the data must always reside at the originating location and cannot be shared. This results in small data sets that are owned by different institutions and are geographically dispersed.

In order to overcome this issue and allow different institutions to collaborate, federated learning has been used with success for various tasks [LTJZ20]. For example, [BCM+18] developed a federated learning setting to train a model that predicts future hospitalizations for patients with heart-related diseases using electronic health records data. [LMX+19] proposes a federated learning system that performs segmentation of brain scans and show that it is possible to achieve segmentation results that are comparable to centralized learning without the need to share patient's data.

Other applications

[NVP22] suggest a new methodology for unsupervised federated learning in a dynamic environment. The proposed methodology consists of two steps. In the first step, clients are grouped based on their corresponding majority patterns. This is performed by training a preliminary anomaly detection model to determine the inlier/outlier split of the local data. Clients that agree on their inliers/outliers proportion exchange their models and join the same community. Each community is then collaboratively trained in a federated manner. This method is evaluated for the OC-SVM using the MNIST and MNIST fashion data sets. Results for the conducted experiments show clear improvement in terms of ROC AUC score compared to training the models on locally available data only.

Some of the work, discussed in this section, already provides an evaluation of federated anomaly detection. In particular, [YZY+19] and [CM22] perform an evaluation of the predictive performance of federated learning on some anomaly detection benchmark data sets. However, the performed experiments are limited in terms of algorithms, data sets, assumption on the data distribution, and baselines used for comparison. This thesis, on the other hand, provides a more comprehensive study on privacy-preserving anomaly detection, including federated learning. It also investigates various algorithms (for different data availability cases), data sets, and data splitting scenarios (i.i.d. and non-i.i.d.).

CHAPTER 4

Methodology

Based on the problem defined in Section 1.2 and the outcomes of our literature review, experiments with the selected privacy preserving methods are defined. These experiments involve training various anomaly detection models in different settings, including synthesis-based learning and federated learning.

In this section, the methodology for the experiments and the choices made during the experiment design are discussed in details. We start by describing the selected data sets, and provide a brief exploratory data analysis for each. Afterwards, the resampling techniques selected to be used in supervised learning are briefly discussed. Then, the selected anomaly detection algorithms are discussed for all training scenarios. Finally, both the data synthesis-based learning and the federated learning settings are described.

4.1 Overview

The literature review provides an overview on the state-of-the-art methods for anomaly detection (cf. Section 2.2), recent advances in federated learning (cf. Section 3.3), and related work that evaluates anomaly detection methods in a federated manner (cf. Section 3.3.4) . Based on that, data sets and algorithms to be used during the experiments are selected, focusing on data sets that are frequently used in related work. The selected models are then trained and fine-tuned on the data sets with the goal to achieve results that are comparable to the ones in literature.

In order to simulate local data at the client side, the original data sets are split considering different statistical heterogeneity scenarios. This is done taking into account various parameter, such as the number of clients and the target class distribution. Each model is then trained and fine-tuned on the obtained subsets, simulating the local training setting. An illustration of the process involved for this training scenario can be seen in Figure 4.6.

Afterwards, the federated setup is defined, and all algorithms are adapted accordingly. Various experiments are then conducted simulating real-life federated scenarios. An overview of the federated learning setting is shown in Figure 4.8.

Synthetic data-based learning, another collaborative learning approach, is also considered. Here, data at the client side are used to generate synthetic data, which in turn are sent to a server to train models in a centralized manner. An overview of this training setting is illustrated in Figure 4.7.

For all experiments, various performance metrics are calculated to allow evaluation and comparison of the obtained results. Significance testing is used to draw conclusion on the predictive performance of the different methods [Die98].

4.2 Data sets

With anomaly detection being one of the most popular and challenging tasks in machine learning, multiple data sets are available. For instance, [PSCvdH22] provides a collection of 21 publicly available data sets with real anomalies. The presented data sets are characterized by high dimensionality and increased complexity. [HHH$^+$22] also provides 57 data sets covering various application domains and having different characteristics.

In order to select data sets for the experiments, we limit ourselves first to tabular data sets that are popular within the anomaly detection community. Further, only data sets from applications that raise privacy concerns are considered, as only for those, applying privacy-preserving methods is justified. In order to ensure experiments that are representative of real-life scenarios, the following criteria should be met within the selected data sets:

- Different sizes and dimensions are present
- Various application domains are considered
- Different contamination rates (the percentage of samples in our data to be anomalous) are present

Following this approach, four data sets, belonging to three different application domains, are selected. These data sets are described in the following subsections.

4.2.1 Fraud detection

The first selected data set is known as the “Credit Card Fraud Detection” data set[1] and was provided by [PCJB15]. This data set contains financial transactions made by European credit card holders via online websites [AB15]. For confidentiality reasons, features have been transformed using Principal Component Analysis (PCA). The meaning of most variables is therefore not relevant. The data set contains 284,807 transactions and is highly imbalanced, as only 492 instances (equivalent to 0.173 %) are fraudulent.

[1] https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

The data set has 31 attributes, including the following:

- Time: time at which the transaction took place
- Amount: amount spent during the transaction
- 28 PCA transformed features: positive and negative real values resulting from applying PCA on the original features
- Class: target attribute defining whether the transaction is fraudulent (1) or not (0)

Due to the big difference in scale between the PCA transformed attributes and the other ones, min-max scaling is applied. In addition, we observed that time does not provide any useful information and therefore has been removed as part of the preprocessing.

4.2.2 Intrusion detection

For intrusion detection, NSL-KDD[2] is used. This data set represents an improved version of the earlier KDD CUP 99 data set[3] [SWW$^+$99], which has been introduced in 1999 and has been since widely used by researchers to evaluate anomaly detection methods [TBLG09]. NSL-KDD contains 148,517 instances and 43 features, already split into a training set with 125,973 instances and a test set of 22,544 instances. The target feature defines whether an instance represents an attack or not. There are 40 unique attacks that are grouped into four categories:

1. Denial of Service Attack (DoS): an attack aiming to make a service inaccessible by occupying significant amount of memory or state resources.
2. User to Root Attack (U2R): an exploit in which an attacker with a normal user account ends up gaining access as root by exploiting vulnerabilities in the system.
3. Remote to Local Attack (R2L): an attack in which an intruder who is initially able to send packets to a remote computer, but initially does not have permission to access it, exploits vulnerabilities to gain a user access.
4. Probing Attack: an attack in which information about the network is gathered with the purpose to get around the security controls.

According to [TBLG09], the features in NSL-KDD may be classified into three groups:

1. Basic features: the set of attributes that are extracted from TCP/IP connection.
2. Traffic features: this includes attributes that are computed within a given time window interval and can be further split into:
   (a) Same host features: information about the connections within the last two seconds that have the current host as a destination host.
   (b) Same service features: information about the connections over the last two seconds for a given network service on the destination host.
3. Content features: while Dos and probing attacks involve multiple connection attempts over a short period of time, R2L and U2R attacks are usually embedded

[2] https://www.unb.ca/cic/datasets/nsl.html

[3] https://archive.ics.uci.edu/ml/datasets/kdd+cup+1999+data

in the packet data and involve only a single connection. In order to detect such kind of attacks, additional features are required, e.g. the number of failed login attempts and percentage of connections to the same service/different services/different hots.

The number of instances for each class can be seen in Table 4.1.

Table 4.1: Number of instances per attack category for NSK-KDD

Attack Category	Normal	DoS	Probe	R2L	U2R
Number of instance	77,054	53,387	14,077	3,880	119

Working with NSL-KDD data set requires using multi-class classifiers, since it includes four attack categories. However, anomaly detection is usually formulated as a binary classification problem. Creating new "view" of the data set by only selecting a single attack category along the non-attack cases alleviates this issue, and is in line with other researchers working on anomaly detection using the NSL-KDD data [Bab15, AC15, DPS$^+$18]. For the purpose of this thesis, two data sets are created:

- NSL-KDD Probe: contains only the normal instances and instances from the Probe class. In total, it has 9,1131 data points with 77,054 being normal and 14,077 attack instances.
- NSL-KDD R2L: contains only instances corresponding to either the normal or Probe class. It has the size of 80,934 with 77,054 data instances belonging to the normal class and 3,880 instances classified as attacks.

As part of the preprocessing, one-hot encoding is applied to all categorical features and all features are min-max scaled. The resulted data set has 97 features for NSL-KDD Probe and 56 features for NSL-KDD R2L. The difference in the number of features is a result of dropping those with single value for all instances corresponding to a given target class.

4.2.3 Anomaly detection in health care

The last data set is the Thyroid Disease data set[4] provided by [DG17]. In particular, Annthyroid (also known as Ann-thryoid), a version of the thyroid data set that is suitable for training artificial neural networks, is used. In contrast to the original Thyroid data set, it only includes numeric features, where all categorical attributes are encoded. It contains medical records from 7,200 patients with the goal to determine whether a patient is hypothyroid. Hypothyroidism is a condition where the thyroid gland does not produce enough thyroid hormone in the bloodstream. The target class contain three values: normal (not hypothyroid), hyperfunction, and subnormal functioning.

The data set includes 22 attributes providing the following information:

[4] https://archive.ics.uci.edu/ml/datasets/thyroid+disease

- Basic patient information: includes among others age, sex, whether the patient is sick or not, whether the patient is pregnant or not, whether the patient is on thyroxine or antithyroid medication, whether the patient had thyroid surgery or not.
- Thyroid hormone measurements: values obtained from blood test including:
  - THS: the thyroid stimulating hormone levels.
  - T3: Triiodothyronine levels in the blood.
  - TT4, FTI, and T4U: total Thyroxine levels, free Thyroxine index and the Thyroxine uptake.
  - TBG: Thyroxine-Binding Globulin levels in the blood.
  - In addition, there are attributes defining whether such values are measured or not.

The data set in its original format is split into 3,772 instances for training and 3,428 instances for testing. Due to the small amount of data available, both subsets are merged together. In addition, the target class is transformed by combining both hyperfunction and subnormal functioning classes, so it only provides information whether the patient is normal or not.

As part of preprocessing, all attributes but the target class are scaled.

4.3 Algorithms

Based on the literature review performed in Chapter 2, well established anomaly detection algorithms that belong to different learning scenarios (supervised, semi-supervised, weakly-supervised, and unsupervised) are selected. These methods include classical anomaly detection methods that have been used for long time in various tasks, as well as new state-of-the-art deep learning-based methods. In this section, we describe the selected methods and how they are used in the experiments.

4.3.1 Supervised anomaly detection

From the wide variety of supervised anomaly detection methods, three established methods are selected for our evaluation: Feedforward Neural Network (FFNN), logistic regression and XGBoost.

Feedforward Neural Network

The first step was to define a custom FFNN architecture for each data set. The different models are implemented using PyTorch[5]. The architectures that showed the best performance are selected and their corresponding architectures are shown in the tables 4.2, 4.3, 4.4, and 4.5.

[5] https://pytorch.org/

In order to define the model that results in the highest predictive performance, the architecture is iteratively changed by choosing a number of layers between two and ten and each time, the model is trained on the training set, and its predictive performance is recorded. The number of neurons in intermediate layers are also varied within the range of $[0.5 * input\_size, 5 * input\_size]$ with a step of $0.5 * input\_size$ to determine the best architecture for each data set.

When defining the architecutre, the considered optimizers, algorithms that update the parameters of the neural network, Adam [KB17] and SGD are used. Based on the provided predictive performance, Adam is used for the Annthyroid data set, while SGD is used for all other data sets.

Both Dice Loss and Binary Cross Entropy Loss (BCELoss) are considered as loss function. It was found that BCELoss provides better results and is therefore used as a loss function for all experiments.

Table 4.2: FFNN architecture for the credit card data set

Layer	In	Out	Activation
Fully connected	29	12	ReLU
Fully connected	12	24	ReLU
Dropout (p=0.5)*	12	24	-
Fully connected	24	12	ReLU
Fully connected	12	1	Sigmoid

*probability of a unit in the layer to be zeroed

Table 4.3: FFNN architecture for the Annthyroid data set

Layer	In	Out	Activation
Fully connected	21	42	ReLU
Fully connected	42	84	ReLU
Fully connected	84	1	Sigmoid

Table 4.4: FFNN architecture for the NSL-KDD Probe data set

Layer	In	Out	Activation
Fully connected	96	64	ReLU
Fully connected	64	48	ReLU
Dropout (p=0.5)	64	48	-
Fully connected	48	24	ReLU
Fully connected	24	16	ReLU
Layer	16	1	Sigmoid

Table 4.5: FFNN architecture for the NSL-KDD R2L data set

Layer	In	Out	Activation
Fully connected	55	96	ReLU
Fully connected	96	48	ReLU
Dropout (p=0.5)	96	48	-
Fully connected	48	24	ReLU
Fully connected	24	16	ReLU
Layer	16	1	Sigmoid

Logistic Regression

All logistic regression models are implemented using PyTorch. This was preferred over using existing implementations (e.g. from scikit-learn[6]), to be able to train on GPU and

[6] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

to allow for more flexibility.

XGBoost

XGBoost is widely popular for its good performance with tabular data [Bro16]. The official implementation of the algorithm[7] is used for all experiments.

4.3.2 Semi-supervised anomaly detection

For semi-supervised learning, Autoencoder is considered. For all data sets, the models are implemented in PyTorch. The architectures used for each data set are shown in tables 4.6, 4.7, 4.8, and 4.9.

Similar to FFNN (cf. Section 4.3.1), the architecture of each data set is selected by varying the number of layers and the neurons in each hidden layer. For each data set, even numbers of hidden layers between two and ten are considered. The number of neurons in these layers are also varied within the range of $[2, \mathit{input\_size}]$ to determine the best architecture.

Both optimizers Adam and SGD are tested and Adam is selected for all data sets since it provides the highest predictive performance. Mean Squared Error (MSE) is used as a reconstruction error (loss function) for all experiments.

Table 4.6: Autoencoder architecture for the credit card data set

	Layer	In	Out	Activation
Encoder	Fully connected	29	20	LReLU
	Fully connected	20	15	LReLU
	Fully connected	15	12	LReLU
Decoder	Fully connected	12	15	LReLU
	Fully connected	15	20	LReLU
	Fully connected	20	29	LReLU

Table 4.7: Autoencoder architecture for the NSL-KDD R2L data set

	Layer	In	Out	Activation
Encoder	Fully connected	55	35	LReLU
	Fully connected	35	20	LReLU
	Fully connected	20	16	LReLU
Decoder	Fully connected	16	20	LReLU
	Fully connected	20	35	LReLU
	Fully connected	35	55	LReLU

4.3.3 Weakly-supervised anomaly detection

From the wealy-supervised learning methods, DevNet is used. The official implementation[8] is adapted to a newer version of TensorFlow[9] and used for all experiments.

In addition to the default architecture, which includes a single hidden layer with 20 ReLU units and a single unit in the output layer, DevNet provides three other variants

[7] `https://xgboost.readthedocs.io/en/stable/`

[8] `https://github.com/GuansongPang/deviation-network`

[9] `https://www.tensorflow.org/`

Table 4.8: Autoencoder architecture for the NSL-KDD Probe data set

	Layer	In	Out	Activation
Encoder	Fully connected	96	64	LReLU
	Fully connected	64	28	LReLU
Decoder	Fully connected	28	64	LReLU
	Fully connected	64	96	LReLU

Table 4.9: Autoencoder architecture for the Annthyroid data set

	Layer	In	Out	Activation
Encoder	Fully connected	21	15	LReLU
	Fully connected	15	10	LReLU
Decoder	Fully connected	10	15	LReLU
	Fully connected	15	21	LReLU

by changing the number of hidden layers. For all experiments, the default version of DevNet is used since it provides always the best performance.

A single training epoch can be defined as a complete pass of the training data set through the algorithm where each sample was used to update the internal model parameters. During each epoch of the DevNet training loop, a certain number of mini-batches are sampled using stratified random sampling. For all experiments, a mini-batch size of 100 is considered. In order to simulate the weakly-supervised scenario with very little labels available, for each model, a maximum of five anomalous data points are provided. If there are no anomalous points available in the data set, the training happens in an unsupervised fashion.

Even though DevNet may operate in an unsupervised-fashion, it was mainly designed to address problems where very limited number of labeled instances are involved [PSvdH19]. It was therefore decided to only use DevNet in the weakly-supervised scenario.

4.3.4 Unsupervised anomaly detection

For the unsupervised setting, two methods are considered: Isolation Forest (IF) and REPEN. Even though REPEN operates in both supervised and unsupervised settings, during all experiments it is used in an unsupervised fashion.

Isolation Forest

The isolation forest implementation from scikit-learn[10] is used in all experiments. In order to select the optimal number of trees for each data set, the full training data are used to train multiple Isolation Forest models and record the predictive performance each time. A number of trees in the range $[50, 500]$ are tested with a step of 10 and the value resulting in the best predictive performance is kept. This resulted in to 350 trees for credit card, 90 for NSL-KDD Probe, 150 for NSL-KDD R2L, and 400 for the Annthyroid data set.

[10] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html

REPEN

The official implementation of REPEN[11] provided by the authors of [PCCL18] is used. The code is updated to be compatible with the current version of TensorFlow.

The default architecture of REPEN with a single hidden layer is used. The training process is performed in mini-batches with random sampling from all available data. For all experiments, 100 batches are considered per epoch.

4.4 Data splitting

The selected data sets are split into training and testing sets. For all experiments, 80% of the data are reserved for training while the other 20% are used for testing. In all experiments, the test set is left out and used to evaluate the model performance and compare the different training scenarios. The training set is first used to train a centralized model and identify optimal hyper-parameter for each method. For the local, synthetic data-based and federated learning scenarios, the training data are split into small subsets simulating a distributed setting.

As already discussed in Section 3.3.3, data distribution significantly affects the performance of federated learning. In fact, other than centralized learning, that also applies to the other learning settings defined in Section 1.2. It is therefore necessary to evaluate different data distribution scenarios. For this reason, three different non-i.i.d. scenarios in addition to the i.i.d. scenario are considered.

The i.i.d. scenario is simply simulated by splitting the full data set into n subsets. The non-i.i.d. scenarios are defined taking into account the five aspects at which data may deviate from being i.i.d. defined in Section 3.3.3. These scenarios are defined as follow:

1. *Feature-based partition*: k-means clustering (Algorithm 2.1) is used to split the data into 5 different clusters. The k-means implementation from scikit-learn[12] is used with the default settings and with a maximum number of iterations of 300. A Dirichlet distribution is then used to sample n subsets from the resulted clusters. The Dirichlet distribution, commonly used as prior distribution in Bayesian statistics [Loc75], represents a good mean of simulating real-world data distribution [LDCH21]. It is parameterized by a vector $\alpha = (\alpha_1, ..., \alpha_k)$ where a higher α gives a more dense distribution while a low α provides a more sparse distribution. The value of α is set to of 10 for this scenario.
2. *Label-based partition*: we start by equally splitting the normal data instances into n subsets simulating n clients. Anomalous instances are then randomly and unequally assigned to the created subsets; 30% of these subset do not receive any anomalous instances.

[11] https://github.com/GuansongPang/deep-outlier-detection

[12] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

3. *Label-based Dirichlet Partition (LDP)*: for n clients, each is allocated a subset of the instances corresponding to each label (normal and anomalous) according to the Dirichlet distribution. The value of α is set to of 5 for this scenario. Unlike the label-based scenario, this does not set any strict rule on labels being missing in certain clients, and it might be the case that all clients have both classes in their local data.

The Earth Mover's Distance (EMD) (also known as Wasserstein distance) is used to evaluate the similarity between the generated subsets for each scenario. EMD represents a measure of the distance between two probability distributions over a given region. For n subsets, simulating local data at the clients in a federated setting, EMD is calculated pairwise for each feature. The resulting values for each feature are then averaged to obtain a single value for comparison. The average EMD values per feature for the credit card data set in the case with 50 clients can be seen in Figure 4.1. It can be seen that for most attributes, the feature-based non-i.i.d. scenario shows the highest difference across clients. For the target attribute (Class), the label-based non-i.i.d. scenario provides the highest difference across clients.

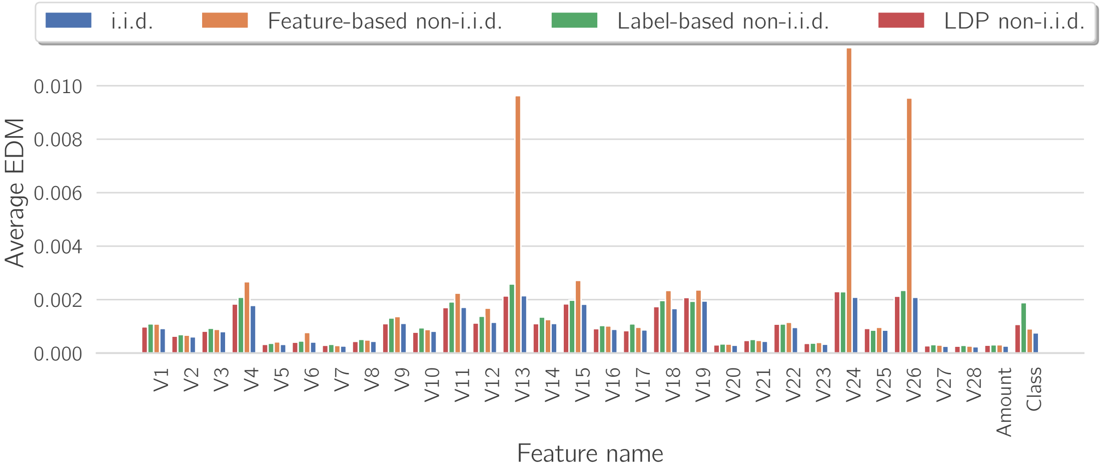


Figure 4.1: Comparison of average EMD value per feature for the 50 clients scenario with the credit card data set

The average EMD distance for some features for the Probe data set can be seen in Figure 4.2. The values for non-i.i.d. data splitting scenarios are overall higher. Unlike the Credit Card data set, the EMD values for the label-based scenario are high for most attributes.

Figure 4.3 shows that the difference across clients per feature is similar to what is observed for the Credit Card data set. Feature-based splitting scenario results in very different local data at each client for most attributes. On the other hand, the i.i.d. splitting scenario seems to provide very similar data across the different clients for all features.

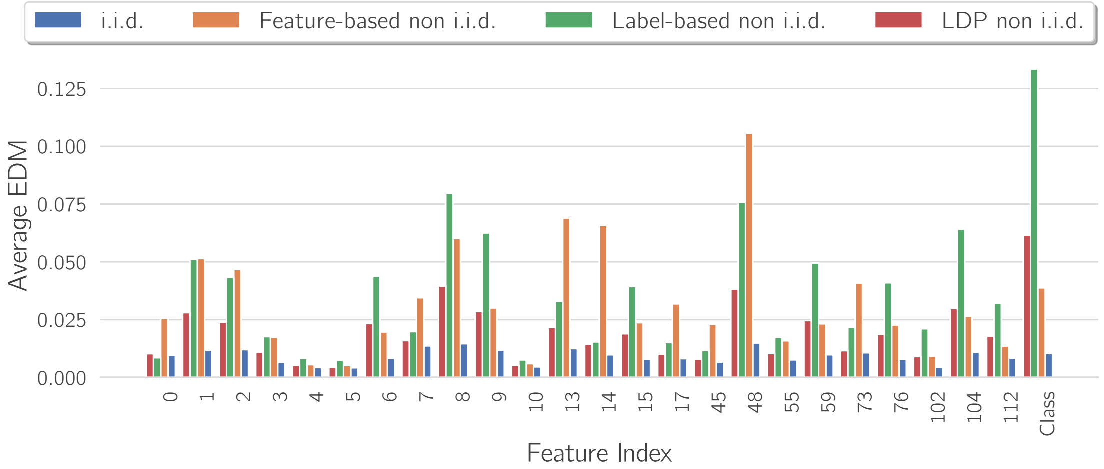


Figure 4.2: Comparison of average EMD value per feature for the 50 clients scenario with the Probe data set (Only few features)

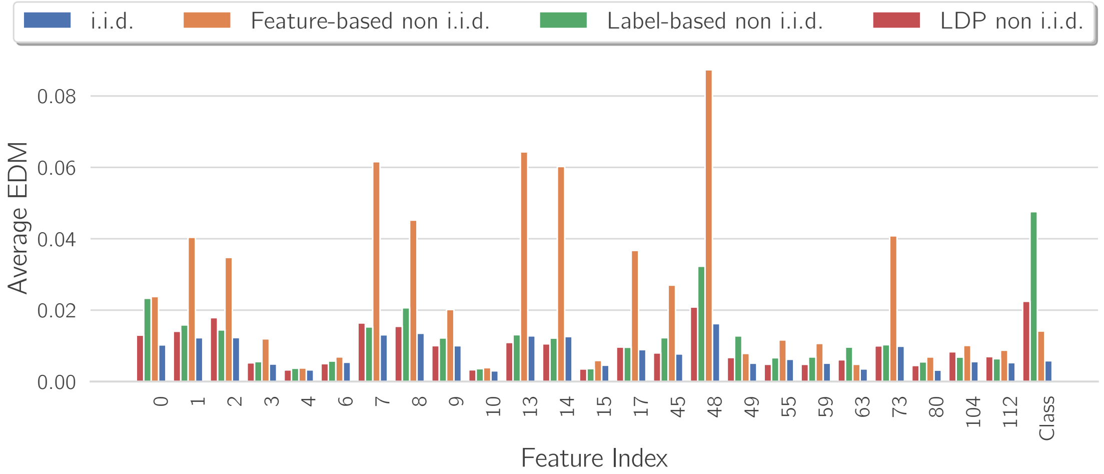


Figure 4.3: Comparison of average EMD value per feature for the 50 clients scenario with the R2L data set (Only few features)

For the Annthyroid data set, the difference across the clients is less visible, with non-i.i.d. scenarios providing slightly more different data across the clients compared to i.i.d. splitting. This can be observed in Figure 4.4

4.5 Performance evaluation

Based on the discussion in Section 2.1.3, two main metrics that are popular within the anomaly detection research community are selected: ROC and PR. These metrics are independent of the anomaly threshold and therefore provide a better mean of comparison.

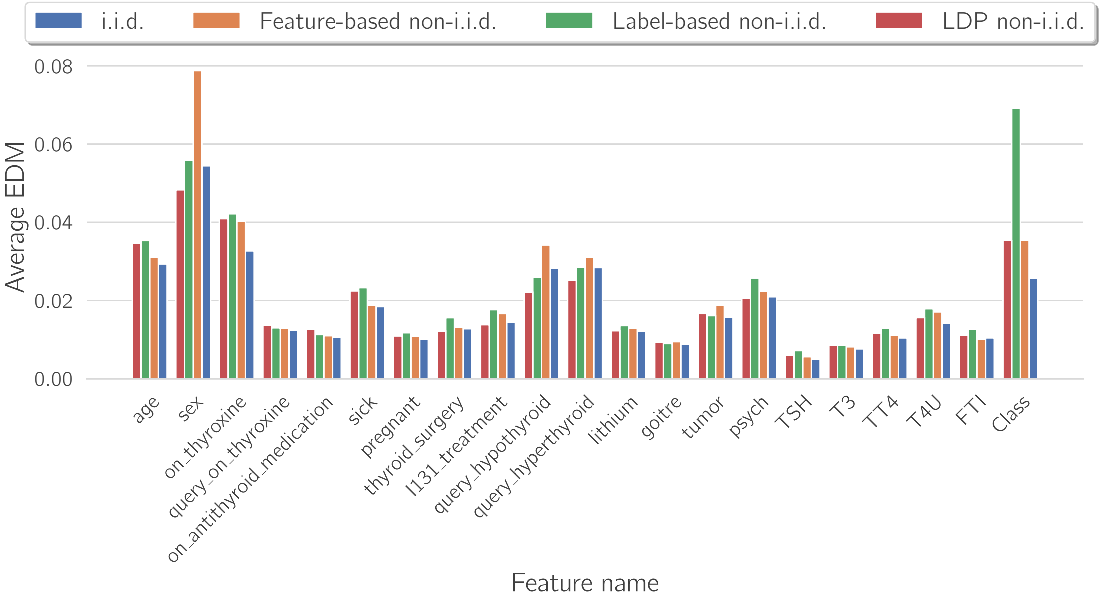


Figure 4.4: Comparison of average EMD value per feature for the 50 clients scenario with the Annthyroid data set

Due to the highly imbalanced nature of the anomaly detection problem, the variation in PR AUC is more important than ROC AUC for the different experiments. Since anomaly detection is focused on predicting anomalies (the positive class), PR AUC represents a better mean of evaluating the performance. In fact, ROC AUC might be misleading and provide high values for imbalanced applications while the model is misclassifying most of the anomalous instances. PR AUC on the other hand is better suited for the case where positive events (anomalies) are less common. While both metrics are considered during the evaluation of results, a special attention is always given to PR AUC.

As part of this thesis, the different training settings are evaluated in terms of effectiveness (i.e. predictive performance), and other aspects such as communication efficiency are not considered.

In each experiment, we set a seed value that controls all aspects of the experiment including data splitting, distributed data simulation, and model initialization. For all experiments, the training is repeated three times with three different random seed values. For each metric, the average and standard deviation of all three runs are then provided. This counters the effect of randomness and provides more representative and comparable results.

As defined in Section 1.2, four different training settings are considered: centralized, local, synthetic data-based and federated learning. Both centralized and local learning serve as reference to the other settings. Centralized learning, where models using aggregated data obtained from all clients, provides the upper limit or the ideal performance that other training settings aim to achieve[13]. On the other hand, local training give the lower limit that privacy-preserving collaborative training settings have to outperform. Since local training results in one model per client, for each performance metric, the average value over all clients is always reported. To illustrate this, an example of a typical learning curve for the federated scenario can be seen in Figure 4.5.

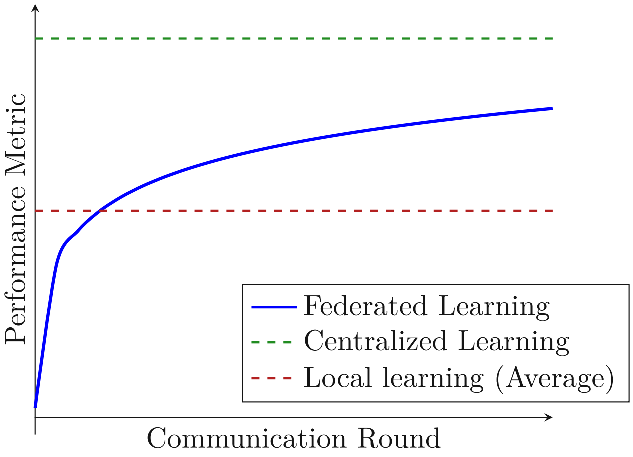


Figure 4.5: Example of federated learning curve compared to references

It can be seen that a federated model is expected to improve during the learning process and the final value falls within the reference range defined by centralized and local training. The same applies to synthetic data-based learning where a well-performing model results in a value that is above the average local learning value.

[13] However, it has to be considered that centralizing data is most of the times not even an option, due to data restrictive protection regulations. Centralized learning is thus often really just an upper limit reference point, but not a realistic alternative to federated learning.

4.6 Local data-based learning

The first training setting that serves as a baseline for the other scenarios is learning from local data available at the client side only. As already mentioned in Section 1.2, this scenario will be referred to as *local training* in the rest of the thesis.

Figure 4.6 provides an overview on the setup for all local training experiments.

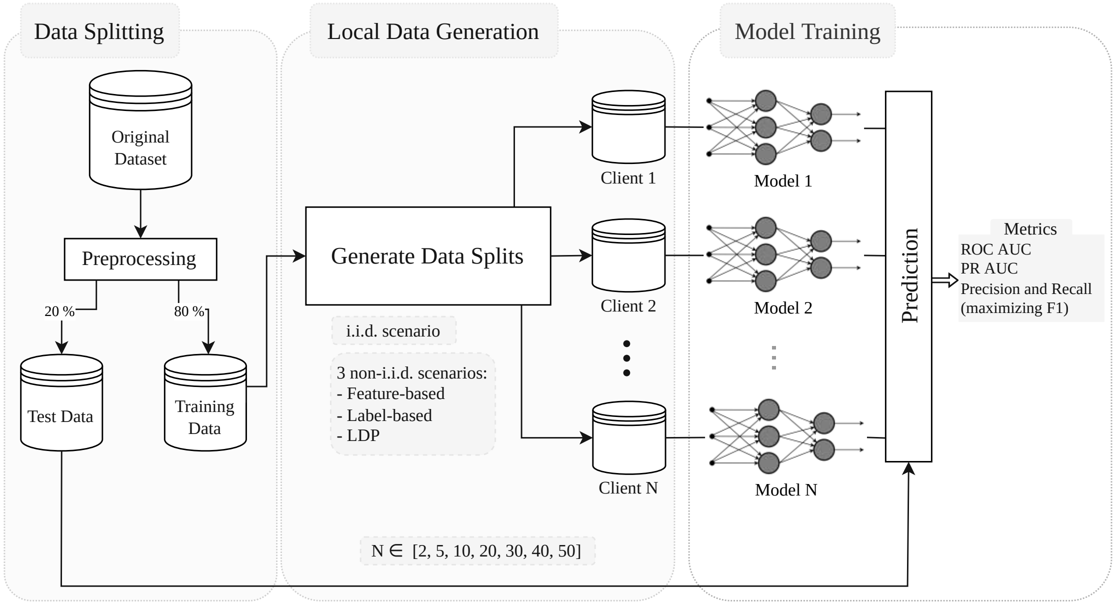


Figure 4.6: Illustration of the local training setting

After splitting the original data set into training and test sets, a distributed learning scenario is simulated by splitting the data into N clients, where the four data scenarios described in Section 4.4 are applied. For each simulated client, a single local model is then trained using only the data locally available. This is performed for each data splitting scenario and for each defined number of clients. Once a model is trained, the test set is used to perform prediction and calculate the performance metrics. As part of investigating the effect of the amount of data available at each client, the number of clients is varied with $N \in [2, 5, 10, 20, 30, 40, 50]$.

As mentioned in the previous section, results from local training are considered as a lower reference point for the other experiments. For each experiment, this reference value is the average over all clients.

4.7 Synthetic data-based learning

The first privacy-preserving learning scenario is synthetic data-based learning. In this scenario, synthetic data is locally generated at the client side, and then sent to a centralized location for aggregation. The thus collected data sets are then used to train a centralized model, which is evaluated on the test set. This process can be seen in Figure 4.7.

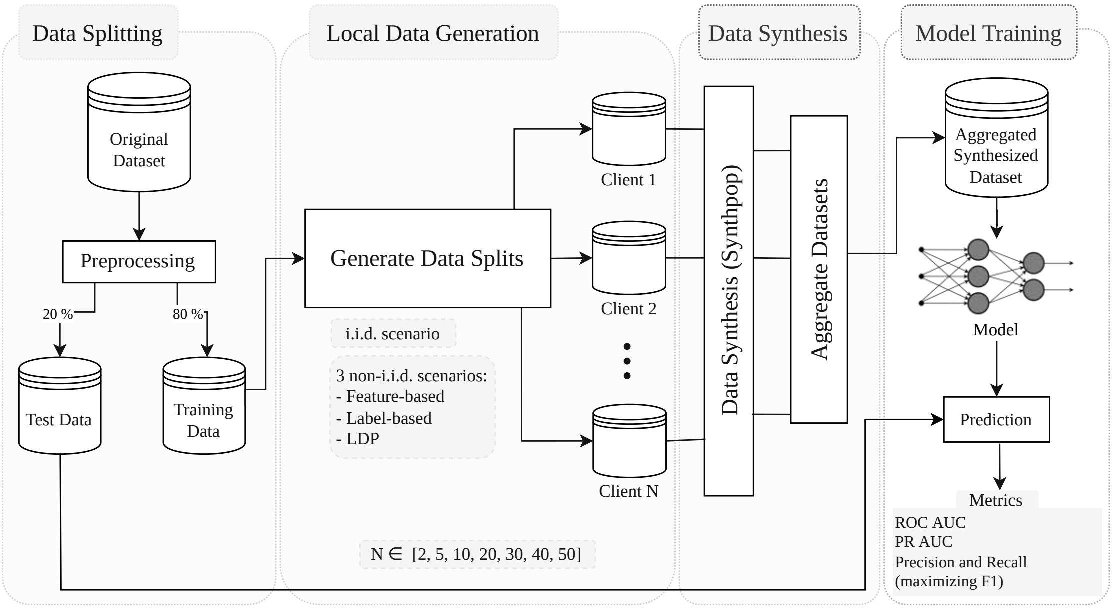


Figure 4.7: Illustration of the synthetic data-based learning setting

4.8 Federated learning

The second privacy-preserving learning setting is federated learning. For all experiments, local models are trained for 10 epochs per communication round. In total, 100 communication rounds are carried out. After the last round, performance metrics are calculated for the global federated model using the test set.

For aggregation, FedAvg is used, for its simplicity and performance. In addition, related work showed that FedAvg performs better than other aggregation methods such as FedSGD and FSVRG [NSU+18].

Due to the nature of the XGBoost and Isolation Forest algorithms, applying FedAvg is not possible and model aggregation is performed differently. In the case of XGBoost, a model is trained locally on the data available at each client, and then all of them are used during inference to make predictions. To perform inference for a given data instance, each model is used to obtain an anomaly score and all obtained values are then averaged to get a single federated anomaly score. Isolation Forest is federated by aggregating the trees from each local model. Each local model is trained on the locally available data and the trees from each model are then sent to a central location where a global model is created by adding them together.

For supervised learning, the effect of resampling through the application of random

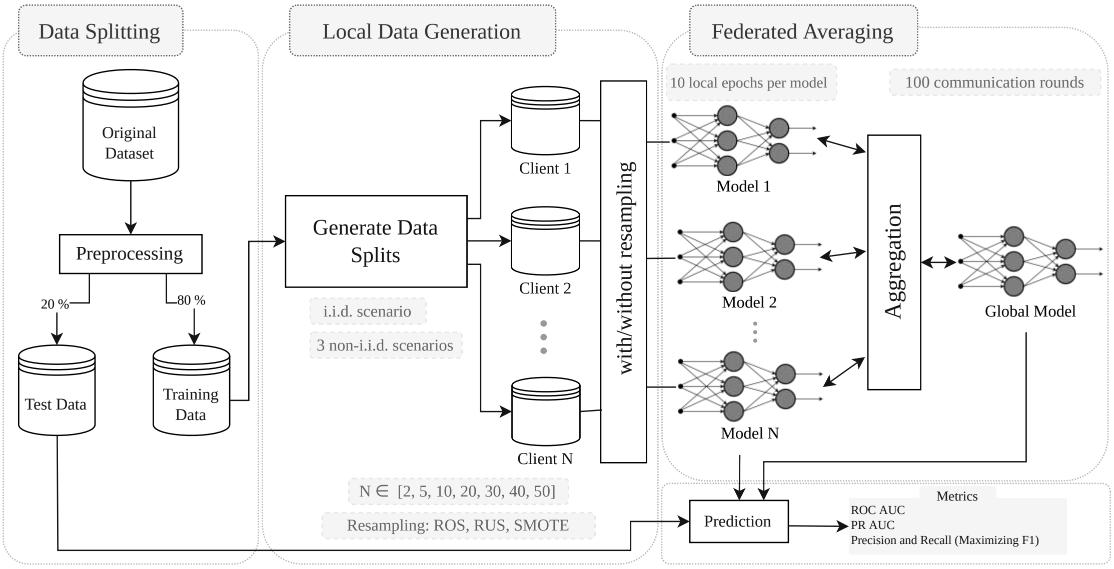


Figure 4.8: Illustration of the federated learning setting

undersampling, random oversampling, and SMOTE, all at the client level, are evaluated. These techniques help with the highly imbalanced nature of the data sets. Oversampling techniques perform data augmentation on the minority class by sampling instances from the minority class or generating new instances based on the existing ones. On the other hand, undersampling techniques work by selecting specific samples from the majority class. The implementation for all resampling techniques is taken from the package imbalanced-learn[14] [LNA16].

[14] https://imbalanced-learn.org/

CHAPTER 5

Results and Evaluation

In this section, we discuss the results from all experiments in terms of predictive performance metrics.

The selected methods are first trained in a centralized fashion. This ensures that the chosen models and their corresponding hyperparameters are able to provide good performance for the different data sets. As mentioned in Chapter 4, centralized results in addition to the average local learning results provide baselines for comparison. Even though the variation between the clients might be also of interest, it is assumed that the average value provides a good indication on the overall performance of the local learning.

As mentioned in Section 4.5, performance is mainly assessed in terms of ROC AUC and PR AUC. In addition, while evaluating and comparing results, a special attention is always given to PR AUC since it better reflects the model's performance.

Results in this section are presented in a way that allows comparing the privacy-preserving settings to learning based on only the local data. As mentioned in Chapter 4, Student's t-test is used to evaluate the significance of the federated and synthetic data-based learning results compared to local training. For this purpose, two significance levels are used: 5% and 10%. In addition, both metrics are always reported in terms of mean and standard deviation of the values obtained over all executions. The comparison between the training settings takes into account the following:

1. Different data splitting scenarios: comparing i.i.d. to non-i.i.d. scenarios and the effect of aspects in which data can deviate from being i.i.d..
2. Number of clients and the amount of data available at each client: the effect is evaluated in terms of performance metrics and also in terms of model convergence. for the learning scenario, the effect of resampling on the performance is also evaluated.
3. Type of data: comparing the results across the data sets reflecting the effect of the type and dimensionality of the data.

4. Various training scenarios: evaluate the difference between the four scenarios reflecting the effect of availability and amount of labels on the performance.

In all result tables in this section, the best score for a specific data set, number of clients and data distribution scenario is highlighted in bold font. In addition, statistically significance of federated and synthetic-based learning scores compared to local learning scores are encoded by color. The federated learning result is given in green if it is significantly better than the local learning result, while analogously, the synthetic data-based learning result is colored in blue if it is significantly better than the the local learning result. In addition, the significance level is indicated by a "+" for 5% and "++" for 10%. On the other hand, the result for federated or synthetic-based learning scenario being significantly worse than local learning is indicated by a "-" for and "- -" for significance levels of 5% and 10% respectively.

Even though a number of clients between 2 and 50 are considered during the experiments, only settings with 10 and 50 clients are presented in this section to improve readability. The detailed results for the other settings can be found in Appendix A.

5.1 Supervised learning

In this section, results from all three supervised learning methods are presented. Overall, federated learning provides a significantly better results compared to learning from local data only and also compared to learning from synthetic data. Synthetic data-based learning also shows good predictive performance in some of the settings but seems to be dependent on the type of the data.

In addition, the effect of resampling on the federated results is evaluated by applying both oversampling and undersampling at the local nodes.

In the next subsections, a detailed analysis of the obtained results is performed with a focus on the previously defined aspects.

5.1.1 Feedforward Neural Network

Results for the experiments using FFNN can be seen in Tables 5.1 and 5.2.

For the Credit Card data set, synthetic data-based learning outperforms federated learning, but both settings show significantly better performance compared to local learning for all i.i.d. and non-i.i.d. scenarios. The difference in performance becomes more significant with increasing number of clients. For instance, in the case of i.i.d. data, the PR AUC value for federated learning drops from 0.528 with 10 clients to 0.448 while the performance for the synthetic data model does not show noticeable change with the PR AUC slightly increasing from 0.638 with 10 clients to 0.669 with 50 clients.

However, for the R2L and Annthyroid data sets, synthetic data-based learning show significantly worse results compared to both local and federated learning for different number of clients. For the Annthyroid data set for example, the PR AUC values range between 0.062 and 0.172 across the various data splitting scenarios and the different number of clients.

Federated learning, on the other hand, provides consistent performance across the different data sets. In particular, for the R2L data sets with 50 clients, it shows a clear advantage over local learning with more than 50% increase in PR AUC for the different data splitting scenarios.

Table 5.1: FFNN results for all data sets with 10 and 50 clients using i.i.d. and Feature-based non-i.i.d. data: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		**Feature-based non-i.i.d.**	
Clients	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC
		Credit Card Data Set			
-	Centralized	0.927 (±0.02)	0.544 (±0.06)	-	-
10	Local	0.694 (±0.00)	0.165 (±0.03)	0.694 (±0.01)	0.146 (±0.03)
	Federated	0.895^{+}(±0.02)	0.528^{+}(±0.05)	0.894^{+}(±0.02)	0.528^{+}(±0.05)
	Synthesis	**0.930^{+}(±0.02)**	**0.638^{+}(±0.05)**	**0.932^{+}(±0.02)**	**0.639^{+}(±0.06)**
50	Local	0.509 (±0.00)	0.105 (±0.02)	0.503 (±0.01)	0.102 (±0.02)
	Federated	0.887^{+}(±0.04)	0.448^{+}(±0.06)	0.888^{+}(±0.04)	0.449^{+}(±0.06)
	Synthesis	**0.939^{+}(±0.02)**	**0.669^{+}(±0.06)**	**0.938^{+}(±0.02)**	**0.672^{+}(±0.06)**
		Probe Data Set			
-	Centralized	1.000 (±0.00)	0.998 (±0.00)	-	-
10	Local	0.997 (±0.00)	0.984 (±0.00)	0.997 (±0.00)	0.981 (±0.00)
	Federated	**0.999^{+}(±0.00)**	**0.997^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.996^{+}(±0.00)**
	Synthesis	0.988^{-}(±0.00)	0.942^{-}(±0.00)	0.981^{-}(±0.00)	0.919^{-}(±0.00)
50	Local	0.994 (±0.00)	0.966 (±0.00)	0.989 (±0.00)	0.942 (±0.00)
	Federated	**0.998^{+}(±0.00)**	**0.988^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.992^{+}(±0.00)**
	Synthesis	0.995^{++}(±0.00)	0.973^{+}(±0.00)	0.991^{+}(±0.00)	0.953^{+}(±0.00)
		R2L Data Set			
-	Centralized	0.997 (±0.00)	0.960 (±0.00)	-	-
10	Local	0.986 (±0.00)	0.753 (±0.01)	0.981 (±0.00)	0.717 (±0.06)
	Federated	**0.995^{+}(±0.00)**	**0.936^{+}(±0.01)**	**0.995^{+}(±0.00)**	**0.932^{+}(±0.01)**
	Synthesis	0.763^{-}(±0.01)	0.100^{-}(±0.01)	0.763^{-}(±0.01)	0.100^{-}(±0.01)
50	Local	0.898 (±0.00)	0.317 (±0.01)	0.891 (±0.00)	0.309 (±0.01)
	Federated	**0.981^{+}(±0.00)**	**0.662^{+}(±0.01)**	**0.982^{+}(±0.00)**	**0.679^{+}(±0.03)**
	Synthesis	0.833^{-}(±0.01)	0.146^{-}(±0.01)	0.833^{-}(±0.01)	0.150^{-}(±0.01)
		Annthyroid Data Set			
-	Centralized	0.995 (±0.00)	0.954 (±0.03)	-	-
10	Local	0.901 (±0.02)	0.782 (±0.02)	0.869 (±0.02)	0.670 (±0.03)
	Federated	**0.964^{+}(±0.01)**	**0.903^{+}(±0.05)**	**0.959^{+}(±0.01)**	**0.872^{+}(±0.04)**
	Synthesis	0.763^{-}(±0.01)	0.100^{-}(±0.01)	0.763^{-}(±0.01)	0.100^{-}(±0.01)
50	Local	0.769 (±0.03)	0.411 (±0.02)	0.768 (±0.02)	0.423 (±0.02)
	Federated	**0.849^{+}(±0.02)**	**0.679^{+}(±0.02)**	0.825 (±0.05)	**0.636^{+}(±0.07)**
	Synthesis	0.833^{+}(±0.01)	0.146^{-}(±0.01)	**0.833^{+}(±0.01)**	0.150^{-}(±0.01)

For the Credit Card data set, increasing the number of clients does not affect the synthetic data-based model. In the case of i.i.d. data, the ROC AUC and PR AUC change from 0.930 and 0.610 respectively with 5 clients to 0.939 and 0.669 with 50 clients. Similar behavior can also be observed for the other non-i.i.d. scenarios. For R2L and Annthyroid data sets, synthetic data-based learning provide very low performance for the different number of scenarios with a slight increase in PR AUC with increasing the number of clients.

The effect of the number of clients on the predictive performance is more visible in the other learning scenarios. For instance, using i.i.d. data, the average PR AUC of local models drops from 0.227 with 2 clients to 0.105 with 50 clients for the Credit Card data set and from 0.940 to 0.317 for the R2L data set. Federated learning is also affected by the increasing number of clients, and this is especially visible with R2L and Annthyroid data sets. In the case of R2L data

Table 5.2: FFNN results for all data sets with 10 and 50 clients using Label-based and LDP non-i.i.d. data: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		Label-based non-i.i.d.		**LDP non-i.i.d.**	
Clients	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC
Credit Card Data Set					
-	Centralized	0.927 (±0.02)	0.544 (±0.06)	-	-
10	Local	0.672 (±0.01)	0.120 (±0.04)	0.686 (±0.01)	0.132 (±0.03)
	Federated	0.905^{+}(±0.02)	0.566^{+}(±0.05)	0.911^{+}(±0.02)	0.592^{+}(±0.06)
	Synthesis	**0.933^{+}(±0.02)**	**0.648^{+}(±0.05)**	**0.932^{+}(±0.02)**	**0.641^{+}(±0.05)**
50	Local	0.509 (±0.00)	0.103 (±0.02)	0.507 (±0.00)	0.103 (±0.02)
	Federated	0.889^{+}(±0.04)	0.466^{+}(±0.06)	0.891^{+}(±0.04)	0.475^{+}(±0.05)
	Synthesis	**0.940^{+}(±0.01)**	**0.683^{+}(±0.06)**	**0.940^{+}(±0.01)**	**0.673^{+}(±0.06)**
Probe Data Set					
-	Centralized	1.000 (±0.00)	0.998 (±0.00)	-	-
10	Local	0.980 (±0.00)	0.891 (±0.00)	0.996 (±0.00)	0.979 (±0.00)
	Federated	**0.999^{+}(±0.00)**	**0.996^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.996^{+}(±0.00)**
	Synthesis	0.981 (±0.00)	0.919^{+}(±0.00)	0.981^{-}(±0.00)	0.919^{-}(±0.00)
50	Local	0.979 (±0.00)	0.893 (±0.00)	0.990 (±0.00)	0.949 (±0.00)
	Federated	**0.999^{+}(±0.00)**	**0.992^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.990^{+}(±0.00)**
	Synthesis	0.991^{+}(±0.00)	0.953^{+}(±0.00)	0.991 (±0.00)	0.953 (±0.00)
R2L Data Set					
-	Centralized	0.997 (±0.00)	0.960 (±0.00)	-	-
10	Local	0.897 (±0.00)	0.522 (±0.01)	0.983 (±0.00)	0.704 (±0.02)
	Federated	**0.994^{+}(±0.00)**	**0.921^{+}(±0.01)**	**0.995^{+}(±0.00)**	**0.923^{+}(±0.01)**
	Synthesis	0.767^{-}(±0.01)	0.104^{-}(±0.01)	0.763^{-}(±0.00)	0.100^{-}(±0.01)
50	Local	0.855 (±0.00)	0.301 (±0.02)	0.899 (±0.00)	0.336 (±0.02)
	Federated	**0.988^{+}(±0.00)**	**0.787^{+}(±0.03)**	**0.990^{+}(±0.00)**	**0.829^{+}(±0.03)**
	Synthesis	0.843^{-}(±0.00)	0.172^{-}(±0.01)	0.833^{-}(±0.01)	0.150^{-}(±0.01)
Annthyroid Data Set					
-	Centralized	0.995 (±0.00)	0.954 (±0.03)	-	-
10	Local	0.801 (±0.02)	0.502 (±0.02)	0.866 (±0.02)	0.664 (±0.01)
	Federated	**0.890^{+}(±0.02)**	**0.683^{+}(±0.03)**	**0.969^{+}(±0.01)**	**0.876^{+}(±0.03)**
	Synthesis	0.767^{-}(±0.01)	0.104^{-}(±0.01)	0.763^{-}(±0.00)	0.100^{-}(±0.01)
50	Local	0.729 (±0.03)	0.343 (±0.01)	0.768 (±0.02)	0.413 (±0.03)
	Federated	0.729 (±0.05)	**0.371^{++}(±0.02)**	0.812 (±0.04)	**0.584^{+}(±0.04)**
	Synthesis	**0.843^{+}(±0.00)**	0.172^{-}(±0.01)	**0.833^{+}(±0.01)**	0.150^{-}(±0.01)

set, while the ROC AUC is almost the same, the PR AUC significantly drops from 0.936 with 10 clients to 0.662 with 50 clients, for the i.i.d. scenario. The same trend can be observed for the other data splitting scenarios. A similar behavior is also observed for the Annthyroid data set. To better investigate this effect, PR AUC is plotted against the number of clients for the label-based scenario (Figure 5.1). It can be seen that the PR AUC drops from 0.864 with 2 clients to 0.371 with 50 clients. This decrease in performance is more significant compared to the local learning, where the PR AUC decreases from 0.519 with 2 clients to 0.343 with 50 clients.

In addition, increasing the number of clients affects the model convergence. Figure 5.2 shows the evolution of the PR AUC value during federated learning for the Annthyroid data set with data split in an LDP fashion. While with 2 clients the model takes around 40 communication rounds to converge, it requires more than 80 rounds to converge with 20 clients.

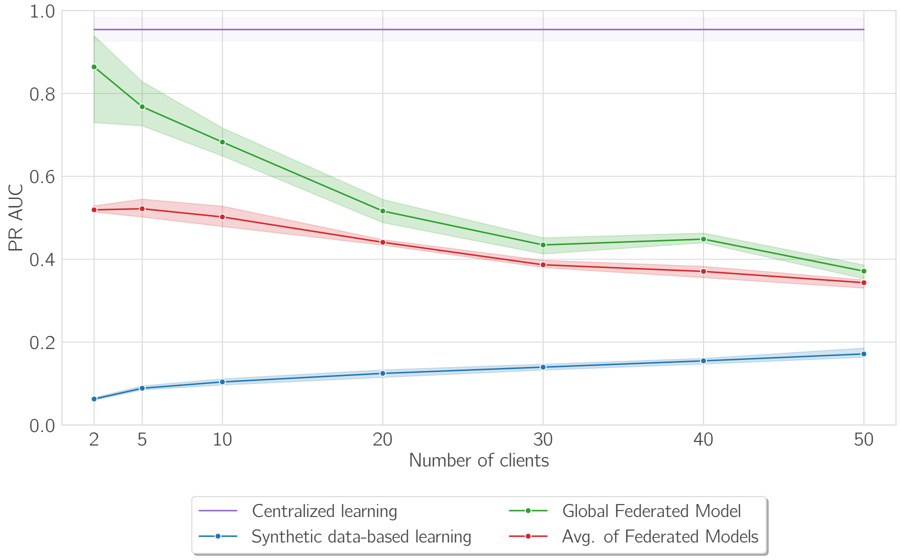


Figure 5.1: PR vs. number of clients for FFNN using Anthyroid data set - Label-based non-i.i.d.

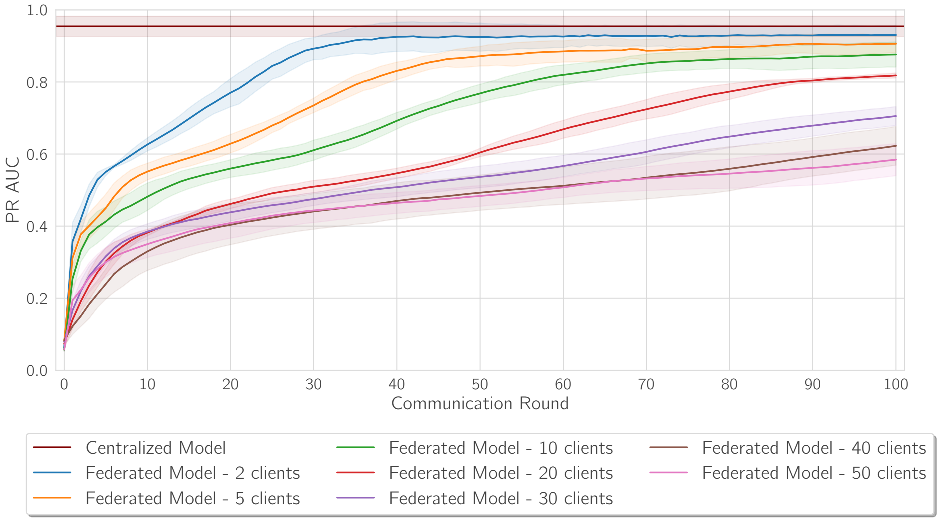


Figure 5.2: PR vs. communication round for FFNN using Anthyroid data set - Label-based Dirichlet Partition non-i.i.d.

The effect of data distribution on the predictive performance of both federated and local learning can be observed for the Annthyroid data set with the label-based non-i.i.d. scenario. For instance, with 10 clients, federated learning shows an ROC AUC and PR AUC of 0.890 and 0.683 respectively, compared to 0.964 and 0.903 for the i.i.d. scenario. The same can be observed for local learning where the ROC AUC and PR AUC drop from 0.901 and 0.782 for the i.i.d. data splitting scenario to 0.801 and 0.502 with the label-based scenario. For the other data sets, i.i.d. and non-i.i.d. scenarios show very comparable performance, suggesting little effect of the data distribution on the learning process.

5.1.2 Logistic Regression

Looking at the results for Logistic Regression shown in Table 5.3 and Table 5.4, it can be seen that overall, federated learning provides good performance for the different data sets.

Table 5.3: Logistic Regression results for all data sets with 10 and 50 clients using i.i.d. and Feature-based non-i.i.d. data: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		**Feature-based non-i.i.d.**	
Clients	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC
Credit Card Data Set					
-	Centralized	0.984 (±0.01)	0.720 (±0.05)	-	-
10	Local	**0.978 (±0.01)**	**0.707 (±0.06)**	**0.978 (±0.01)**	**0.701 (±0.06)**
	Federated	0.968 (±0.01)	0.700 (±0.06)	0.968 (±0.01)	0.699 (±0.05)
	Synthesis	0.944^{-} (±0.01)	0.680 (±0.05)	0.943^{-} (±0.01)	0.680 (±0.05)
50	Local	0.958 (±0.01)	0.644 (±0.06)	0.955 (±0.01)	0.636 (±0.06)
	Federated	**0.965 (±0.01)**	**0.701 (±0.06)**	**0.965 (±0.01)**	**0.701 (±0.06)**
	Synthesis	0.928^{--} (±0.02)	0.587 (±0.04)	0.923^{--} (±0.02)	0.580 (±0.06)
Probe Data Set					
-	Centralized	0.997 (±0.00)	0.980 (±0.00)	-	-
10	Local	0.996 (±0.00)	0.977 (±0.00)	0.996 (±0.00)	0.976 (±0.00)
	Federated	**0.996 (±0.00)**	**0.977 (±0.00)**	**0.996 (±0.00)**	**0.977 (±0.00)**
	Synthesis	0.995^{-} (±0.00)	0.969^{--} (±0.00)	0.995^{-} (±0.00)	0.972^{--} (±0.00)
50	Local	0.995 (±0.00)	0.966 (±0.00)	0.994 (±0.00)	0.964 (±0.00)
	Federated	**0.996^{+} (±0.00)**	**0.977^{+} (±0.00)**	**0.996^{+} (±0.00)**	**0.977^{+} (±0.00)**
	Synthesis	0.995 (±0.00)	0.972^{+} (±0.00)	0.995^{+} (±0.00)	0.973^{+} (±0.00)
R2L Data Set					
-	Centralized	0.985 (±0.00)	0.762 (±0.01)	-	-
10	Local	0.983 (±0.00)	0.746 (±0.01)	0.983 (±0.00)	0.747 (±0.01)
	Federated	**0.984 (±0.00)**	**0.760^{++} (±0.01)**	**0.984 (±0.00)**	**0.758 (±0.01)**
	Synthesis	0.970^{-} (±0.00)	0.622^{-} (±0.02)	0.970^{-} (±0.00)	0.619^{-} (±0.01)
50	Local	0.979 (±0.00)	0.701 (±0.01)	0.978 (±0.00)	0.699 (±0.00)
	Federated	**0.983^{+} (±0.00)**	**0.755^{+} (±0.00)**	**0.983^{++} (±0.00)**	**0.758^{+} (±0.00)**
	Synthesis	0.953^{-} (±0.00)	0.449^{-} (±0.02)	0.955^{-} (±0.00)	0.461^{-} (±0.01)
Annthyroid Data Set					
-	Centralized	0.956 (±0.01)	0.779 (±0.01)	-	-
10	Local	0.875 (±0.01)	0.588 (±0.02)	0.836 (±0.01)	0.497 (±0.00)
	Federated	0.909^{+} (±0.01)	0.625^{++} (±0.02)	0.894^{+} (±0.02)	0.596^{+} (±0.04)
	Synthesis	**0.923^{+} (±0.01)**	**0.647^{++} (±0.04)**	**0.904^{+} (±0.03)**	**0.615^{+} (±0.05)**
50	Local	0.804 (±0.02)	0.453 (±0.02)	0.766 (±0.02)	0.368 (±0.02)
	Federated	**0.868^{+} (±0.01)**	**0.543^{+} (±0.02)**	**0.859^{+} (±0.01)**	**0.542^{+} (±0.01)**
	Synthesis	0.844^{+} (±0.01)	0.501^{++} (±0.02)	0.858^{+} (±0.04)	0.498^{++} (±0.09)

For the Credit Card data set, the difference among the various training scenarios is not significant, with relatively high PR AUC variance. These difference gets however larger with increasing number of clients. Similar behavior can be observed for the Probe and R2L data sets, with the exception for the 50 clients scenario, where federated learning shows a significantly better result compared to local and synthetic data-based learning scenarios. This is more significant for the non-i.i.d. scenarios than for the i.i.d., especially with label-based split data.

Table 5.4: Logistic Regression results for all data sets with 10 and 50 clients: Label-based and LDP non-i.i.d. data: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		Label-based non-i.i.d.		**LDP non-i.i.d.**	
Clients	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC
		Credit Card Data Set			
-	Centralized	0.984 (±0.01)	0.720 (±0.05)	-	-
10	Local	0.957 (±0.01)	0.648 (±0.05)	0.961 (±0.01)	0.671 (±0.05)
	Federated	**0.968 (±0.01)**	**0.701 (±0.06)**	**0.968 (±0.01)**	**0.700 (±0.06)**
	Synthesis	0.945 (±0.01)	0.682 (±0.06)	0.945 (±0.01)	0.681 (±0.06)
50	Local	0.944 (±0.01)	0.607 (±0.05)	0.954 (±0.01)	0.666 (±0.05)
	Federated	**0.966 (±0.01)**	**0.703^{++}(±0.06)**	**0.965 (±0.01)**	**0.702 (±0.06)**
	Synthesis	0.939 (±0.01)	0.628 (±0.05)	0.930^{--} (±0.02)	0.608 (±0.06)
		Probe Data Set			
-	Centralized	0.997 (±0.00)	0.980 (±0.00)	-	-
10	Local	0.989 (±0.00)	0.937 (±0.00)	0.996 (±0.00)	0.977 (±0.00)
	Federated	**0.996^{+}(±0.00)**	**0.977^{+}(±0.00)**	**0.996 (±0.00)**	**0.980^{++}(±0.00)**
	Synthesis	0.995^{+}(±0.00)	0.972^{+}(±0.00)	0.995^{-}(±0.00)	0.973^{-}(±0.00)
50	Local	0.988 (±0.00)	0.930 (±0.00)	0.994 (±0.00)	0.966 (±0.00)
	Federated	**0.996^{+}(±0.00)**	**0.977^{+}(±0.00)**	**0.996^{+}(±0.00)**	**0.977^{+}(±0.00)**
	Synthesis	0.995^{+}(±0.00)	0.974^{+}(±0.00)	0.995 (±0.00)	0.973^{+}(±0.00)
		R2L Data Set			
-	Centralized	0.985 (±0.00)	0.762 (±0.01)	-	-
10	Local	0.914 (±0.00)	0.545 (±0.01)	0.983 (±0.00)	**0.745 (±0.01)**
	Federated	**0.984^{+}(±0.00)**	**0.729^{+}(±0.01)**	**0.984 (±0.00)**	0.740 (±0.01)
	Synthesis	0.975^{+}(±0.00)	0.668^{+}(±0.01)	0.970^{-}(±0.00)	0.604^{-}(±0.01)
50	Local	0.910 (±0.00)	0.514 (±0.01)	0.978 (±0.00)	0.694 (±0.00)
	Federated	**0.983^{+}(±0.00)**	**0.731^{+}(±0.01)**	**0.983^{+}(±0.00)**	**0.745^{+}(±0.00)**
	Synthesis	0.964^{+}(±0.00)	0.542 (±0.03)	0.958^{-}(±0.00)	0.469^{-}(±0.02)
		Annthyroid Data Set			
-	Centralized	0.956 (±0.01)	0.779 (±0.01)	-	-
10	Local	0.772 (±0.02)	0.393 (±0.02)	0.840 (±0.01)	0.504 (±0.02)
	Federated	0.894^{+}(±0.02)	0.587^{+}(±0.03)	0.907^{+}(±0.01)	0.616^{+}(±0.01)
	Synthesis	**0.924^{+}(±0.01)**	**0.643^{+}(±0.04)**	**0.915^{+}(±0.01)**	**0.636^{+}(±0.04)**
50	Local	0.729 (±0.03)	0.313 (±0.02)	0.769 (±0.02)	0.367 (±0.03)
	Federated	**0.873^{+}(±0.01)**	**0.557^{+}(±0.01)**	**0.863^{+}(±0.02)**	**0.551^{+}(±0.01)**
	Synthesis	0.855^{+}(±0.04)	0.513^{+}(±0.08)	0.847^{+}(±0.03)	0.518^{+}(±0.06)

Synthetic data-based learning show lower performance compared to local learning for the Credit Card and R2L data sets. For instance, using R2L datastes, the models trained using synthetic data show an ROC AUC and PR AUC of 0.953 and 0.449 respectively compared to 0.979 and 0.701 provided by the local models for the i.i.d. data splitting scenario.

A different behavior can be seen for the Annthyroid data set, where both privacy-preserving learning methods perform better than local learning. Figure 5.3 shows the results with label-based split data. Synthetic data-based learning performs better for lower number of clients, with the PR AUC decreasing for the case with more than 40 clients.

Compared to FFNN, the decrease in performance with increasing the number of clients is less significant for all scenarios. Logistic regression seems to provide good performance even when little amount of data is available.

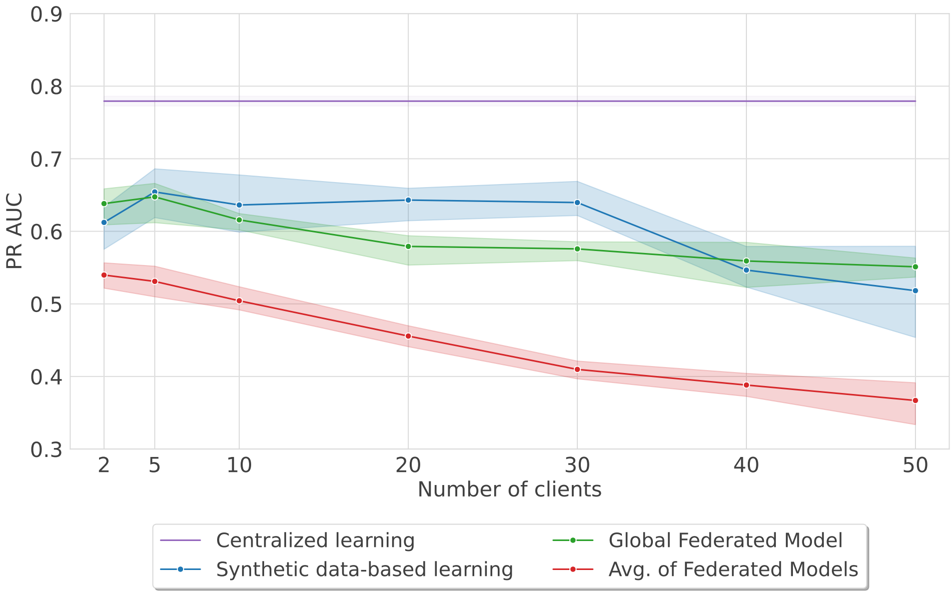


Figure 5.3: PR AUC vs. number of clients for Logistic Regression with the Annthyroid data set - LDP non-i.i.d. scenario

In addition, the convergence of federated Logistic Regression models is much faster than FFNN, and also less sensitive to the amount of local data available. Figure 5.4 shows the PR AUC learning curve for different number of clients. It can be observed that the effect of number of clients does not have large effect on the PR AUC values, with all cases converging after around 20 communication rounds.

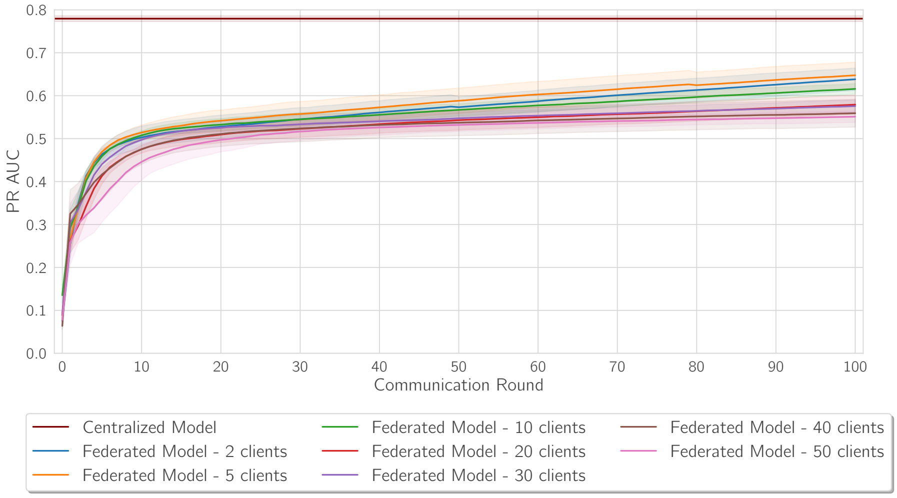


Figure 5.4: PR AUC vs. communication round for Logistic Regression with the Annthyroid data set - LDP non-i.i.d. scenario

5.1.3 XGBoost

Using XGBoost, federated learning provides significantly better performance than local learning in most cases, as shown in Tables 5.5 and 5.6. The difference in performance between the learning scenarios is however minimal especially with low number of clients. In particular, the synthetic data-based learning scenario is affected the most by the decreasing amount of data at each client. This is noticeable for the Annthyroid data set where the ROC AUC and PR AUC values decreased from 0.989 and 0.804 with 10 clients to 0.935 and 0.455 with 50 clients in the i.i.d. scenario.

Table 5.5: XGBoost results for all data sets with 10 and 50 clients using i.i.d. and Feature-based non-i.i.d. data: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		**Feature-based non-i.i.d.**	
Clients	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC
		Credit Card Data Set			
-	Centralized	0.974 (±0.00)	0.847 (±0.03)	-	-
10	Local	0.964 (±0.01)	0.765 (±0.04)	0.962 (±0.01)	0.758 (±0.04)
	Federated	**0.983^{+}(±0.00)**	**0.812 (±0.03)**	**0.982^{+}(±0.01)**	**0.810 (±0.04)**
	Synthesis	0.921^{-}(±0.02)	0.629^{--}(±0.09)	0.915^{--}(±0.03)	0.611^{-}(±0.06)
50	Local	0.946 (±0.02)	0.640 (±0.06)	0.947 (±0.01)	0.639 (±0.05)
	Federated	**0.979^{+}(±0.01)**	**0.757^{+}(±0.04)**	**0.976^{+}(±0.01)**	**0.756^{+}(±0.04)**
	Synthesis	0.928 (±0.01)	0.532 (±0.13)	0.920 (±0.02)	0.567 (±0.14)
		Probe Data Set			
-	Centralized	1.000 (±0.00)	1.000 (±0.00)	-	-
10	Local	0.999 (±0.00)	0.996 (±0.00)	0.999 (±0.00)	0.996 (±0.00)
	Federated	**1.000^{+}(±0.00)**	**0.999^{+}(±0.00)**	**1.000^{+}(±0.00)**	**0.999^{+}(±0.00)**
	Synthesis	0.998^{-}(±0.00)	0.992^{-}(±0.00)	0.999 (±0.00)	0.993^{-}(±0.00)
50	Local	0.998 (±0.00)	0.988 (±0.00)	0.997 (±0.00)	0.988 (±0.00)
	Federated	**0.999^{+}(±0.00)**	**0.997^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.997^{+}(±0.00)**
	Synthesis	0.997^{-}(±0.00)	0.985^{-}(±0.00)	0.997 (±0.00)	0.986 (±0.00)
		R2L Data Set			
-	Centralized	0.999 (±0.00)	0.990 (±0.00)	-	-
10	Local	0.998 (±0.00)	0.973 (±0.01)	0.997 (±0.00)	0.971 (±0.01)
	Federated	**0.999 (±0.00)**	**0.985^{++}(±0.01)**	**0.999^{++}(±0.00)**	**0.984^{+}(±0.00)**
	Synthesis	0.992^{-}(±0.00)	0.903^{-}(±0.02)	0.993^{-}(±0.00)	0.909^{-}(±0.01)
50	Local	0.990 (±0.00)	0.902 (±0.01)	0.989 (±0.00)	0.891 (±0.01)
	Federated	**0.997^{+}(±0.00)**	**0.958^{+}(±0.01)**	**0.997^{+}(±0.00)**	**0.957^{+}(±0.01)**
	Synthesis	0.975^{-}(±0.00)	0.661^{-}(±0.05)	0.978^{-}(±0.00)	0.710^{-}(±0.05)
		Annthyroid Data Set			
-	Centralized	0.999 (±0.00)	0.984 (±0.01)	-	-
10	Local	0.995 (±0.00)	0.934 (±0.02)	0.995 (±0.00)	0.932 (±0.01)
	Federated	**0.998^{+}(±0.00)**	**0.951 (±0.03)**	**0.998 (±0.00)**	**0.963 (±0.02)**
	Synthesis	0.989^{-}(±0.00)	0.804^{-}(±0.01)	0.990^{-}(±0.00)	0.815^{-}(±0.01)
50	Local	0.972 (±0.01)	0.804 (±0.01)	0.969 (±0.00)	0.798 (±0.00)
	Federated	**0.993^{+}(±0.00)**	**0.899^{+}(±0.01)**	**0.993^{+}(±0.00)**	**0.898^{+}(±0.00)**
	Synthesis	0.935^{-}(±0.01)	0.455^{-}(±0.06)	0.957 (±0.01)	0.574^{-}(±0.07)

The similarity in performance between federated learning and other learning scenarios, especially when little data are available at each client, can be explained by the way XGBoost models are aggregated. As mentioned in Section 4.8, model aggregation for XGBoost is performed by averaging the anomaly scores provided by all models. This might also explain the fact that the higher number of clients, the more significant is the federated model performance compared to

Table 5.6: XGBoost results for all data sets with 10 and 50 clients using Label-based and LDP non-i.i.d. data: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		Label-based non-i.i.d.		**LDP non-i.i.d.**	
Clients	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC
Credit Card Data Set					
-	Centralized	0.974 (±0.00)	0.847 (±0.03)	-	-
10	Local	0.828 (±0.01)	0.538 (±0.03)	0.963 (±0.01)	0.749 (±0.05)
	Federated	**0.982^{+}(±0.01)**	**0.813^{+}(±0.03)**	**0.981^{++}(±0.01)**	**0.812 (±0.03)**
	Synthesis	0.922^{+}(±0.02)	0.610 (±0.07)	0.913^{-}(±0.03)	0.620^{-}(±0.06)
50	Local	0.816 (±0.01)	0.460 (±0.04)	0.945 (±0.01)	0.638 (±0.05)
	Federated	**0.978^{+}(±0.01)**	**0.772^{+}(±0.04)**	**0.979^{+}(±0.00)**	**0.750^{+}(±0.04)**
	Synthesis	0.922^{+}(±0.02)	0.642^{+}(±0.02)	0.922 (±0.02)	0.611 (±0.07)
Probe Data Set					
-	Centralized	1.000 (±0.00)	1.000 (±0.00)	-	-
10	Local	0.849 (±0.00)	0.744 (±0.00)	0.999 (±0.00)	0.996 (±0.00)
	Federated	**1.000^{+}(±0.00)**	**0.999^{+}(±0.00)**	**1.000^{+}(±0.00)**	**0.999^{+}(±0.00)**
	Synthesis	0.999^{+}(±0.00)	0.996^{+}(±0.00)	0.999 (±0.00)	0.993^{-}(±0.00)
50	Local	0.848 (±0.00)	0.738 (±0.00)	0.997 (±0.00)	0.987 (±0.00)
	Federated	**0.999^{+}(±0.00)**	**0.997^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.996^{+}(±0.00)**
	Synthesis	0.998^{+}(±0.00)	0.990^{+}(±0.00)	0.997 (±0.00)	0.986 (±0.00)
R2L Data Set					
-	Centralized	0.999 (±0.00)	0.990 (±0.00)	-	-
10	Local	0.848 (±0.00)	0.696 (±0.00)	0.997 (±0.00)	0.971 (±0.01)
	Federated	**0.999^{+}(±0.00)**	**0.985^{+}(±0.00)**	**0.999^{++}(±0.00)**	**0.985^{+}(±0.00)**
	Synthesis	0.995^{+}(±0.00)	0.942^{+}(±0.00)	0.992^{-}(±0.00)	0.891^{-}(±0.00)
50	Local	0.844 (±0.00)	0.649 (±0.00)	0.989 (±0.00)	0.895 (±0.01)
	Federated	**0.997^{+}(±0.00)**	**0.962^{+}(±0.01)**	**0.997^{+}(±0.00)**	**0.957^{+}(±0.01)**
	Synthesis	0.989^{+}(±0.00)	0.879^{+}(±0.01)	0.982^{-}(±0.00)	0.755^{-}(±0.02)
Annthyroid Data Set					
-	Centralized	0.999 (±0.00)	0.984 (±0.01)	-	-
10	Local	0.846 (±0.00)	0.672 (±0.01)	0.992 (±0.00)	0.926 (±0.01)
	Federated	**0.998^{+}(±0.00)**	**0.948^{+}(±0.03)**	**0.998^{++}(±0.00)**	**0.965^{+}(±0.01)**
	Synthesis	0.992^{+}(±0.00)	0.864^{+}(±0.06)	0.992 (±0.00)	0.847^{-}(±0.02)
50	Local	0.838 (±0.00)	0.601 (±0.01)	0.972 (±0.01)	0.807 (±0.01)
	Federated	**0.994^{+}(±0.00)**	**0.905^{+}(±0.02)**	**0.993^{+}(±0.00)**	**0.893^{+}(±0.01)**
	Synthesis	0.978^{+}(±0.01)	0.704^{+}(±0.06)	0.965 (±0.02)	0.667^{--}(±0.09)

the other scenarios. With little data available at each client, federated learning seems to provide better predictive performance than the average of all local models.

Data distribution does not have an important effect on the predictive performance for any of the data sets. All i.i.d. and non-i.i.d. data splitting scenarios provide very similar ROC AUC and PR AUC for each data set.

5.1.4 Effect of resampling

The effect of both undersampling and oversampling on the predictive performance is investigated for all supervised learning methods and all data sets. Resampling techniques are parameterized with the ratio of the number of samples in the minority class over the number of samples in the majority class after resampling. It is defined as $\alpha_{rs} = N_{minority}/N_{majority}$ where $N_{minority}$ is the number of samples in the minority class and $N_{majority}$ is the number of samples in the majority class respectively. In all experiments, two values of α_{rs} are considered: $\alpha_{rs} = 20\%/80\% = 0.25$ and $\alpha_{rs} = 10\%/90\% = 0.11$.

Random undersampling

Applying random undersampling to local data for all clients resulted in a lower predictive performance for both Logistic Regression and XGBoost for all data splitting scenarios and all data sets. FFNN on the other hand show slight improvement for some data sets, especially with high number of clients. This is more noticeable for non-i.i.d. scenarios, where the difference in performance provided by undersampling increases with the increasing number of clients. Figure 5.5 shows a comparison between the performance of federated learning without and with undersampling. While the difference in PR AUC is not significant, it keeps increasing with increasing number of clients.

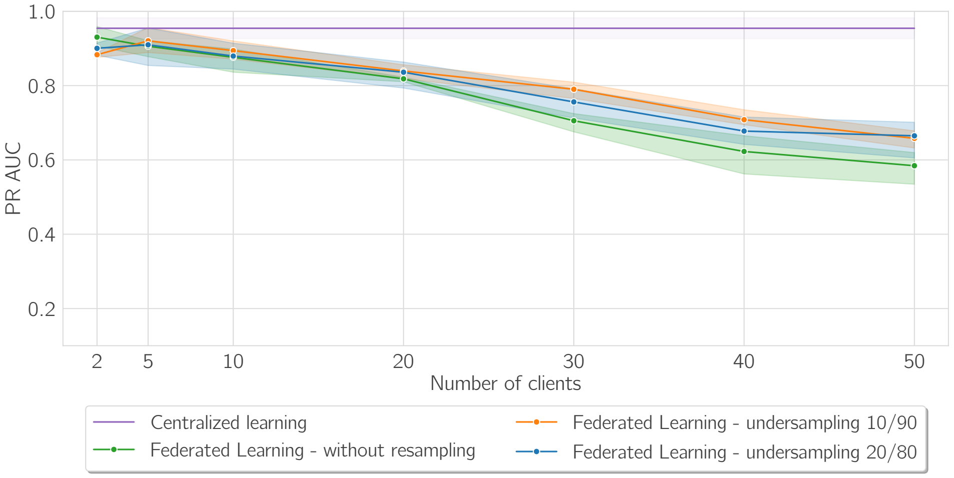


Figure 5.5: PR AUC vs. number of clients for FFNN with the Annthyroid data set - LDP scenario

Random oversampling

Random oversampling applied to local data does not seem to provide any increase in performance for both Logistic Regression and XGBoost. For FFNN, random oversampling provide a slight improvement in performance especially when the amount of data available at the clients is limited. Figure 5.6 shows the performance of federated FFNN for the R2L data set with and without random oversampling. It can be seen that the difference in PR AUC increases with an increasing number of clients, where random oversampling applied with a 20/80 ratio provides the best results.

The fact that with less data oversampling performs slightly better can be explained by the fact that it is able to make up for the scarcity of the anomalous class in the data available at the client. The lower the amount of data available at the client side, the more significant is the effect of oversampling.

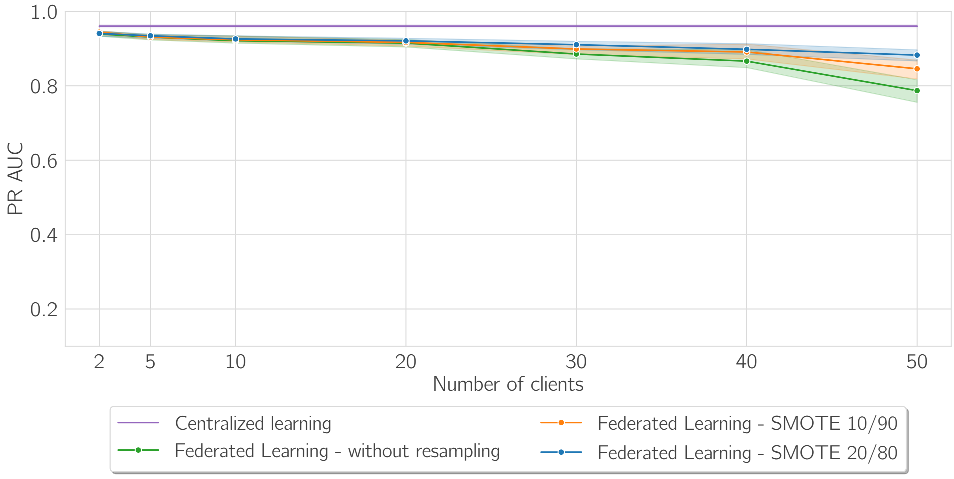


Figure 5.6: PR AUC vs. number of clients for FFNN with the R2L data set - Label-based scenario

SMOTE

For all experiments, applying SMOTE on local data provides little to no increase in performance. Similar to other resampling techniques, this slight increase in performance becomes more noticeable with higher number of clients, as can be seen in Figure 5.7.

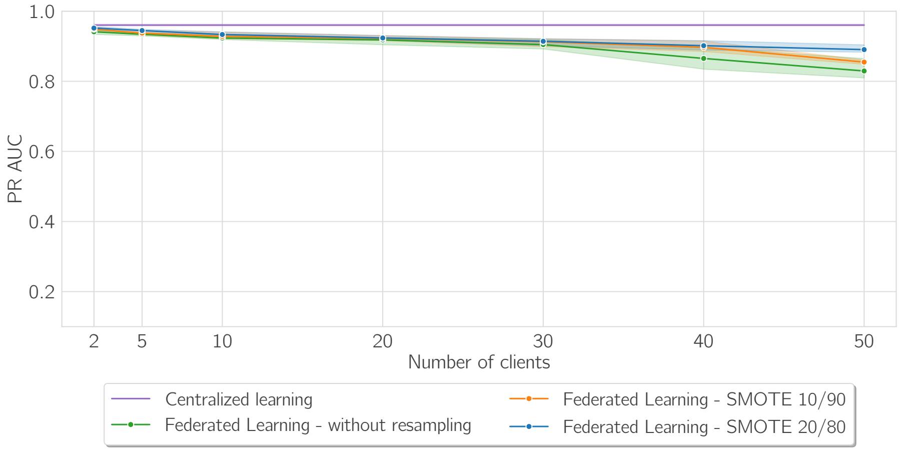


Figure 5.7: PR AUC vs. number of clients for FFNN with the R2L data set - LDP non-i.i.d. scenario

5.2 Weakly-supervised learning

Results for the weakly-supervised learning scenario using DevNet can be seen in Table 5.7. In addition to i.i.d., only the feature-based non-i.i.d. scenario is considered, since very limited amount of labels are available. Each client has at most five labeled anomalies, while some clients are assumed to not to have any anomalous data points.

Table 5.7: DevNet results for all data sets with 10 and 50 clients: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		**Feature-based non-i.i.d.**	
Clients	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC
Credit Card Data Set					
-	Centralized	0.953 (±0.01)	0.683 (±0.07)	-	-
10	Local	0.959 (±0.02)	0.648 (±0.06)	0.957 (±0.01)	0.659 (±0.03)
	Federated	**0.984^{++}(±0.01)**	**0.707 (±0.05)**	**0.981^{+}(±0.01)**	**0.712 (±0.06)**
	Synthesis	0.958 (±0.01)	0.645 (±0.11)	0.915 (±0.08)	0.540 (±0.13)
50	Local	0.955 (±0.01)	0.633 (±0.05)	0.955 (±0.01)	0.635 (±0.06)
	Federated	**0.986^{+}(±0.00)**	**0.711 (±0.06)**	**0.976^{+}(±0.00)**	**0.710 (±0.05)**
	Synthesis	0.907 (±0.07)	0.394 (±0.32)	0.873 (±0.08)	0.185^{-}(±0.18)
Probe Data Set					
-	Centralized	0.933 (±0.09)	0.877 (±0.12)	-	-
10	Local	0.933 (±0.03)	0.859 (±0.04)	0.933 (±0.02)	0.863 (±0.02)
	Federated	**0.994^{+}(±0.00)**	**0.962^{+}(±0.00)**	**0.993^{+}(±0.00)**	**0.960^{+}(±0.00)**
	Synthesis	0.975 (±0.01)	0.923^{++}(±0.01)	0.932 (±0.09)	0.858 (±0.12)
50	Local	0.948 (±0.01)	0.885 (±0.02)	0.923 (±0.01)	0.857 (±0.01)
	Federated	**0.994^{+}(±0.00)**	**0.962^{+}(±0.01)**	**0.994^{+}(±0.00)**	**0.961^{+}(±0.01)**
	Synthesis	0.919 (±0.04)	0.666^{-}(±0.04)	0.970^{+}(±0.01)	0.833 (±0.06)
R2L Data Set					
-	Centralized	0.805 (±0.01)	0.444 (±0.02)	-	-
10	Local	0.756 (±0.05)	0.268 (±0.08)	0.755 (±0.09)	0.284 (±0.16)
	Federated	**0.930^{+}(±0.03)**	**0.354 (±0.10)**	**0.965^{+}(±0.01)**	**0.442 (±0.03)**
	Synthesis	0.723 (±0.08)	0.293 (±0.11)	0.782 (±0.02)	0.221 (±0.01)
50	Local	0.762 (±0.05)	0.270 (±0.13)	0.762 (±0.05)	0.270 (±0.12)
	Federated	**0.926^{+}(±0.00)**	**0.290 (±0.01)**	**0.965 (±0.00)**	**0.422 (±0.00)**
	Synthesis	0.703 (±0.04)	0.169 (±0.09)	0.519^{-}(±0.11)	0.081^{--}(±0.02)
Anthyroid Data Set					
-	Centralized	0.688 (±0.13)	0.220 (±0.19)	-	-
10	Local	**0.744 (±0.01)**	0.300 (±0.03)	0.742 (±0.04)	0.294 (±0.07)
	Federated	0.724 (±0.04)	**0.444^{+}(±0.04)**	**0.759 (±0.02)**	**0.478^{+}(±0.03)**
	Synthesis	0.738 (±0.04)	0.279 (±0.05)	0.722 (±0.06)	0.244 (±0.11)
50	Local	0.723 (±0.02)	0.273 (±0.05)	0.721 (±0.01)	0.268 (±0.03)
	Federated	**0.747 (±0.07)**	**0.428^{+}(±0.02)**	**0.783^{+}(±0.02)**	**0.439^{+}(±0.04)**
	Synthesis	0.683 (±0.10)	0.252 (±0.13)	0.648 (±0.15)	0.185 (±0.11)

Overall, federated learning shows better performance compared to both synthetic data-based and local learning. For the Credit Card data set, results for federated learning are quite comparable to local learning. With i.i.d. data for instance, the PR AUC values range between 0.626 and 0.683 for local training and 0.708 and 0.711 for federated learning with variance values between 0.05 and 0.06.

For the Probe data set, federated learning provides significantly better results for both i.i.d. and non-i.i.d. data splitting scenarios. With i.i.d. data, federated learning provides an ROC AUC and PR AUC of 0.994 and 0.962 compared to 0.948 and 0.885 for local learning.

Results for R2L data set, show that the PR AUC values for federated learning are not significantly better than local learning. In addition, local learning results show high variance (up to 0.21) for both i.i.d. and non-i.i.d. data splitting scenarios.

For the Annthyroid data set, the superior performance of federated learning for both i.i.d. and non-i.i.d. data is more noticeable in terms of PR AUC with values ranging between 0.439 and 0.478 across the different number of clients.

The synthetic data-based learning scenario, however, shows lower performance compared to federated and local learning, and a very high variance for all data sets. For example, in the case of 50 clients, it shows an PR AUC of 0.394 with a variance of 0.32.

DevNet performs well with very limited amount of data, and therefore the decrease in performance due to increasing number of clients or due to different data distribution is very limited. For both the Credit Card and Probe data sets, very similar performance can be seen in the experiments with different number of clients. This characteristic of DevNet models is also reflected in the global federated model convergence. It only takes a few communication rounds (usually below 10) to achieve good predictive performance.

5.3 Semi-supervised learning

Results for semi-supervised learning using Autoencoder can be seen in Table 5.8.

For the Credit Card data set, results for the different learning scenarios are comparable, where federated and synthetic data-based learning do not provide significant improvement over local learning for both i.i.d. and non-i.i.d. data splits.

For the Probe data set, federated learning shows slightly better performance compared to local learning for low number of clients. This difference becomes more significant with higher number of clients where federated learning provides a PR AUC of 0.822 compared to 0.729 given by local training.

For both R2L and Annthyroid data sets, federated learning shows significantly better results compared to both local and synthetic data-based learning. In particular, with more than ten clients, federated learning provides more than 50% increase in PR AUC compared to local learning.

The effect of data distribution on the results is not clearly visible in all experiments. Both data splitting scenarios show similar ROC AUC and PR AUC – with the exception of Annthyroid data set, where non-i.i.d. data show lower performance compared to i.i.d. data.

Figure 5.8 shows the evolution of PR AUC during the federated learning process. It can be observed that the higher the number of clients, the slower the model converges. Models, however, converge much faster than those from e.g. FFNN (cf. Figure 5.2). In addition, different number of clients results in very similar final PR AUC values.

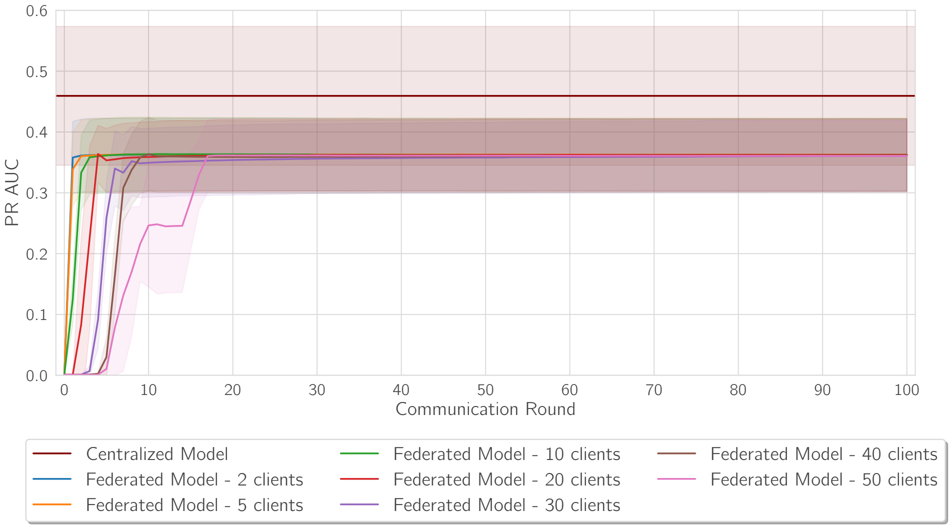


Figure 5.8: PR AUC vs. number of clients for Autoencoder with the credit card data set - i.i.d. scenario

Table 5.8: Autoencoder results for all data sets with 10 and 50 clients: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		**Feature-based non-i.i.d.**	
Clients	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC
Credit Card Data Set					
-	Centralized	0.938 (±0.01)	0.459 (±0.11)	-	-
10	Local	0.941 (±0.02)	0.361 (±0.06)	0.939 (±0.02)	0.358 (±0.06)
	Federated	**0.941 (±0.02)**	**0.362 (±0.06)**	0.941 (±0.02)	**0.364 (±0.06)**
	Synthesis	0.933 (±0.01)	0.068^{-}(±0.02)	**0.943 (±0.01)**	0.098^{-}(±0.05)
50	Local	0.941 (±0.02)	0.360 (±0.06)	0.938 (±0.02)	0.357 (±0.06)
	Federated	0.941 (±0.02)	0.361 (±0.06)	0.940 (±0.02)	0.359 (±0.06)
	Synthesis	**0.950 (±0.02)**	**0.399 (±0.06)**	**0.951 (±0.02)**	**0.367 (±0.05)**
Probe Data Set					
-	Centralized	0.961 (±0.00)	0.826 (±0.01)	-	-
10	Local	**0.974 (±0.00)**	**0.850 (±0.01)**	**0.970 (±0.00)**	0.833 (±0.00)
	Federated	0.970 (±0.00)	0.841 (±0.02)	0.967 (±0.01)	**0.834 (±0.05)**
	Synthesis	0.832^{-}(±0.03)	0.395^{-}(±0.03)	0.901^{-}(±0.01)	0.502^{-}(±0.02)
50	Local	0.955 (±0.00)	0.729 (±0.01)	0.924 (±0.01)	0.651 (±0.03)
	Federated	**0.969^{+}(±0.01)**	**0.822^{+}(±0.01)**	**0.964^{+}(±0.02)**	**0.821^{+}(±0.05)**
	Synthesis	0.827^{-}(±0.02)	0.432^{-}(±0.02)	0.891 (±0.03)	0.520^{-}(±0.04)
R2L Data Set					
-	Centralized	0.928 (±0.01)	0.454 (±0.03)	-	-
10	Local	**0.896 (±0.00)**	0.186 (±0.01)	0.822 (±0.00)	0.129 (±0.01)
	Federated	0.874^{-}(±0.01)	**0.231^{+}(±0.00)**	**0.920^{+}(±0.01)**	**0.234^{+}(±0.01)**
	Synthesis	0.785^{-}(±0.03)	0.112^{-}(±0.03)	0.769^{--}(±0.03)	0.097^{-}(±0.01)
50	Local	0.777 (±0.01)	0.098 (±0.01)	0.764 (±0.01)	0.094 (±0.01)
	Federated	**0.897^{+}(±0.01)**	**0.184^{+}(±0.02)**	**0.889^{+}(±0.01)**	**0.313^{+}(±0.06)**
	Synthesis	0.802^{+}(±0.01)	0.117^{+}(±0.01)	0.795 (±0.09)	0.121 (±0.02)
Annthyroid Data Set					
-	Centralized	0.856 (±0.02)	0.295 (±0.08)	-	-
10	Local	0.613 (±0.02)	0.103 (±0.00)	0.613 (±0.02)	0.103 (±0.00)
	Federated	0.717^{+}(±0.02)	**0.181^{+}(±0.02)**	**0.658^{++}(±0.02)**	**0.139^{+}(±0.01)**
	Synthesis	**0.774^{+}(±0.01)**	0.164^{+}(±0.03)	0.642 (±0.12)	0.126 (±0.02)
50	Local	0.534 (±0.01)	0.082 (±0.00)	0.534 (±0.01)	0.082 (±0.00)
	Federated	**0.687^{+}(±0.02)**	**0.162^{+}(±0.02)**	**0.681^{+}(±0.03)**	**0.143^{+}(±0.02)**
	Synthesis	0.672^{+}(±0.08)	0.142^{++}(±0.05)	0.612 (±0.09)	0.104^{++}(±0.02)

5.4 Unsupervised learning

In this section, results for unsupervised learning using Isolation Forest and REPEN are presented. With no labels available, the anomaly detection task becomes more challenging. Overall, results are inconclusive and there is no learning scenario that shows significantly better performance compared to the others.

5.4.1 Isolation Forest

The results for isolation forest can be seen in Table 5.9 and Table 5.10. Overall, the different learning scenarios are comparable and privacy-preserving approaches do not provide any improvement over local learning.

Table 5.9: Isolation Forest results for all data sets with 10 and 50 clients using i.i.d. and Feature-based non-i.i.d. data: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		**Feature-based non-i.i.d.**	
Clients	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC
		Credit Card Data Set			
-	Centralized	0.947 (±0.01)	0.213 (±0.04)	-	-
10	Local	0.947 (±0.01)	0.215 (±0.05)	0.947 (±0.01)	0.215 (±0.04)
	Federated	0.948 (±0.01)	0.214 (±0.05)	**0.948 (±0.01)**	0.214 (±0.05)
	Synthesis	**0.949 (±0.01)**	**0.219 (±0.04)**	0.947 (±0.01)	**0.223 (±0.05)**
50	Local	0.947 (±0.01)	0.216 (±0.05)	0.946 (±0.01)	0.219 (±0.05)
	Federated	0.947 (±0.01)	0.216 (±0.05)	0.947 (±0.01)	0.220 (±0.05)
	Synthesis	**0.949 (±0.01)**	**0.255 (±0.06)**	**0.948 (±0.01)**	**0.245 (±0.04)**
		Probe Data Set			
-	Centralized	0.885 (±0.02)	0.443 (±0.05)	-	-
10	Local	0.888 (±0.01)	0.461 (±0.02)	0.885 (±0.01)	0.463 (±0.03)
	Federated	0.891 (±0.01)	0.463 (±0.03)	0.892 (±0.02)	0.467 (±0.04)
	Synthesis	**0.892 (±0.01)**	**0.472 (±0.03)**	**0.898 (±0.01)**	**0.478 (±0.02)**
50	Local	0.896 (±0.01)	0.504 (±0.03)	0.891 (±0.01)	0.502 (±0.03)
	Federated	0.899 (±0.01)	**0.507 (±0.03)**	0.901 (±0.01)	**0.512 (±0.03)**
	Synthesis	**0.900 (±0.01)**	0.490 (±0.02)	**0.903 (±0.01)**	0.499 (±0.02)
		R2L Data Set			
-	Centralized	0.818 (±0.00)	0.112 (±0.00)	-	-
10	Local	0.818 (±0.01)	0.112 (±0.01)	0.813 (±0.01)	0.110 (±0.01)
	Federated	0.819 (±0.01)	0.112 (±0.01)	0.818 (±0.01)	0.111 (±0.01)
	Synthesis	**0.831 (±0.01)**	**0.121 (±0.01)**	**0.832**$^{+}$(±0.00)	**0.120**$^{++}$(±0.00)
50	Local	0.812 (±0.01)	0.108 (±0.01)	0.806 (±0.01)	0.106 (±0.01)
	Federated	0.815 (±0.01)	0.109 (±0.01)	0.815 (±0.01)	0.109 (±0.01)
	Synthesis	**0.838**$^{+}$(±0.01)	**0.124**$^{+}$(±0.00)	**0.840**$^{+}$(±0.00)	**0.125**$^{+}$(±0.01)
		Annthyroid Data Set			
-	Centralized	0.789 (±0.02)	0.291 (±0.02)	-	-
10	Local	0.783 (±0.00)	0.267 (±0.03)	0.782 (±0.01)	0.270 (±0.03)
	Federated	**0.790**$^{+}$(±0.00)	0.271 (±0.03)	**0.789 (±0.01)**	0.272 (±0.04)
	Synthesis	0.787 (±0.02)	**0.284 (±0.04)**	0.786 (±0.01)	**0.281 (±0.04)**
50	Local	0.778 (±0.01)	0.263 (±0.03)	0.777 (±0.01)	0.264 (±0.03)
	Federated	0.786 (±0.01)	0.273 (±0.03)	0.787 (±0.01)	0.276 (±0.03)
	Synthesis	**0.792 (±0.01)**	**0.290 (±0.03)**	**0.793**$^{++}$(±0.01)	**0.293 (±0.03)**

For all data sets, there is no significant difference between the ROC AUC and PR AUC for the various learning scenarios, with data synthesis providing slightly better performance in most of the cases. In addition, it can be observed that even centralized training does not provide any advantage over local learning. This can be explained by the fact that isolation forest performs well with small amount of data, and therefore still provides a good performance on average across all local models. The data distribution also does not seem to have an effect on the results, with i.i.d. and non-i.i.d. scenarios performing similarly.

As mentioned in Section 4.8, model aggregation is achieved by training local models and combining the trees from each to create a global model. According to the obtained results, this does not seem to provide any benefit compared to local learning.

Table 5.10: Isolation Forest results for all data sets with 10 and 50 clients - Label-based and LDP non-i.i.d. data: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		Label-based non-i.i.d.		**LDP non-i.i.d.**	
Clients	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC
		Credit Card Data Set			
-	Centralized	0.947 (±0.01)	0.213 (±0.04)	-	-
10	Local	0.928 (±0.02)	0.179 (±0.03)	0.930 (±0.01)	0.189 (±0.04)
	Federated	0.942 (±0.01)	0.190 (±0.03)	0.940 (±0.01)	0.200 (±0.04)
	Synthesis	**0.949 (±0.01)**	**0.221 (±0.03)**	**0.949 (±0.01)**	**0.214 (±0.04)**
50	Local	0.929 (±0.01)	0.155 (±0.04)	0.930 (±0.01)	0.170 (±0.04)
	Federated	0.941 (±0.01)	0.170 (±0.06)	0.941 (±0.01)	0.190 (±0.06)
	Synthesis	**0.948^{++}(±0.01)**	**0.239^{++}(±0.04)**	**0.946 (±0.01)**	**0.241^{++}(±0.04)**
		Probe Data Set			
-	Centralized	0.885 (±0.02)	0.443 (±0.05)	-	-
10	Local	0.833 (±0.01)	0.455 (±0.04)	0.822 (±0.02)	0.413 (±0.06)
	Federated	0.878^{+}(±0.02)	**0.495 (±0.07)**	0.852 (±0.02)	0.437 (±0.07)
	Synthesis	**0.898^{+}(±0.02)**	0.485 (±0.04)	**0.899^{+}(±0.01)**	**0.483 (±0.02)**
50	Local	0.838 (±0.02)	0.463 (±0.07)	0.836 (±0.03)	0.445 (±0.09)
	Federated	0.873^{++}(±0.01)	0.490 (±0.09)	0.857 (±0.01)	0.467 (±0.09)
	Synthesis	**0.904^{+}(±0.01)**	**0.501 (±0.01)**	**0.904^{+}(±0.01)**	**0.503 (±0.02)**
		R2L Data Set			
-	Centralized	0.818 (±0.00)	0.112 (±0.00)	-	-
10	Local	0.785 (±0.02)	0.109 (±0.01)	0.774 (±0.01)	0.099 (±0.01)
	Federated	0.807 (±0.02)	0.108 (±0.01)	0.792 (±0.01)	0.101 (±0.01)
	Synthesis	**0.832 (±0.00)**	**0.118 (±0.00)**	**0.834^{+}(±0.00)**	**0.121^{+}(±0.00)**
50	Local	0.780 (±0.00)	0.107 (±0.01)	0.783 (±0.01)	0.106 (±0.00)
	Federated	0.799^{+}(±0.01)	0.104 (±0.00)	0.799^{++}(±0.01)	0.105 (±0.01)
	Synthesis	**0.837 (±0.00)**	**0.121 (±0.00)**	**0.845^{+}(±0.00)**	**0.129^{+}(±0.00)**
		Annthyroid Data Set			
-	Centralized	0.789 (±0.02)	0.291 (±0.02)	-	-
10	Local	0.682 (±0.14)	0.198 (±0.09)	0.681 (±0.14)	0.192 (±0.10)
	Federated	0.722 (±0.17)	0.219 (±0.11)	0.714 (±0.19)	0.215 (±0.12)
	Synthesis	**0.791 (±0.03)**	**0.283 (±0.03)**	**0.796 (±0.01)**	**0.279 (±0.03)**
50	Local	0.693 (±0.16)	0.196 (±0.10)	0.710 (±0.16)	0.212 (±0.11)
	Federated	0.731 (±0.20)	0.231 (±0.13)	0.742 (±0.20)	0.245 (±0.14)
	Synthesis	**0.792 (±0.01)**	**0.285 (±0.03)**	**0.799 (±0.01)**	**0.296 (±0.03)**

5.4.2 REPEN

Similar to isolation forest, the results for REPEN show that on average privacy-preserving learning does not provide any significant improvement over local learning. Results for all data splitting scenarios can be seen in Tables 5.11 and 5.12.

Table 5.11: REPEN results for all data sets with 10 and 50 clients using i.i.d. and Feature-based non-i.i.d. data: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		**Feature-based non-i.i.d.**	
Clients	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC
		Credit Card Data Set			
-	Centralized	0.946 (±0.02)	0.511 (±0.04)	-	-
10	Local	0.929 (±0.02)	0.508 (±0.08)	0.932 (±0.01)	0.485 (±0.01)
	Federated	0.939 (±0.02)	0.659^{++}(±0.08)	0.929 (±0.02)	**0.631^{+}(±0.02)**
	Synthesis	**0.945 (±0.01)**	**0.680^{+}(±0.06)**	**0.933 (±0.01)**	0.451 (±0.35)
50	Local	0.932 (±0.02)	0.569 (±0.07)	**0.931 (±0.02)**	0.588 (±0.06)
	Federated	0.917 (±0.02)	**0.675 (±0.06)**	0.915 (±0.02)	**0.674 (±0.06)**
	Synthesis	**0.943 (±0.01)**	0.414 (±0.14)	0.924 (±0.04)	0.377 (±0.31)
		Probe Data Set			
-	Centralized	0.945 (±0.02)	0.731 (±0.05)	-	-
10	Local	**0.918 (±0.01)**	**0.559 (±0.01)**	0.901 (±0.01)	0.552 (±0.02)
	Federated	0.899^{--}(±0.01)	0.521 (±0.03)	0.839 (±0.09)	0.455 (±0.14)
	Synthesis	0.825 (±0.10)	0.405 (±0.17)	**0.948^{+}(±0.01)**	**0.667^{+}(±0.06)**
50	Local	0.920 (±0.00)	0.613 (±0.01)	0.911 (±0.00)	0.609 (±0.01)
	Federated	**0.922 (±0.02)**	**0.644^{++}(±0.02)**	**0.918 (±0.02)**	**0.632 (±0.07)**
	Synthesis	0.860 (±0.07)	0.462 (±0.19)	0.904 (±0.05)	0.570 (±0.17)
		R2L Data Set			
-	Centralized	0.807 (±0.01)	0.107 (±0.01)	-	-
10	Local	0.779 (±0.00)	**0.126 (±0.01)**	0.796 (±0.01)	**0.132 (±0.01)**
	Federated	0.724 (±0.11)	0.098 (±0.03)	**0.819 (±0.04)**	0.130 (±0.02)
	Synthesis	**0.809 (±0.04)**	0.119 (±0.02)	0.747 (±0.10)	0.091^{--}(±0.03)
50	Local	0.775 (±0.01)	0.120 (±0.01)	0.770 (±0.02)	0.121 (±0.00)
	Federated	0.728 (±0.05)	0.107 (±0.02)	0.731 (±0.10)	0.099 (±0.03)
	Synthesis	**0.799 (±0.09)**	**0.197 (±0.18)**	0.828^{++}(±0.04)	**0.148 (±0.06)**
		Annthyroid Data Set			
-	Centralized	0.638 (±0.02)	0.168 (±0.02)	-	-
10	Local	0.581 (±0.02)	0.142 (±0.01)	0.567 (±0.00)	0.137 (±0.01)
	Federated	**0.664^{+}(±0.03)**	**0.219 (±0.08)**	**0.670 (±0.08)**	**0.236 (±0.14)**
	Synthesis	0.609 (±0.04)	0.153 (±0.06)	0.593^{++}(±0.02)	0.181^{++}(±0.03)
50	Local	0.583 (±0.01)	0.151 (±0.01)	0.577 (±0.02)	0.149 (±0.01)
	Federated	**0.651^{+}(±0.01)**	**0.166 (±0.02)**	**0.624^{+}(±0.01)**	0.135 (±0.01)
	Synthesis	0.625 (±0.04)	0.150 (±0.04)	0.571 (±0.07)	**0.152 (±0.04)**

For the Credit Card data set, synthetic data-based learning provides the highest performance for label-based and LDP non-i.i.d. data splitting scenarios. This becomes more obvious as the number of clients increases and the PR AUC value becoming significantly higher than the one obtained from local training. With 50 clients, the ROC AUC and PR AUC values for synthetic data-based learning are 0.944 and 0.680 respectively, compared to 0.934 and 0.679 achieved by local learning.

For the Probe data set, synthetic data-based learning provides better performance for low number of clients with data split across the client in non-i.i.d. manner. On the other hand, federated learning shows slightly better ROC AUC and PR AUC for more clients with i.i.d. data.

For the Annthyroid data set, federated learning provides significantly better ROC AUC values for the i.i.d. data splitting scenario. The same behavior can be observed when using Label-based Dirichlet Partition non-i.i.d. data.

Data distribution does not seem to have an effect on the results of REPEN, as non-i.i.d. scenarios perform equally well to their i.i.d. counterpart. This can be explained by the ability of REPEN to perform well with very limited amount of labeled data [PCCL18].

When trained in federated manner, REPEN show fast convergence where after few communication rounds, the global model already reaches maximum predictive performance.

Table 5.12: REPEN results for all data sets with 10 and 50 clients using Feature-based and LDP non-i.i.d. data: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		Label-based non-i.i.d.		LDP non-i.i.d.	
Clients	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC
Credit Card Data Set					
-	Centralized	0.946 (±0.02)	0.511 (±0.04)	-	-
10	Local	0.923 (±0.02)	0.421 (±0.12)	0.932 (±0.01)	0.481 (±0.02)
	Federated	0.886 (±0.05)	0.459 (±0.08)	0.920 (±0.03)	0.616^{+}(±0.02)
	Synthesis	**0.936 (±0.02)**	**0.554 (±0.23)**	**0.936 (±0.03)**	**0.669^{+}(±0.06)**
50	Local	0.926 (±0.02)	0.478 (±0.02)	0.932 (±0.01)	0.574 (±0.02)
	Federated	0.875^{--} (±0.02)	0.599^{+}(±0.00)	0.901 (±0.03)	0.663^{++}(±0.05)
	Synthesis	**0.944 (±0.03)**	**0.680^{+}(±0.07)**	**0.934 (±0.04)**	**0.679^{+}(±0.06)**
Probe Data Set					
-	Centralized	0.945 (±0.02)	0.731 (±0.05)	-	-
10	Local	**0.872 (±0.04)**	0.559 (±0.03)	0.886 (±0.02)	0.520 (±0.04)
	Federated	0.861 (±0.06)	0.552 (±0.07)	0.743^{-}(±0.04)	0.329^{-}(±0.09)
	Synthesis	0.865 (±0.15)	**0.578 (±0.29)**	**0.921 (±0.04)**	**0.637 (±0.11)**
50	Local	0.878 (±0.00)	0.576 (±0.02)	**0.903 (±0.01)**	**0.608 (±0.02)**
	Federated	0.861 (±0.03)	0.585 (±0.05)	0.898 (±0.03)	0.549 (±0.08)
	Synthesis	**0.936^{+}(±0.03)**	**0.648 (±0.15)**	0.877 (±0.08)	0.540 (±0.18)
R2L Data Set					
-	Centralized	0.807 (±0.01)	0.107 (±0.01)	-	-
10	Local	**0.802 (±0.02)**	**0.145 (±0.01)**	**0.794 (±0.00)**	**0.124 (±0.01)**
	Federated	0.681 (±0.15)	0.100 (±0.05)	0.599 (±0.21)	0.082 (±0.04)
	Synthesis	0.773 (±0.09)	0.103 (±0.03)	0.789 (±0.01)	0.103^{-}(±0.01)
50	Local	**0.777 (±0.01)**	**0.126 (±0.00)**	0.773 (±0.01)	0.121 (±0.00)
	Federated	0.704^{-}(±0.02)	0.104 (±0.02)	0.758 (±0.08)	**0.133 (±0.05)**
	Synthesis	0.730 (±0.12)	0.092 (±0.03)	**0.804 (±0.03)**	0.117 (±0.02)
Annthyroid Data Set					
-	Centralized	0.638 (±0.02)	0.168 (±0.02)	-	-
10	Local	0.596 (±0.02)	**0.148 (±0.01)**	0.602 (±0.03)	0.166 (±0.03)
	Federated	**0.623 (±0.03)**	0.146 (±0.05)	**0.693^{++}(±0.06)**	**0.248 (±0.12)**
	Synthesis	0.611 (±0.06)	0.133 (±0.06)	0.635 (±0.02)	0.185 (±0.03)
50	Local	0.577 (±0.02)	0.146 (±0.02)	0.586 (±0.01)	0.154 (±0.01)
	Federated	0.562 (±0.13)	0.106 (±0.05)	0.615 (±0.06)	0.128 (±0.04)
	Synthesis	**0.643^{++}(±0.04)**	**0.183 (±0.03)**	**0.624 (±0.09)**	**0.200 (±0.11)**

CHAPTER 6

Conclusion

This thesis provided a comprehensive evaluation of privacy-preserving machine learning methods for anomaly detection in distributed systems. It addresses the predictive performance of two privacy-preserving solutions in a multi-nodal system where multiple clients are connected to a centralized server. The first approach involved training a centralized model using aggregated data obtained from generating synthetic instances at the local clients. The second method is federated learning, where a global model is collaboratively trained across the different clients. Both settings ensure that the original data are not shared and always remain at the client side. We compared both approaches to the average performance of models trained on the client's local data, and to the performance of a centralized model trained on data aggregated from all clients.

As part of the evaluation, multiple state-of-the-art anomaly detection algorithms were implemented and tested on different benchmark anomaly detection data sets. The selected algorithms cover different machine learning approaches, suited for different availability of labels in the training data.

The selected data sets cover several critical anomaly detection applications, where privacy is a serious concern. To ensure that the distributed learning experiments reflect real-world scenarios, besides an i.i.d. data splitting, three different non-i.i.d. scenarios take into consideration different ways in which data may deviate from being i.i.d..

The evaluation results showed that federated learning clearly outperforms local data-based learning for all supervised and weakly-supervised learning methods. Even though synthetic-based learning provides good performance in some cases, its behavior appears to be inconsistent and highly depends on the used data. For federated learning, applying resampling techniques for supervised learning did not provide significant improvement in the predictive performance, even when only a small amount of data is available at each client.

For semi-supervised learning, we showed that federated learning provides the best performance for most data sets – especially when the amount of data available at each client is limited. In addition, the decrease in performance due to distributed learning is more significant.

For unsupervised learning, we observed that privacy-preserving learning do not provide any advantage over local data-based learning. Even though synthetic data-based learning shows slightly better performance for some settings, its behavior is not consistent across the different experiments.

6.1 Research questions revisited

In this section, we revisit the research questions defined in Section 1.3 and discuss the research outcomes related to each of them.

RQ 1) What is the effect of collaborative learning when applied to anomaly detection models compared to centralized learning?

Overall, the predictive performance provided by collaborative learning is comparable to centralized learning. The difference in ROC AUC and PR AUC provided by federated and synthetic data-based learning when compared to centralized training is very limited. However, this difference gets larger with increasing number of clients. In addition, we observed that semi-supervised learning is the only training approach that shows large reduction in performance between the distributed learning scenarios and centralized training.

RQ 1.1) To what extent does training such models in a federated manner affect the overall predictive performance?

In most cases, federated learning shows slightly lower ROC AUC and PR AUC compared to centralized learning. Such decrease in performance increases with an increasing number of clients. In particular, we observed that weakly-supervised federated learning using DevNet provides a higher performance compared to centralized learning. This method is designed for applications involving a small amount of data, and is therefore able to perform well in a federated setting. On the other hand, semi-supervised learning using Autoencoder shows the highest decrease in performance for federated learning.

RQ 1.2) Which models are more suited to be used in a federated architecture for more effective anomaly detection?

Both supervised and weakly-supervised learning methods provided a good predictive performance when trained in a federated manner. In fact, the availability of labels had a large impact on the anomaly detection task. In supervised learning, where labels are available for all data points, Feedforward Neural Network (FFNN) and XGBoost provided high ROC AUC and PR AUC in almost all experiments compared to local data-based learning. The weakly-supervised learning setting using DevNet also showed high predictive performance for federated learning – in some experiments even higher than centralized learning.

Auotencoder trained on normal instances only showed significantly better performance for federated learning when compared to local data-based learning for three out of the four used data sets. The significance of the performance of federated learning becomes more apparent with an increasing number of clients. However, its behavior seems to be dependent of the training data types.

On the other hand, we showed that federated learning using unsupervised learning algorithms does not provide any advantage over training local models at each individual client.

It can be therefore concluded that, depending on the data labels availability, supervised, weakly-supervised, and semi-supervised anomaly detection algorithms are well suited to be used in a federated architecture.

RQ 2) What impact does data heterogeneity have on the anomaly detection models in federated learning?

In general, the effect of data distribution on the predictive performs vary across the different experiments. It was observed that the effect of data being non-i.i.d. highly depends on the used algorithm and the type of data. The effect of the amount of data available at each client, which is inversely proportional to the number of clients, is on the other hand more prominent.

RQ 2.1) To what degree do the amount and distribution of data available at local nodes influence the global model performance?

Overall, the difference in performance between the data being split in i.i.d. and non-i.i.d. manners is not significant. The way the different models are affected highly depends on the training data. For FFNN, while the difference in performance between i.i.d. and non-i.i.d. data splitting scenarios cannot be seen for the Credit Card data set, it was more visible for the other data sets. For instance, with the Annthyroid data set, a high decrease in predictive performance for FFNN can be seen between i.i.d. and non-i.i.d. data distribution scenarios. This is more prominent in the case of label-based data splitting especially with increasing number of clients where the PR AUC value decreased by almost half when using 50 clients. This behavior cannot be seen for other methods such as DevNet where different data distribution scenarios resulted in very similar performance.

We also observed that the predictive performance of supervised and semi-supervised models when trained in federated manner might depend on the amount of data available at the individual clients. From the results, it was clear that higher number of clients decreases both ROC AUC and PR AUC of the global federated model. For instance, federated FFNN trained on i.i.d. R2L data shows more than 30% decrease in PR AUC when going from 2 to 50 clients.

This effect is not visible in weakly-supervised learning using DevNet, since this model is known to perform well with very limited amount of data. The same applies to unsupervised methods where increasing the number of clients does not necessarily result in decreasing of the federated model performance, albeit on generally rather low results.

RQ 2.2) To what extent does applying resampling techniques to local data at individual nodes affect the global model predictive performance?

Applying resmapling techniques in the supervised learning setting did not result in a clear increase in predictive performance of the global federated model. The methods evaluated included random undersampling, random oversampling and SMOTE. All of them have resulted in almost the same predictive performance of the federated model when applied to local data at the client side. In particular, random undersampling provided slightly lower performance for some of the non-i.i.d. data splitting scenario.

RQ 3) How do models trained in federated manner perform when compared to central models trained using synthetic data locally generated at client nodes?

Unlike federated learning, the performance of models trained using synthetic data is not consistent across the different algorithms and data sets. For instance, while the synthetic data-based models perform well with the Credit Card data set, they show very low performance for the R2L data set. On the other hand, federated learning consistently showed good predictive performance and does not seem to be highly dependent on the type of training data.

6.2 Future work

It was observed that the obtained results in this thesis do not allow drawing clear conclusions for the different unsupervised learning scenarios. Investigating other unsupervised algorithms could lead to better understanding of both federated and synthetic-data based learning.

The tested non-i.i.d. data splitting scenarios cover most of the aspects in which data might deviate from being i.i.d.. In order to evaluate the similarity between the generated subsets, Earth Mover's Distance (EMD) was used. The obtained EMD values clearly show that the defined non-i.i.d. scenarios introduced significant statistical heterogeneity among the simulated clients. However, as opposed to what was expected, the effect of data distribution was not clearly visible for most experiments. Even though it was observed that federated learning is able to handle well non-i.i.d. data, it would be interesting to evaluate the effect of data distribution in depth. The defined data splitting scenarios can be easily parameterized to control the degree to which the data get close and deviate from being non-i.i.d.. They can therefore be used as a starting point to define more extreme cases of data distributed in a non-i.i.d. manner.

The obtained results show different behavior for the different data sets. In particular, synthetic data-based learning results for the Credit Card data set are usually different than the other data sets. It is believed that this behavior is related to the nature of the data since PCA has been applied on the feature for anonymization purposes. In order to investigate this further, experiments can be repeated after applying PCA to the other three data sets. In addition, to obtain a better understanding on the effect of the nature of training data on the learning process, the experiments conducted in this thesis may be conducted again with new data sets.

APPENDIX A

Results

This section provides results for all experiments. For each number of clients, algorithm, data set, training setting and data splitting scenario the metrics ROC and PR score are provided. The column with the name N indicates the number of clients that are used to simulate the data splitting scenarios.

Table A.1: Results for FFNN using Credit Card data set: Significance of federated learning and synthetic-based learning results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		Feature-based non-i.i.d.		Label-based non-i.i.d.		LDP non-i.i.d.	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.927 (±0.02)	0.544 (±0.06)	-	-	-	-	-	-
2	Local	0.569 (±0.01)	0.227 (±0.05)	0.568 (±0.01)	0.231 (±0.04)	0.568 (±0.01)	0.230 (±0.05)	0.566 (±0.01)	0.235 (±0.04)
	Federated	**0.956^{+}(±0.01)**	**0.686^{+}(±0.07)**	**0.956^{+}(±0.01)**	**0.686^{+}(±0.07)**	**0.961^{+}(±0.01)**	**0.691^{+}(±0.07)**	**0.966^{+}(±0.00)**	**0.694^{+}(±0.06)**
	Synthesis	0.929^{+}(±0.02)	0.570^{+}(±0.06)	0.930^{+}(±0.02)	0.572^{+}(±0.05)	0.931^{+}(±0.02)	0.582^{+}(±0.05)	0.930^{+}(±0.02)	0.573^{+}(±0.05)
5	Local	0.575 (±0.02)	0.151 (±0.03)	0.572 (±0.02)	0.150 (±0.03)	0.594 (±0.01)	0.187 (±0.01)	0.591 (±0.02)	0.183 (±0.01)
	Federated	0.922^{+}(±0.03)	**0.655^{+}(±0.06)**	0.922^{+}(±0.03)	**0.655^{+}(±0.06)**	**0.946^{+}(±0.02)**	**0.670^{+}(±0.06)**	**0.948^{+}(±0.03)**	**0.681^{+}(±0.08)**
	Synthesis	**0.930^{+}(±0.02)**	0.610^{+}(±0.05)	**0.932^{+}(±0.02)**	0.613^{+}(±0.06)	0.933^{+}(±0.02)	0.623^{+}(±0.05)	0.932^{+}(±0.02)	0.613^{+}(±0.05)
10	Local	0.694 (±0.00)	0.165 (±0.03)	0.694 (±0.01)	0.146 (±0.03)	0.672 (±0.01)	0.120 (±0.04)	0.686 (±0.01)	0.132 (±0.03)
	Federated	0.895^{+}(±0.02)	0.528^{+}(±0.05)	0.894^{+}(±0.02)	0.528^{+}(±0.05)	0.905^{+}(±0.02)	0.566^{+}(±0.05)	0.911^{+}(±0.02)	0.592^{+}(±0.06)
	Synthesis	**0.930^{+}(±0.02)**	**0.638^{+}(±0.05)**	**0.932^{+}(±0.02)**	**0.639^{+}(±0.06)**	**0.933^{+}(±0.02)**	**0.648^{+}(±0.05)**	**0.932^{+}(±0.02)**	**0.641^{+}(±0.05)**
20	Local	0.640 (±0.01)	0.151 (±0.02)	0.648 (±0.03)	0.150 (±0.02)	0.637 (±0.04)	0.155 (±0.03)	0.640 (±0.04)	0.154 (±0.03)
	Federated	**0.986^{+}(±0.01)**	**0.716^{+}(±0.06)**	**0.945^{+}(±0.01)**	0.651^{+}(±0.07)	**0.965^{+}(±0.01)**	**0.693^{+}(±0.07)**	**0.965^{+}(±0.01)**	**0.691^{+}(±0.07)**
	Synthesis	0.932^{+}(±0.02)	0.651^{+}(±0.06)	0.933^{+}(±0.02)	**0.653^{+}(±0.06)**	0.934^{+}(±0.02)	0.663^{+}(±0.06)	0.934^{+}(±0.02)	0.654^{+}(±0.05)
30	Local	0.557 (±0.00)	0.100 (±0.01)	0.558 (±0.00)	0.100 (±0.01)	0.551 (±0.01)	0.098 (±0.01)	0.551 (±0.01)	0.099 (±0.01)
	Federated	0.902^{+}(±0.03)	0.610^{+}(±0.05)	0.902^{+}(±0.03)	0.610^{+}(±0.05)	0.909^{+}(±0.03)	0.624^{+}(±0.05)	0.908^{+}(±0.03)	0.622^{+}(±0.05)
	Synthesis	**0.935^{+}(±0.02)**	**0.661^{+}(±0.05)**	**0.934^{+}(±0.02)**	**0.663^{+}(±0.06)**	**0.936^{+}(±0.02)**	**0.671^{+}(±0.06)**	**0.936^{+}(±0.02)**	**0.662^{+}(±0.06)**
40	Local	0.631 (±0.01)	0.110 (±0.02)	0.630 (±0.01)	0.111 (±0.02)	0.630 (±0.00)	0.113 (±0.02)	0.632 (±0.01)	0.114 (±0.02)
	Federated	**0.982^{+}(±0.01)**	**0.719^{+}(±0.05)**	**0.942^{+}(±0.01)**	0.585^{+}(±0.04)	**0.954^{+}(±0.01)**	0.627^{+}(±0.04)	**0.954^{+}(±0.01)**	0.628^{+}(±0.05)
	Synthesis	0.937^{+}(±0.02)	0.665^{+}(±0.06)	0.937^{+}(±0.02)	**0.668^{+}(±0.06)**	0.938^{+}(±0.02)	**0.678^{+}(±0.06)**	0.938^{+}(±0.01)	**0.668^{+}(±0.06)**
50	Local	0.509 (±0.00)	0.105 (±0.02)	0.503 (±0.01)	0.102 (±0.02)	0.509 (±0.00)	0.103 (±0.02)	0.507 (±0.00)	0.103 (±0.02)
	Federated	0.887^{+}(±0.04)	0.448^{+}(±0.06)	0.888^{+}(±0.04)	0.449^{+}(±0.06)	0.889^{+}(±0.04)	0.466^{+}(±0.06)	0.891^{+}(±0.04)	0.475^{+}(±0.05)
	Synthesis	**0.939^{+}(±0.02)**	**0.669^{+}(±0.06)**	**0.938^{+}(±0.02)**	**0.672^{+}(±0.06)**	**0.940^{+}(±0.01)**	**0.683^{+}(±0.06)**	**0.940^{+}(±0.01)**	**0.673^{+}(±0.06)**

Table A.2: Results for FFNN using Probe data set: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		Feature-based non-i.i.d.		Label-based non-i.i.d.		LDP non-i.i.d.	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	1.000 (±0.00)	0.998 (±0.00)	-	-	-	-	-	-
2	Local	0.999 (±0.00)	0.996 (±0.00)	0.999 (±0.00)	0.991 (±0.00)	0.974 (±0.00)	0.860 (±0.00)	0.999 (±0.00)	0.992 (±0.00)
	Federated	**1.000^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.999 (±0.00)**	**0.996^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.997^{+}(±0.00)**	**0.999 (±0.00)**	**0.997^{+}(±0.00)**
	Synthesis	0.959^{-}(±0.00)	0.850^{-}(±0.01)	0.887^{-}(±0.01)	0.615^{-}(±0.03)	0.890^{-}(±0.00)	0.629^{-}(±0.01)	0.885^{-}(±0.01)	0.607^{-}(±0.02)
5	Local	0.999 (±0.00)	0.991 (±0.00)	0.998 (±0.00)	0.986 (±0.00)	0.987 (±0.00)	0.920 (±0.00)	0.998 (±0.00)	0.985 (±0.00)
	Federated	**1.000^{+}(±0.00)**	**0.998^{+}(±0.00)**	**1.000^{+}(±0.00)**	**0.998^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.997^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.997^{+}(±0.00)**
	Synthesis	0.983^{-}(±0.00)	0.926^{-}(±0.01)	0.968^{-}(±0.00)	0.880^{-}(±0.01)	0.968^{-}(±0.00)	0.881^{-}(±0.01)	0.968^{-}(±0.00)	0.881^{-}(±0.01)
10	Local	0.997 (±0.00)	0.984 (±0.00)	0.997 (±0.00)	0.981 (±0.00)	0.980 (±0.00)	0.891 (±0.00)	0.996 (±0.00)	0.979 (±0.00)
	Federated	**0.999^{+}(±0.00)**	**0.997^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.996^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.996^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.996^{+}(±0.00)**
	Synthesis	0.988^{-}(±0.00)	0.942^{-}(±0.00)	0.981^{-}(±0.00)	0.919^{-}(±0.00)	0.981 (±0.00)	0.919^{+}(±0.00)	0.981^{-}(±0.00)	0.919^{-}(±0.00)
20	Local	0.996 (±0.00)	0.978 (±0.00)	0.995 (±0.00)	0.973 (±0.00)	0.982 (±0.00)	0.905 (±0.00)	0.995 (±0.00)	0.972 (±0.00)
	Federated	**0.999^{+}(±0.00)**	**0.996^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.994^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.995^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.994^{+}(±0.00)**
	Synthesis	0.991^{-}(±0.00)	0.951^{-}(±0.00)	0.986^{-}(±0.00)	0.935^{-}(±0.00)	0.986^{+}(±0.00)	0.937^{+}(±0.00)	0.986^{-}(±0.00)	0.938^{-}(±0.00)
30	Local	0.996 (±0.00)	0.975 (±0.00)	0.994 (±0.00)	0.968 (±0.00)	0.970 (±0.00)	0.866 (±0.00)	0.994 (±0.00)	0.965 (±0.00)
	Federated	**0.999^{+}(±0.00)**	**0.995^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.994^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.994^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.994^{+}(±0.00)**
	Synthesis	0.994^{-}(±0.00)	0.965^{-}(±0.00)	0.988^{-}(±0.00)	0.944^{-}(±0.00)	0.989^{+}(±0.00)	0.946^{+}(±0.00)	0.988^{-}(±0.00)	0.946^{-}(±0.00)
40	Local	0.994 (±0.00)	0.966 (±0.00)	0.991 (±0.00)	0.948 (±0.00)	0.978 (±0.00)	0.892 (±0.00)	0.992 (±0.00)	0.955 (±0.00)
	Federated	**0.999^{+}(±0.00)**	**0.994^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.993^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.993^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.993^{+}(±0.00)**
	Synthesis	0.994 (±0.00)	0.970 (±0.00)	0.990 (±0.00)	0.951 (±0.00)	0.990^{+}(±0.00)	0.951^{+}(±0.00)	0.990^{-}(±0.00)	0.951 (±0.00)
50	Local	0.994 (±0.00)	0.966 (±0.00)	0.989 (±0.00)	0.942 (±0.00)	0.979 (±0.00)	0.893 (±0.00)	0.990 (±0.00)	0.949 (±0.00)
	Federated	**0.998^{+}(±0.00)**	**0.988^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.992^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.992^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.990^{+}(±0.00)**
	Synthesis	0.995^{++}(±0.00)	0.973^{+}(±0.00)	0.991^{+}(±0.00)	0.953^{+}(±0.00)	0.991^{+}(±0.00)	0.953^{+}(±0.00)	0.991 (±0.00)	0.953 (±0.00)

Table A.3: Results for FFNN using R2L data set: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		**Feature-based non-i.i.d.**		**Label-based non-i.i.d.**		**LDP non-i.i.d.**	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.997 (±0.00)	0.960 (±0.00)	-	-	-	-	-	-
2	Local	0.995 (±0.00)	0.940 (±0.01)	0.995 (±0.00)	0.934 (±0.01)	0.863 (±0.00)	0.503 (±0.00)	0.993 (±0.00)	0.892 (±0.01)
	Federated	**0.998^{+}(±0.00)**	**0.971^{+}(±0.00)**	**0.997 (±0.00)**	**0.957^{++}(±0.01)**	**0.996^{+}(±0.00)**	**0.939^{+}(±0.01)**	**0.996^{+}(±0.00)**	**0.941^{+}(±0.01)**
	Synthesis	0.641^{-}(±0.01)	0.062^{-}(±0.00)	0.642^{-}(±0.01)	0.062^{-}(±0.00)	0.644^{-}(±0.00)	0.063^{-}(±0.00)	0.642^{-}(±0.00)	0.062^{-}(±0.00)
5	Local	0.992 (±0.00)	0.879 (±0.01)	0.991 (±0.00)	0.858 (±0.03)	0.891 (±0.00)	0.539 (±0.02)	0.990 (±0.00)	0.843 (±0.02)
	Federated	**0.996^{+}(±0.00)**	**0.952^{+}(±0.01)**	**0.996^{+}(±0.00)**	**0.947^{+}(±0.01)**	**0.995^{+}(±0.00)**	**0.931^{+}(±0.01)**	**0.996^{+}(±0.00)**	**0.934^{+}(±0.00)**
	Synthesis	0.733^{-}(±0.00)	0.088^{-}(±0.00)	0.733^{-}(±0.01)	0.088^{-}(±0.00)	0.735^{-}(±0.01)	0.089^{-}(±0.00)	0.733^{-}(±0.01)	0.088^{-}(±0.00)
10	Local	0.986 (±0.00)	0.753 (±0.01)	0.981 (±0.00)	0.717 (±0.06)	0.897 (±0.00)	0.522 (±0.01)	0.983 (±0.00)	0.704 (±0.02)
	Federated	**0.995^{+}(±0.00)**	**0.936^{+}(±0.01)**	**0.995^{+}(±0.00)**	**0.932^{+}(±0.01)**	**0.994^{+}(±0.00)**	**0.921^{+}(±0.01)**	**0.995^{+}(±0.00)**	**0.923^{+}(±0.01)**
	Synthesis	0.763^{-}(±0.01)	0.100^{-}(±0.01)	0.763^{-}(±0.01)	0.100^{-}(±0.01)	0.767^{-}(±0.01)	0.104^{-}(±0.01)	0.763^{-}(±0.00)	0.100^{-}(±0.01)
20	Local	0.957 (±0.00)	0.566 (±0.01)	0.952 (±0.01)	0.549 (±0.07)	0.903 (±0.00)	0.451 (±0.00)	0.969 (±0.00)	0.588 (±0.02)
	Federated	**0.993^{+}(±0.00)**	**0.901^{+}(±0.01)**	**0.992^{+}(±0.00)**	**0.878^{+}(±0.04)**	**0.994^{+}(±0.00)**	**0.915^{+}(±0.01)**	**0.994^{+}(±0.00)**	**0.918^{+}(±0.01)**
	Synthesis	0.789^{-}(±0.01)	0.115^{-}(±0.01)	0.790^{-}(±0.01)	0.117^{-}(±0.01)	0.798^{-}(±0.01)	0.125^{-}(±0.01)	0.790^{-}(±0.01)	0.116^{-}(±0.01)
30	Local	0.933 (±0.00)	0.471 (±0.01)	0.916 (±0.01)	0.397 (±0.07)	0.871 (±0.00)	0.364 (±0.03)	0.930 (±0.00)	0.449 (±0.02)
	Federated	**0.989^{+}(±0.00)**	**0.828^{+}(±0.02)**	**0.987^{+}(±0.00)**	**0.800^{+}(±0.09)**	**0.992^{+}(±0.00)**	**0.885^{+}(±0.01)**	**0.993^{+}(±0.00)**	**0.905^{+}(±0.01)**
	Synthesis	0.807^{-}(±0.01)	0.125^{-}(±0.01)	0.808^{-}(±0.01)	0.127^{-}(±0.01)	0.818^{-}(±0.00)	0.140^{-}(±0.01)	0.807^{-}(±0.00)	0.126^{-}(±0.01)
40	Local	0.899 (±0.00)	0.312 (±0.01)	0.897 (±0.00)	0.301 (±0.01)	0.856 (±0.00)	0.324 (±0.01)	0.913 (±0.00)	0.379 (±0.01)
	Federated	**0.986^{+}(±0.00)**	**0.783^{+}(±0.00)**	**0.987^{+}(±0.00)**	**0.792^{+}(±0.01)**	**0.991^{+}(±0.00)**	**0.866^{+}(±0.03)**	**0.992^{+}(±0.00)**	**0.865^{+}(±0.03)**
	Synthesis	0.820^{-}(±0.01)	0.136^{-}(±0.01)	0.822^{-}(±0.00)	0.140^{-}(±0.01)	0.834^{-}(±0.00)	0.155^{-}(±0.01)	0.820^{-}(±0.00)	0.138^{-}(±0.01)
50	Local	0.898 (±0.00)	0.317 (±0.01)	0.891 (±0.00)	0.309 (±0.01)	0.855 (±0.00)	0.301 (±0.02)	0.899 (±0.00)	0.336 (±0.02)
	Federated	**0.981^{+}(±0.00)**	**0.662^{+}(±0.01)**	**0.982^{+}(±0.00)**	**0.679^{+}(±0.03)**	**0.988^{+}(±0.00)**	**0.787^{+}(±0.03)**	**0.990^{+}(±0.00)**	**0.829^{+}(±0.03)**
	Synthesis	0.833^{-}(±0.01)	0.146^{-}(±0.01)	0.833^{-}(±0.01)	0.150^{-}(±0.01)	0.843^{-}(±0.00)	0.172^{-}(±0.01)	0.833^{-}(±0.01)	0.150^{-}(±0.01)

Table A.4: Results for FFNN using Annthyroid data set: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		Feature-based non-i.i.d.		Label-based non-i.i.d.		LDP non-i.i.d.	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.995 (±0.00)	0.954 (±0.03)	-	-	-	-	-	-
2	Local	0.976 (±0.01)	0.904 (±0.04)	0.966 (±0.01)	0.899 (±0.02)	0.792 (±0.02)	0.519 (±0.01)	0.961 (±0.01)	0.891 (±0.02)
	Federated	**0.984 (±0.01)**	**0.940 (±0.04)**	**0.976 (±0.01)**	**0.907 (±0.04)**	**0.978^{+}(±0.03)**	**0.864^{+}(±0.12)**	**0.979^{++}(±0.01)**	**0.930^{++}(±0.03)**
	Synthesis	0.641^{-} (±0.01)	0.062^{-} (±0.00)	0.642^{-} (±0.01)	0.062^{-} (±0.00)	0.644^{-} (±0.00)	0.063^{-} (±0.00)	0.642^{-} (±0.00)	0.062^{-} (±0.00)
5	Local	0.936 (±0.01)	0.841 (±0.02)	0.917 (±0.01)	0.789 (±0.04)	0.813 (±0.02)	0.522 (±0.02)	0.912 (±0.02)	0.770 (±0.04)
	Federated	**0.974^{+}(±0.01)**	**0.920^{+}(±0.03)**	**0.955^{+}(±0.01)**	**0.897^{+}(±0.04)**	**0.944^{+}(±0.01)**	**0.768^{+}(±0.05)**	**0.963^{+}(±0.01)**	**0.906^{+}(±0.02)**
	Synthesis	0.733^{-} (±0.00)	0.088^{-} (±0.00)	0.733^{-} (±0.01)	0.088^{-} (±0.00)	0.735^{-} (±0.01)	0.089^{-} (±0.00)	0.733^{-} (±0.01)	0.088^{-} (±0.00)
10	Local	0.901 (±0.02)	0.782 (±0.02)	0.869 (±0.02)	0.670 (±0.03)	0.801 (±0.02)	0.502 (±0.02)	0.866 (±0.02)	0.664 (±0.01)
	Federated	**0.964^{+}(±0.01)**	**0.903^{+}(±0.05)**	**0.959^{+}(±0.01)**	**0.872^{+}(±0.04)**	**0.890^{+}(±0.02)**	**0.683^{+}(±0.03)**	**0.969^{+}(±0.01)**	**0.876^{+}(±0.03)**
	Synthesis	0.763^{-} (±0.01)	0.100^{-} (±0.01)	0.763^{-} (±0.01)	0.100^{-} (±0.01)	0.767^{-} (±0.01)	0.104^{-} (±0.01)	0.763^{-} (±0.00)	0.100^{-} (±0.01)
20	Local	0.856 (±0.02)	0.660 (±0.03)	0.826 (±0.02)	0.562 (±0.02)	0.772 (±0.02)	0.441 (±0.01)	0.824 (±0.02)	0.562 (±0.01)
	Federated	**0.948^{+}(±0.02)**	**0.844^{+}(±0.04)**	**0.915^{+}(±0.04)**	**0.809^{+}(±0.06)**	0.777 (±0.04)	**0.516^{+}(±0.03)**	**0.923^{+}(±0.00)**	**0.818^{+}(±0.01)**
	Synthesis	0.789^{-} (±0.01)	0.115^{-} (±0.01)	0.790^{-} (±0.01)	0.117^{-} (±0.01)	**0.798 (±0.01)**	0.125^{-} (±0.01)	0.790^{--} (±0.01)	0.116^{-} (±0.01)
30	Local	0.819 (±0.02)	0.564 (±0.03)	0.798 (±0.02)	0.497 (±0.03)	0.753 (±0.03)	0.387 (±0.01)	0.795 (±0.02)	0.490 (±0.01)
	Federated	**0.904^{+}(±0.01)**	**0.786^{+}(±0.02)**	**0.895^{+}(±0.05)**	**0.753^{+}(±0.07)**	0.750 (±0.04)	**0.435^{+}(±0.02)**	**0.875^{+}(±0.02)**	**0.705^{+}(±0.03)**
	Synthesis	0.807 (±0.01)	0.125^{-} (±0.01)	0.808 (±0.01)	0.127^{-} (±0.01)	**0.818^{+}(±0.00)**	0.140^{-} (±0.01)	0.807 (±0.00)	0.126^{-} (±0.01)
40	Local	0.791 (±0.02)	0.471 (±0.03)	0.780 (±0.03)	0.462 (±0.04)	0.743 (±0.02)	0.371 (±0.01)	0.775 (±0.03)	0.443 (±0.01)
	Federated	**0.897^{+}(±0.02)**	**0.762^{+}(±0.04)**	**0.836 (±0.06)**	**0.662^{+}(±0.10)**	0.745 (±0.04)	**0.449^{+}(±0.01)**	**0.823 (±0.05)**	**0.623^{+}(±0.05)**
	Synthesis	0.820 (±0.01)	0.136^{-} (±0.01)	0.822^{+}(±0.00)	0.140^{-} (±0.01)	**0.834^{+}(±0.00)**	0.155^{-} (±0.01)	0.820^{+}(±0.00)	0.138^{-} (±0.01)
50	Local	0.769 (±0.03)	0.411 (±0.02)	0.768 (±0.02)	0.423 (±0.02)	0.729 (±0.03)	0.343 (±0.01)	0.768 (±0.02)	0.413 (±0.03)
	Federated	**0.849^{+}(±0.02)**	**0.679^{+}(±0.02)**	0.825 (±0.05)	**0.636^{+}(±0.07)**	0.729 (±0.05)	**0.371^{++}(±0.02)**	0.812 (±0.04)	**0.584^{+}(±0.04)**
	Synthesis	0.833^{+}(±0.01)	0.146^{-} (±0.01)	**0.833^{+}(±0.01)**	0.150^{-} (±0.01)	**0.843^{+}(±0.00)**	0.172^{-} (±0.01)	**0.833^{+}(±0.01)**	0.150^{-} (±0.01)

Table A.5: Results for Logistic Regression using Credit Card data set: Significance of federated learning and synthetic-based learning results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		Feature-based non-i.i.d.		Label-based non-i.i.d.		LDP non-i.i.d.	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.984 (±0.01)	0.720 (±0.05)	-	-	-	-	-	-
2	Local	**0.983 (±0.01)**	**0.718 (±0.05)**	**0.982 (±0.01)**	**0.716 (±0.05)**	0.947 (±0.01)	0.620 (±0.05)	0.968 (±0.01)	0.694 (±0.05)
	Federated	0.969 (±0.01)	0.705 (±0.06)	0.969 (±0.01)	0.706 (±0.05)	0.969^{++}(±0.01)	**0.705 (±0.06)**	0.969 (±0.01)	**0.705 (±0.06)**
	Synthesis	0.970^{--} (±0.00)	0.697 (±0.06)	0.970 (±0.01)	0.700 (±0.06)	0.972^{+}(±0.00)	0.700 (±0.05)	**0.972 (±0.00)**	0.697 (±0.06)
5	Local	**0.981 (±0.01)**	**0.714 (±0.05)**	**0.981 (±0.01)**	**0.711 (±0.06)**	0.952 (±0.01)	0.638 (±0.05)	**0.965 (±0.01)**	0.691 (±0.06)
	Federated	0.962^{--} (±0.01)	0.697 (±0.06)	0.962^{--} (±0.01)	0.697 (±0.06)	**0.964 (±0.01)**	**0.699 (±0.06)**	0.962 (±0.01)	**0.697 (±0.06)**
	Synthesis	0.928^{-}(±0.02)	0.651 (±0.05)	0.936^{-}(±0.02)	0.654 (±0.06)	0.948 (±0.01)	0.666 (±0.05)	0.939^{-}(±0.01)	0.654 (±0.05)
10	Local	**0.978 (±0.01)**	**0.707 (±0.06)**	**0.978 (±0.01)**	**0.701 (±0.06)**	0.957 (±0.01)	0.648 (±0.05)	0.961 (±0.01)	0.671 (±0.05)
	Federated	0.968 (±0.01)	0.700 (±0.06)	0.968 (±0.01)	0.699 (±0.05)	**0.968 (±0.01)**	**0.701 (±0.06)**	**0.968 (±0.01)**	**0.700 (±0.06)**
	Synthesis	0.944^{-}(±0.01)	0.680 (±0.05)	0.943^{-}(±0.01)	0.680 (±0.05)	0.945 (±0.01)	0.682 (±0.06)	0.945 (±0.01)	0.681 (±0.06)
20	Local	**0.973 (±0.01)**	0.690 (±0.06)	**0.972 (±0.01)**	0.680 (±0.06)	0.953 (±0.01)	0.640 (±0.06)	0.956 (±0.01)	0.677 (±0.06)
	Federated	0.971 (±0.01)	**0.701 (±0.05)**	0.971 (±0.01)	**0.701 (±0.06)**	**0.970 (±0.01)**	**0.702 (±0.05)**	**0.970 (±0.01)**	**0.701 (±0.05)**
	Synthesis	0.954^{-}(±0.01)	0.678 (±0.06)	0.956^{--} (±0.01)	0.679 (±0.06)	0.959 (±0.00)	0.686 (±0.06)	0.960 (±0.01)	0.681 (±0.06)
30	Local	**0.967 (±0.01)**	0.670 (±0.06)	**0.967 (±0.01)**	0.669 (±0.06)	0.948 (±0.01)	0.629 (±0.05)	0.952 (±0.01)	0.665 (±0.06)
	Federated	0.964 (±0.01)	**0.703 (±0.06)**	0.964 (±0.01)	**0.703 (±0.05)**	**0.965 (±0.01)**	**0.704 (±0.06)**	0.964 (±0.01)	**0.703 (±0.05)**
	Synthesis	0.967 (±0.00)	0.668 (±0.05)	0.960 (±0.01)	0.673 (±0.06)	0.963 (±0.01)	0.677 (±0.06)	**0.965 (±0.00)**	0.660 (±0.07)
40	Local	0.963 (±0.01)	0.660 (±0.06)	0.963 (±0.01)	0.650 (±0.06)	0.950 (±0.01)	0.626 (±0.05)	0.952 (±0.01)	0.661 (±0.06)
	Federated	**0.969 (±0.01)**	**0.705 (±0.06)**	**0.969 (±0.01)**	**0.705 (±0.06)**	0.968^{++}(±0.01)	**0.704 (±0.06)**	**0.968 (±0.01)**	**0.705 (±0.06)**
	Synthesis	0.943^{--} (±0.01)	0.653 (±0.05)	0.947^{--} (±0.01)	0.661 (±0.06)	0.946 (±0.01)	0.675 (±0.06)	0.947 (±0.01)	0.658 (±0.06)
50	Local	0.958 (±0.01)	0.644 (±0.06)	0.955 (±0.01)	0.636 (±0.06)	0.944 (±0.01)	0.607 (±0.05)	0.954 (±0.01)	0.666 (±0.05)
	Federated	**0.965 (±0.01)**	**0.701 (±0.06)**	**0.965 (±0.01)**	**0.701 (±0.06)**	**0.966 (±0.01)**	0.703^{++}(±0.06)	**0.965 (±0.01)**	**0.702 (±0.06)**
	Synthesis	0.928^{--} (±0.02)	0.587 (±0.04)	0.923^{--} (±0.02)	0.580 (±0.06)	0.939 (±0.01)	0.628 (±0.05)	0.930^{--} (±0.02)	0.608 (±0.06)

Table A.6: Results for Logistic Regression using Probe data set: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		Feature-based non-i.i.d.		Label-based non-i.i.d.		LDP non-i.i.d.	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.997 (±0.00)	0.980 (±0.00)	-	-	-	-	-	-
2	Local	0.996 (±0.00)	**0.978 (±0.00)**	0.996 (±0.00)	**0.978 (±0.00)**	0.985 (±0.00)	0.915 (±0.00)	0.996 (±0.00)	0.978 (±0.00)
	Federated	**0.996 (±0.00)**	0.977 (±0.00)	**0.996 (±0.00)**	0.977 (±0.00)	**0.996^{+}(±0.00)**	**0.977^{+}(±0.00)**	**0.996 (±0.00)**	**0.979 (±0.00)**
	Synthesis	0.995^{-} (±0.00)	0.975^{--} (±0.00)	0.995^{-} (±0.00)	0.974 (±0.00)	0.996^{+}(±0.00)	0.975^{+}(±0.00)	0.996 (±0.00)	0.975^{--} (±0.00)
5	Local	0.996 (±0.00)	**0.978 (±0.00)**	0.996 (±0.00)	0.977 (±0.00)	0.987 (±0.00)	0.926 (±0.00)	0.996 (±0.00)	0.978 (±0.00)
	Federated	**0.996 (±0.00)**	0.977 (±0.00)	**0.996 (±0.00)**	**0.977 (±0.00)**	**0.996^{+}(±0.00)**	**0.977^{+}(±0.00)**	**0.996 (±0.00)**	**0.979 (±0.00)**
	Synthesis	0.995^{-} (±0.00)	0.972^{-} (±0.00)	0.995^{-} (±0.00)	0.974^{--} (±0.00)	0.996^{+}(±0.00)	0.975^{+}(±0.00)	0.995^{-} (±0.00)	0.973^{-} (±0.00)
10	Local	0.996 (±0.00)	0.977 (±0.00)	0.996 (±0.00)	0.976 (±0.00)	0.989 (±0.00)	0.937 (±0.00)	0.996 (±0.00)	0.977 (±0.00)
	Federated	**0.996 (±0.00)**	**0.977 (±0.00)**	**0.996 (±0.00)**	**0.977 (±0.00)**	**0.996^{+}(±0.00)**	**0.977^{+}(±0.00)**	**0.996 (±0.00)**	**0.980^{++}(±0.00)**
	Synthesis	0.995^{-} (±0.00)	0.969^{--} (±0.00)	0.995^{-} (±0.00)	0.972^{--} (±0.00)	0.995^{+}(±0.00)	0.972^{+}(±0.00)	0.995^{-} (±0.00)	0.973^{-} (±0.00)
20	Local	0.996 (±0.00)	0.973 (±0.00)	0.995 (±0.00)	0.973 (±0.00)	0.989 (±0.00)	0.936 (±0.00)	0.996 (±0.00)	0.973 (±0.00)
	Federated	**0.996 (±0.00)**	**0.977^{++}(±0.00)**	**0.996^{+}(±0.00)**	**0.977^{++}(±0.00)**	**0.996^{+}(±0.00)**	**0.977^{+}(±0.00)**	**0.996 (±0.00)**	**0.979^{+}(±0.00)**
	Synthesis	0.995^{-} (±0.00)	0.973 (±0.00)	0.995 (±0.00)	0.974 (±0.00)	0.995^{+}(±0.00)	0.973^{+}(±0.00)	0.995^{-} (±0.00)	0.973 (±0.00)
30	Local	0.995 (±0.00)	0.971 (±0.00)	0.995 (±0.00)	0.970 (±0.00)	0.989 (±0.00)	0.934 (±0.00)	0.995 (±0.00)	0.970 (±0.00)
	Federated	**0.996^{+}(±0.00)**	**0.977^{+}(±0.00)**	**0.996^{+}(±0.00)**	**0.977^{+}(±0.00)**	**0.996^{+}(±0.00)**	**0.977^{+}(±0.00)**	**0.996^{+}(±0.00)**	**0.978^{+}(±0.00)**
	Synthesis	0.995 (±0.00)	0.973 (±0.00)	0.995 (±0.00)	0.973^{+}(±0.00)	0.995^{+}(±0.00)	0.974^{+}(±0.00)	0.995 (±0.00)	0.973^{+}(±0.00)
40	Local	0.995 (±0.00)	0.969 (±0.00)	0.995 (±0.00)	0.968 (±0.00)	0.988 (±0.00)	0.932 (±0.00)	0.995 (±0.00)	0.968 (±0.00)
	Federated	**0.996^{+}(±0.00)**	**0.977^{+}(±0.00)**	**0.996^{+}(±0.00)**	**0.977^{+}(±0.00)**	**0.996^{+}(±0.00)**	**0.977^{+}(±0.00)**	**0.996^{+}(±0.00)**	**0.978^{+}(±0.00)**
	Synthesis	0.995 (±0.00)	0.973^{+}(±0.00)	0.995 (±0.00)	0.974^{+}(±0.00)	0.995^{+}(±0.00)	0.974^{+}(±0.00)	0.995 (±0.00)	0.973^{+}(±0.00)
50	Local	0.995 (±0.00)	0.966 (±0.00)	0.994 (±0.00)	0.964 (±0.00)	0.988 (±0.00)	0.930 (±0.00)	0.994 (±0.00)	0.966 (±0.00)
	Federated	**0.996^{+}(±0.00)**	**0.977^{+}(±0.00)**	**0.996^{+}(±0.00)**	**0.977^{+}(±0.00)**	**0.996^{+}(±0.00)**	**0.977^{+}(±0.00)**	**0.996^{+}(±0.00)**	**0.977^{+}(±0.00)**
	Synthesis	0.995 (±0.00)	0.972^{+}(±0.00)	0.995^{+}(±0.00)	0.973^{+}(±0.00)	0.995^{+}(±0.00)	0.974^{+}(±0.00)	0.995 (±0.00)	0.973^{+}(±0.00)

Table A.7: Results for Logistic Regression using R2L data set: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		**Feature-based non-i.i.d.**		**Label-based non-i.i.d.**		**LDP non-i.i.d.**	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.985 (±0.00)	0.762 (±0.01)	-	-	-	-	-	-
2	Local	0.984 (±0.00)	0.756 (±0.01)	**0.984 (±0.00)**	**0.756 (±0.01)**	0.870 (±0.00)	0.418 (±0.01)	0.984 (±0.00)	**0.756 (±0.01)**
	Federated	**0.984 (±0.00)**	**0.759 (±0.01)**	0.982 (±0.00)	0.735^{-} (±0.01)	**0.984^{+}(±0.00)**	**0.734^{+}(±0.00)**	**0.984 (±0.00)**	0.751 (±0.02)
	Synthesis	0.977^{-} (±0.00)	0.690^{-} (±0.00)	0.978^{-} (±0.00)	0.698^{-} (±0.01)	0.980^{+}(±0.00)	0.721^{+}(±0.02)	0.977^{-} (±0.00)	0.695^{-} (±0.01)
5	Local	0.984 (±0.00)	0.752 (±0.01)	0.983 (±0.00)	**0.751 (±0.01)**	0.888 (±0.00)	0.482 (±0.01)	0.983 (±0.00)	**0.754 (±0.01)**
	Federated	**0.984 (±0.00)**	**0.758 (±0.01)**	**0.983 (±0.00)**	0.743 (±0.01)	**0.984^{+}(±0.00)**	**0.728^{+}(±0.01)**	**0.984 (±0.00)**	0.728^{-} (±0.01)
	Synthesis	0.974^{-} (±0.00)	0.643^{-} (±0.02)	0.975^{-} (±0.00)	0.668^{-} (±0.01)	0.977^{+}(±0.00)	0.692^{+}(±0.01)	0.973^{-} (±0.00)	0.645^{-} (±0.01)
10	Local	0.983 (±0.00)	0.746 (±0.01)	0.983 (±0.00)	0.747 (±0.01)	0.914 (±0.00)	0.545 (±0.01)	0.983 (±0.00)	**0.745 (±0.01)**
	Federated	**0.984 (±0.00)**	**0.760^{++}(±0.01)**	**0.984 (±0.00)**	**0.758 (±0.01)**	**0.984^{+}(±0.00)**	**0.729^{+}(±0.01)**	**0.984 (±0.00)**	0.740 (±0.01)
	Synthesis	0.970^{-} (±0.00)	0.622^{-} (±0.02)	0.970^{-} (±0.00)	0.619^{-} (±0.01)	0.975^{+}(±0.00)	0.668^{+}(±0.01)	0.970^{-} (±0.00)	0.604^{-} (±0.01)
20	Local	0.982 (±0.00)	0.730 (±0.01)	0.982 (±0.00)	0.736 (±0.01)	0.913 (±0.00)	0.536 (±0.01)	0.982 (±0.00)	0.734 (±0.01)
	Federated	**0.983 (±0.00)**	**0.757^{+}(±0.00)**	**0.983 (±0.00)**	**0.758^{+}(±0.00)**	**0.984^{+}(±0.00)**	**0.730^{+}(±0.02)**	**0.984 (±0.00)**	**0.738 (±0.01)**
	Synthesis	0.961^{-} (±0.00)	0.531^{-} (±0.02)	0.966^{-} (±0.00)	0.568^{-} (±0.01)	0.969^{+}(±0.00)	0.596^{+}(±0.02)	0.964^{-} (±0.00)	0.544^{-} (±0.02)
30	Local	0.981 (±0.00)	0.721 (±0.01)	0.981 (±0.00)	0.720 (±0.01)	0.912 (±0.00)	0.525 (±0.01)	0.981 (±0.00)	0.720 (±0.01)
	Federated	**0.983 (±0.00)**	**0.757^{+}(±0.01)**	**0.983 (±0.00)**	**0.754^{+}(±0.00)**	**0.984^{+}(±0.00)**	**0.737^{+}(±0.01)**	**0.984^{++}(±0.00)**	**0.741^{+}(±0.01)**
	Synthesis	0.959^{-} (±0.00)	0.499^{-} (±0.03)	0.963^{-} (±0.00)	0.518^{-} (±0.02)	0.968^{+}(±0.00)	0.570 (±0.04)	0.963^{-} (±0.00)	0.531^{-} (±0.01)
40	Local	0.980 (±0.00)	0.710 (±0.01)	0.980 (±0.00)	0.712 (±0.01)	0.912 (±0.00)	0.521 (±0.01)	0.979 (±0.00)	0.708 (±0.01)
	Federated	**0.983^{++}(±0.00)**	**0.756^{+}(±0.00)**	**0.983 (±0.00)**	**0.754^{+}(±0.00)**	**0.984^{+}(±0.00)**	**0.739^{+}(±0.01)**	**0.984^{+}(±0.00)**	**0.741^{+}(±0.01)**
	Synthesis	0.956^{-} (±0.00)	0.460^{-} (±0.03)	0.959^{-} (±0.00)	0.501^{-} (±0.01)	0.966^{+}(±0.00)	0.542^{+}(±0.01)	0.961^{-} (±0.01)	0.498^{-} (±0.04)
50	Local	0.979 (±0.00)	0.701 (±0.01)	0.978 (±0.00)	0.699 (±0.00)	0.910 (±0.00)	0.514 (±0.01)	0.978 (±0.00)	0.694 (±0.00)
	Federated	**0.983^{+}(±0.00)**	**0.755^{+}(±0.00)**	**0.983^{++}(±0.00)**	**0.758^{+}(±0.00)**	**0.983^{+}(±0.00)**	**0.731^{+}(±0.01)**	**0.983^{+}(±0.00)**	**0.745^{+}(±0.00)**
	Synthesis	0.953^{-} (±0.00)	0.449^{-} (±0.02)	0.955^{-} (±0.00)	0.461^{-} (±0.01)	0.964^{+}(±0.00)	0.542 (±0.03)	0.958^{-} (±0.00)	0.469^{-} (±0.02)

Table A.8: Results for Logistic Regression using Annthyroid data set: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		**Feature-based non-i.i.d.**		**Label-based non-i.i.d.**		**LDP non-i.i.d.**	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.956 (±0.01)	0.779 (±0.01)	-	-	-	-	-	-
2	Local	0.907 (±0.01)	0.630 (±0.02)	0.864 (±0.02)	0.541 (±0.02)	0.740 (±0.03)	0.336 (±0.02)	0.865 (±0.01)	0.540 (±0.02)
	Federated	**0.956^{+}(±0.01)**	**0.781^{+}(±0.01)**	0.880 (±0.02)	0.602^{++}(±0.03)	0.884^{+}(±0.02)	0.576^{+}(±0.03)	**0.918^{+}(±0.02)**	**0.638^{+}(±0.03)**
	Synthesis	0.879 (±0.03)	0.608 (±0.09)	**0.895 (±0.02)**	**0.604 (±0.07)**	**0.909^{+}(±0.02)**	**0.629^{+}(±0.04)**	0.904^{+}(±0.02)	0.612^{+}(±0.03)
5	Local	0.896 (±0.01)	0.617 (±0.02)	0.849 (±0.02)	0.521 (±0.02)	0.757 (±0.03)	0.368 (±0.02)	0.854 (±0.01)	0.531 (±0.02)
	Federated	**0.933^{+}(±0.01)**	**0.690^{+}(±0.02)**	0.892^{+}(±0.01)	0.608^{+}(±0.03)	0.892^{+}(±0.01)	0.585^{+}(±0.02)	0.920^{+}(±0.02)	0.647^{+}(±0.03)
	Synthesis	0.917^{+}(±0.00)	0.640 (±0.03)	**0.911^{+}(±0.02)**	**0.632^{+}(±0.05)**	**0.926^{+}(±0.02)**	**0.660^{+}(±0.04)**	**0.926^{+}(±0.01)**	**0.654^{+}(±0.03)**
10	Local	0.875 (±0.01)	0.588 (±0.02)	0.836 (±0.01)	0.497 (±0.00)	0.772 (±0.02)	0.393 (±0.02)	0.840 (±0.01)	0.504 (±0.02)
	Federated	0.909^{+}(±0.01)	0.625^{++}(±0.02)	0.894^{+}(±0.02)	0.596^{+}(±0.04)	0.894^{+}(±0.02)	0.587^{+}(±0.03)	0.907^{+}(±0.01)	0.616^{+}(±0.01)
	Synthesis	**0.923^{+}(±0.01)**	**0.647^{++}(±0.04)**	**0.904^{+}(±0.03)**	**0.615^{+}(±0.05)**	**0.924^{+}(±0.01)**	**0.643^{+}(±0.04)**	**0.915^{+}(±0.01)**	**0.636^{+}(±0.04)**
20	Local	0.850 (±0.01)	0.543 (±0.01)	0.813 (±0.01)	0.460 (±0.01)	0.758 (±0.02)	0.371 (±0.01)	0.812 (±0.02)	0.456 (±0.01)
	Federated	0.890^{+}(±0.01)	0.587^{+}(±0.02)	0.888^{+}(±0.02)	0.585^{+}(±0.03)	0.885^{+}(±0.01)	0.578^{+}(±0.02)	0.889^{+}(±0.01)	0.579^{+}(±0.02)
	Synthesis	**0.915^{+}(±0.03)**	**0.647^{+}(±0.05)**	**0.904^{+}(±0.04)**	**0.630^{+}(±0.09)**	**0.921^{+}(±0.03)**	**0.659^{+}(±0.04)**	**0.917^{+}(±0.02)**	**0.643^{+}(±0.02)**
30	Local	0.829 (±0.01)	0.507 (±0.01)	0.792 (±0.01)	0.412 (±0.02)	0.747 (±0.03)	0.345 (±0.01)	0.791 (±0.02)	0.410 (±0.01)
	Federated	**0.882^{+}(±0.01)**	0.566^{+}(±0.02)	**0.888^{+}(±0.02)**	0.583^{+}(±0.02)	0.878^{+}(±0.02)	0.561^{+}(±0.01)	0.885^{+}(±0.01)	0.576^{+}(±0.01)
	Synthesis	0.873^{++}(±0.03)	**0.573 (±0.06)**	0.887^{+}(±0.02)	**0.584^{+}(±0.04)**	**0.924^{+}(±0.02)**	**0.658^{+}(±0.05)**	**0.914^{+}(±0.01)**	**0.640^{+}(±0.03)**
40	Local	0.814 (±0.02)	0.476 (±0.02)	0.777 (±0.02)	0.388 (±0.03)	0.737 (±0.02)	0.331 (±0.01)	0.778 (±0.02)	0.388 (±0.02)
	Federated	0.874^{+}(±0.02)	0.562^{+}(±0.03)	**0.872^{+}(±0.02)**	**0.566^{+}(±0.02)**	0.874^{+}(±0.01)	0.556^{+}(±0.02)	0.873^{+}(±0.02)	**0.559^{+}(±0.03)**
	Synthesis	**0.897^{+}(±0.01)**	**0.588^{+}(±0.02)**	0.869^{+}(±0.03)	0.529^{+}(±0.05)	**0.886^{+}(±0.03)**	**0.579^{+}(±0.06)**	**0.876^{+}(±0.02)**	0.547^{+}(±0.03)
50	Local	0.804 (±0.02)	0.453 (±0.02)	0.766 (±0.02)	0.368 (±0.02)	0.729 (±0.03)	0.313 (±0.02)	0.769 (±0.02)	0.367 (±0.03)
	Federated	**0.868^{+}(±0.01)**	**0.543^{+}(±0.02)**	**0.859^{+}(±0.01)**	**0.542^{+}(±0.01)**	**0.873^{+}(±0.01)**	**0.557^{+}(±0.01)**	**0.863^{+}(±0.02)**	**0.551^{+}(±0.01)**
	Synthesis	0.844^{+}(±0.01)	0.501^{++}(±0.02)	0.858^{+}(±0.04)	0.498^{++}(±0.09)	0.855^{+}(±0.04)	0.513^{+}(±0.08)	0.847^{+}(±0.03)	0.518^{+}(±0.06)

Table A.9: Results for XGBoost using Credit Card data set: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		**Feature-based non-i.i.d.**		**Label-based non-i.i.d.**		**LDP non-i.i.d.**	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.974 (±0.00)	0.847 (±0.03)	-	-	-	-	-	-
2	Local	0.978 (±0.00)	0.831 (±0.03)	0.974 (±0.00)	0.820 (±0.04)	0.738 (±0.00)	0.419 (±0.01)	0.975 (±0.00)	0.825 (±0.03)
	Federated	**0.981 (±0.00)**	**0.841 (±0.03)**	**0.982^{+}(±0.00)**	**0.835 (±0.03)**	**0.977^{+}(±0.00)**	**0.837^{+}(±0.03)**	**0.981 (±0.01)**	**0.837 (±0.03)**
	Synthesis	0.915^{-} (±0.02)	0.695^{-} (±0.06)	0.919^{-} (±0.03)	0.714^{--} (±0.06)	0.918^{+}(±0.03)	0.722^{+}(±0.01)	0.918^{-} (±0.03)	0.715^{-} (±0.03)
5	Local	0.969 (±0.01)	0.793 (±0.03)	0.970 (±0.00)	0.794 (±0.03)	0.784 (±0.00)	0.480 (±0.02)	0.970 (±0.01)	0.791 (±0.04)
	Federated	**0.982 (±0.01)**	**0.831 (±0.03)**	**0.982^{+}(±0.01)**	**0.814 (±0.04)**	**0.986^{+}(±0.01)**	**0.825^{+}(±0.03)**	**0.981 (±0.01)**	**0.819 (±0.03)**
	Synthesis	0.909^{-} (±0.02)	0.605^{-} (±0.05)	0.921^{-} (±0.02)	0.622^{-} (±0.08)	0.913^{+}(±0.03)	0.652^{+}(±0.06)	0.913^{-} (±0.02)	0.622^{-} (±0.04)
10	Local	0.964 (±0.01)	0.765 (±0.04)	0.962 (±0.01)	0.758 (±0.04)	0.828 (±0.01)	0.538 (±0.03)	0.963 (±0.01)	0.749 (±0.05)
	Federated	**0.983^{+}(±0.00)**	**0.812 (±0.03)**	**0.982^{+}(±0.01)**	**0.810 (±0.04)**	**0.982^{+}(±0.01)**	**0.813^{+}(±0.03)**	**0.981^{++}(±0.01)**	**0.812 (±0.03)**
	Synthesis	0.921^{-} (±0.02)	0.629^{--} (±0.09)	0.915^{--} (±0.03)	0.611^{-} (±0.06)	0.922^{+}(±0.02)	0.610 (±0.07)	0.913^{-} (±0.03)	0.620^{-} (±0.06)
20	Local	0.955 (±0.01)	0.713 (±0.04)	0.952 (±0.01)	0.712 (±0.05)	0.822 (±0.01)	0.511 (±0.03)	0.954 (±0.01)	0.705 (±0.05)
	Federated	**0.982^{+}(±0.01)**	**0.790^{++}(±0.03)**	**0.977^{++}(±0.01)**	**0.793^{++}(±0.04)**	**0.981^{+}(±0.01)**	**0.801^{+}(±0.04)**	**0.978^{++}(±0.01)**	**0.787^{++}(±0.03)**
	Synthesis	0.923^{--} (±0.02)	0.611^{--} (±0.06)	0.922^{--} (±0.01)	0.668 (±0.05)	0.928^{+}(±0.01)	0.636^{+}(±0.05)	0.920^{--} (±0.02)	0.601^{-} (±0.03)
30	Local	0.949 (±0.02)	0.677 (±0.05)	0.950 (±0.01)	0.679 (±0.04)	0.818 (±0.01)	0.486 (±0.03)	0.949 (±0.01)	0.684 (±0.05)
	Federated	**0.979^{+}(±0.01)**	**0.776^{+}(±0.04)**	**0.980^{+}(±0.01)**	**0.777^{+}(±0.04)**	**0.979^{+}(±0.01)**	**0.792^{+}(±0.04)**	**0.979^{+}(±0.01)**	**0.782^{++}(±0.04)**
	Synthesis	0.921 (±0.02)	0.610 (±0.04)	0.922 (±0.02)	0.639 (±0.10)	0.922^{+}(±0.02)	0.601^{+}(±0.03)	0.922^{--} (±0.02)	0.642 (±0.08)
40	Local	0.947 (±0.02)	0.655 (±0.05)	0.948 (±0.01)	0.655 (±0.06)	0.818 (±0.01)	0.478 (±0.03)	0.946 (±0.02)	0.663 (±0.05)
	Federated	**0.980^{+}(±0.01)**	**0.767^{+}(±0.04)**	**0.982^{+}(±0.00)**	**0.764^{++}(±0.04)**	**0.978^{+}(±0.01)**	**0.770^{+}(±0.04)**	**0.977^{+}(±0.01)**	**0.768^{+}(±0.04)**
	Synthesis	0.924 (±0.02)	0.526^{-} (±0.05)	0.925 (±0.02)	0.558 (±0.07)	0.923^{+}(±0.01)	0.612^{+}(±0.06)	0.930 (±0.01)	0.530^{--} (±0.09)
50	Local	0.946 (±0.02)	0.640 (±0.06)	0.947 (±0.01)	0.639 (±0.05)	0.816 (±0.01)	0.460 (±0.04)	0.945 (±0.01)	0.638 (±0.05)
	Federated	**0.979^{+}(±0.01)**	**0.757^{+}(±0.04)**	**0.976^{+}(±0.01)**	**0.756^{+}(±0.04)**	**0.978^{+}(±0.01)**	**0.772^{+}(±0.04)**	**0.979^{+}(±0.00)**	**0.750^{+}(±0.04)**
	Synthesis	0.928 (±0.01)	0.532 (±0.13)	0.920 (±0.02)	0.567 (±0.14)	0.922^{+}(±0.02)	0.642^{+}(±0.02)	0.922 (±0.02)	0.611 (±0.07)

Table A.10: Results for XGBoost using Probe data set: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		**Feature-based non-i.i.d.**		**Label-based non-i.i.d.**		**LDP non-i.i.d.**	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	1.000 (±0.00)	1.000 (±0.00)	-	-	-	-	-	-
2	Local	1.000 (±0.00)	0.999 (±0.00)	1.000 (±0.00)	0.999 (±0.00)	0.750 (±0.00)	0.578 (±0.00)	1.000 (±0.00)	0.999 (±0.00)
	Federated	**1.000 (±0.00)**	**1.000^{+}(±0.00)**	**1.000 (±0.00)**	**0.999 (±0.00)**	**1.000^{+}(±0.00)**	**0.999^{+}(±0.00)**	**1.000 (±0.00)**	**1.000^{+}(±0.00)**
	Synthesis	0.999^{-}(±0.00)	0.997^{-}(±0.00)	0.999^{-}(±0.00)	0.997^{-}(±0.00)	1.000^{+}(±0.00)	0.998^{+}(±0.00)	0.999^{-}(±0.00)	0.997^{-}(±0.00)
5	Local	1.000 (±0.00)	0.998 (±0.00)	1.000 (±0.00)	0.998 (±0.00)	0.800 (±0.00)	0.662 (±0.00)	1.000 (±0.00)	0.998 (±0.00)
	Federated	**1.000 (±0.00)**	**0.999^{+}(±0.00)**	**1.000 (±0.00)**	**0.999^{+}(±0.00)**	**1.000^{+}(±0.00)**	**0.999^{+}(±0.00)**	**1.000 (±0.00)**	**0.999^{+}(±0.00)**
	Synthesis	0.999^{-}(±0.00)	0.996^{-}(±0.00)	0.999^{-}(±0.00)	0.996^{-}(±0.00)	0.999^{+}(±0.00)	0.997^{+}(±0.00)	0.999^{-}(±0.00)	0.995^{-}(±0.00)
10	Local	0.999 (±0.00)	0.996 (±0.00)	0.999 (±0.00)	0.996 (±0.00)	0.849 (±0.00)	0.744 (±0.00)	0.999 (±0.00)	0.996 (±0.00)
	Federated	**1.000^{+}(±0.00)**	**0.999^{+}(±0.00)**	**1.000^{+}(±0.00)**	**0.999^{+}(±0.00)**	**1.000^{+}(±0.00)**	**0.999^{+}(±0.00)**	**1.000^{+}(±0.00)**	**0.999^{+}(±0.00)**
	Synthesis	0.998^{-}(±0.00)	0.992^{-}(±0.00)	0.999 (±0.00)	0.993^{-}(±0.00)	0.999^{+}(±0.00)	0.996^{+}(±0.00)	0.999 (±0.00)	0.993^{-}(±0.00)
20	Local	0.999 (±0.00)	0.994 (±0.00)	0.999 (±0.00)	0.994 (±0.00)	0.849 (±0.00)	0.743 (±0.00)	0.999 (±0.00)	0.993 (±0.00)
	Federated	**1.000^{+}(±0.00)**	**0.998^{+}(±0.00)**	**1.000^{+}(±0.00)**	**0.998^{+}(±0.00)**	**1.000^{+}(±0.00)**	**0.998^{+}(±0.00)**	**1.000^{+}(±0.00)**	**0.998^{+}(±0.00)**
	Synthesis	0.998^{-}(±0.00)	0.990^{-}(±0.00)	0.998^{-}(±0.00)	0.990^{-}(±0.00)	0.998^{+}(±0.00)	0.992^{+}(±0.00)	0.998^{-}(±0.00)	0.991^{-}(±0.00)
30	Local	0.998 (±0.00)	0.991 (±0.00)	0.998 (±0.00)	0.991 (±0.00)	0.849 (±0.00)	0.741 (±0.00)	0.998 (±0.00)	0.991 (±0.00)
	Federated	**1.000^{+}(±0.00)**	**0.998^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.998^{+}(±0.00)**	**1.000^{+}(±0.00)**	**0.998^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.997^{+}(±0.00)**
	Synthesis	0.997^{-}(±0.00)	0.985^{-}(±0.00)	0.998 (±0.00)	0.988^{-}(±0.00)	0.998^{+}(±0.00)	0.991^{+}(±0.00)	0.997^{-}(±0.00)	0.986^{-}(±0.00)
40	Local	0.998 (±0.00)	0.990 (±0.00)	0.998 (±0.00)	0.989 (±0.00)	0.849 (±0.00)	0.740 (±0.00)	0.998 (±0.00)	0.989 (±0.00)
	Federated	**0.999^{+}(±0.00)**	**0.997^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.997^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.997^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.997^{+}(±0.00)**
	Synthesis	0.997^{-}(±0.00)	0.985^{-}(±0.00)	0.997$^{- -}$(±0.00)	0.987^{-}(±0.00)	0.998^{+}(±0.00)	0.989^{+}(±0.00)	0.997 (±0.00)	0.986 (±0.00)
50	Local	0.998 (±0.00)	0.988 (±0.00)	0.997 (±0.00)	0.988 (±0.00)	0.848 (±0.00)	0.738 (±0.00)	0.997 (±0.00)	0.987 (±0.00)
	Federated	**0.999^{+}(±0.00)**	**0.997^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.997^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.997^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.996^{+}(±0.00)**
	Synthesis	0.997^{-}(±0.00)	0.985^{-}(±0.00)	0.997 (±0.00)	0.986 (±0.00)	0.998^{+}(±0.00)	0.990^{+}(±0.00)	0.997 (±0.00)	0.986 (±0.00)

Table A.11: Results for XGBoost using R2L data set: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		Feature-based non-i.i.d.		Label-based non-i.i.d.		LDP non-i.i.d.	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.999 (±0.00)	0.990 (±0.00)	-	-	-	-	-	-
2	Local	0.999 (±0.00)	0.987 (±0.00)	0.999 (±0.00)	0.987 (±0.00)	0.750 (±0.00)	0.518 (±0.00)	0.999 (±0.00)	0.987 (±0.00)
	Federated	**0.999 (±0.00)**	**0.989 (±0.00)**	**0.999 (±0.00)**	**0.989 (±0.00)**	**0.999^{+}(±0.00)**	**0.989^{+}(±0.00)**	**0.999 (±0.00)**	**0.989 (±0.00)**
	Synthesis	0.997^{-} (±0.00)	0.964^{-} (±0.01)	0.997^{-} (±0.00)	0.961^{-} (±0.01)	0.997^{+}(±0.00)	0.973^{+}(±0.01)	0.997^{-} (±0.00)	0.961^{-} (±0.01)
5	Local	0.998 (±0.00)	0.981 (±0.00)	0.998 (±0.00)	0.980 (±0.00)	0.799 (±0.00)	0.609 (±0.00)	0.998 (±0.00)	0.979 (±0.01)
	Federated	**0.999 (±0.00)**	**0.988 (±0.00)**	**0.999 (±0.00)**	**0.987 (±0.00)**	**0.999^{+}(±0.00)**	**0.987^{+}(±0.00)**	**0.999 (±0.00)**	**0.987 (±0.00)**
	Synthesis	0.996^{-} (±0.00)	0.949^{-} (±0.01)	0.995^{-} (±0.00)	0.947^{-} (±0.01)	0.996^{+}(±0.00)	0.961^{+}(±0.01)	0.995^{-} (±0.00)	0.944^{-} (±0.01)
10	Local	0.998 (±0.00)	0.973 (±0.01)	0.997 (±0.00)	0.971 (±0.01)	0.848 (±0.00)	0.696 (±0.00)	0.997 (±0.00)	0.971 (±0.01)
	Federated	**0.999 (±0.00)**	**0.985^{++}(±0.01)**	**0.999^{++}(±0.00)**	**0.984^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.985^{+}(±0.00)**	**0.999^{++}(±0.00)**	**0.985^{+}(±0.00)**
	Synthesis	0.992^{-} (±0.00)	0.903^{-} (±0.02)	0.993^{-} (±0.00)	0.909^{-} (±0.01)	0.995^{+}(±0.00)	0.942^{+}(±0.00)	0.992^{-} (±0.00)	0.891^{-} (±0.00)
20	Local	0.996 (±0.00)	0.956 (±0.01)	0.995 (±0.00)	0.951 (±0.01)	0.848 (±0.00)	0.687 (±0.00)	0.995 (±0.00)	0.953 (±0.00)
	Federated	**0.998^{+}(±0.00)**	**0.979^{+}(±0.00)**	**0.998^{+}(±0.00)**	**0.978^{+}(±0.00)**	**0.999^{+}(±0.00)**	**0.980^{+}(±0.00)**	**0.998^{+}(±0.00)**	**0.978^{+}(±0.01)**
	Synthesis	0.986^{-} (±0.00)	0.755^{-} (±0.05)	0.989^{-} (±0.00)	0.836^{-} (±0.04)	0.992^{+}(±0.00)	0.891^{+}(±0.03)	0.989^{-} (±0.00)	0.828^{-} (±0.03)
30	Local	0.994 (±0.00)	0.937 (±0.01)	0.993 (±0.00)	0.930 (±0.01)	0.846 (±0.00)	0.675 (±0.00)	0.993 (±0.00)	0.932 (±0.01)
	Federated	**0.998^{+}(±0.00)**	**0.972^{+}(±0.01)**	**0.998^{+}(±0.00)**	**0.971^{+}(±0.01)**	**0.998^{+}(±0.00)**	**0.974^{+}(±0.01)**	**0.998^{+}(±0.00)**	**0.970^{+}(±0.01)**
	Synthesis	0.980^{-} (±0.01)	0.743^{-} (±0.08)	0.984^{-} (±0.00)	0.759^{-} (±0.00)	0.990^{+}(±0.00)	0.876^{+}(±0.03)	0.986^{-} (±0.00)	0.816^{-} (±0.03)
40	Local	0.992 (±0.00)	0.919 (±0.01)	0.991 (±0.00)	0.913 (±0.01)	0.845 (±0.00)	0.663 (±0.00)	0.991 (±0.00)	0.912 (±0.01)
	Federated	**0.997^{+}(±0.00)**	**0.965^{+}(±0.01)**	**0.997^{+}(±0.00)**	**0.964^{+}(±0.01)**	**0.997^{+}(±0.00)**	**0.967^{+}(±0.01)**	**0.997^{+}(±0.00)**	**0.964^{+}(±0.01)**
	Synthesis	0.979^{-} (±0.00)	0.720^{-} (±0.04)	0.982^{-} (±0.00)	0.743^{-} (±0.02)	0.989^{+}(±0.00)	0.869^{+}(±0.03)	0.982^{-} (±0.00)	0.781^{-} (±0.03)
50	Local	0.990 (±0.00)	0.902 (±0.01)	0.989 (±0.00)	0.891 (±0.01)	0.844 (±0.00)	0.649 (±0.00)	0.989 (±0.00)	0.895 (±0.01)
	Federated	**0.997^{+}(±0.00)**	**0.958^{+}(±0.01)**	**0.997^{+}(±0.00)**	**0.957^{+}(±0.01)**	**0.997^{+}(±0.00)**	**0.962^{+}(±0.01)**	**0.997^{+}(±0.00)**	**0.957^{+}(±0.01)**
	Synthesis	0.975^{-} (±0.00)	0.661^{-} (±0.05)	0.978^{-} (±0.00)	0.710^{-} (±0.05)	0.989^{+}(±0.00)	0.879^{+}(±0.01)	0.982^{-} (±0.00)	0.755^{-} (±0.02)

Table A.12: Results for XGBoost using Annthyroid data set: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		**Feature-based non-i.i.d.**		**Label-based non-i.i.d.**		**LDP non-i.i.d.**	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.999 (±0.00)	0.984 (±0.01)	-	-	-	-	-	-
2	Local	0.999 (±0.00)	0.980 (±0.00)	0.999 (±0.00)	0.978 (±0.01)	0.750 (±0.00)	0.527 (±0.00)	0.999 (±0.00)	0.978 (±0.02)
	Federated	**0.999 (±0.00)**	**0.980 (±0.00)**	**0.999 (±0.00)**	**0.982 (±0.02)**	**0.999^{+}(±0.00)**	**0.979^{+}(±0.01)**	**0.999 (±0.00)**	**0.982 (±0.01)**
	Synthesis	0.997^{-} (±0.00)	0.950^{-} (±0.01)	0.997^{--} (±0.00)	0.938 (±0.02)	0.998^{+}(±0.00)	0.966^{+}(±0.03)	0.997^{-} (±0.00)	0.954^{--} (±0.01)
5	Local	0.998 (±0.00)	0.963 (±0.01)	0.998 (±0.00)	0.964 (±0.01)	0.799 (±0.00)	0.605 (±0.01)	0.996 (±0.00)	0.956 (±0.01)
	Federated	**0.999 (±0.00)**	**0.976 (±0.02)**	**0.999 (±0.00)**	**0.977 (±0.01)**	**0.999^{+}(±0.00)**	**0.981^{+}(±0.01)**	**0.999 (±0.00)**	**0.975 (±0.01)**
	Synthesis	0.994^{-} (±0.00)	0.906^{--} (±0.03)	0.993^{--} (±0.00)	0.883^{--} (±0.04)	0.996^{+}(±0.00)	0.932^{+}(±0.03)	0.995 (±0.00)	0.920^{--} (±0.02)
10	Local	0.995 (±0.00)	0.934 (±0.02)	0.995 (±0.00)	0.932 (±0.01)	0.846 (±0.00)	0.672 (±0.01)	0.992 (±0.00)	0.926 (±0.01)
	Federated	**0.998^{+}(±0.00)**	**0.951 (±0.03)**	**0.998 (±0.00)**	**0.963 (±0.02)**	**0.998^{+}(±0.00)**	**0.948^{+}(±0.03)**	**0.998^{++}(±0.00)**	**0.965^{+}(±0.01)**
	Synthesis	0.989^{-} (±0.00)	0.804^{-} (±0.01)	0.990^{-} (±0.00)	0.815^{-} (±0.01)	0.992^{+}(±0.00)	0.864^{+}(±0.06)	0.992 (±0.00)	0.847^{-} (±0.02)
20	Local	0.987 (±0.00)	0.885 (±0.01)	0.986 (±0.00)	0.884 (±0.01)	0.844 (±0.00)	0.645 (±0.02)	0.986 (±0.00)	0.880 (±0.02)
	Federated	**0.997^{+}(±0.00)**	**0.935^{+}(±0.03)**	**0.997^{+}(±0.00)**	**0.956^{+}(±0.00)**	**0.997^{+}(±0.00)**	**0.943^{+}(±0.02)**	**0.997^{+}(±0.00)**	**0.941^{+}(±0.02)**
	Synthesis	0.988 (±0.00)	0.783^{-} (±0.06)	0.990 (±0.00)	0.809 (±0.05)	0.991^{+}(±0.00)	0.839^{+}(±0.07)	0.988 (±0.00)	0.766^{-} (±0.06)
30	Local	0.976 (±0.01)	0.840 (±0.01)	0.978 (±0.01)	0.852 (±0.00)	0.841 (±0.00)	0.629 (±0.00)	0.983 (±0.00)	0.848 (±0.01)
	Federated	**0.995^{+}(±0.00)**	**0.915^{+}(±0.02)**	**0.996^{++}(±0.00)**	**0.942^{+}(±0.00)**	**0.996^{+}(±0.00)**	**0.940^{+}(±0.01)**	**0.995^{+}(±0.00)**	**0.907^{+}(±0.02)**
	Synthesis	0.978 (±0.01)	0.705 (±0.17)	0.984 (±0.00)	0.773^{-} (±0.00)	0.991^{+}(±0.00)	0.845^{+}(±0.04)	0.986 (±0.01)	0.775 (±0.08)
40	Local	0.974 (±0.01)	0.818 (±0.02)	0.971 (±0.00)	0.815 (±0.01)	0.836 (±0.00)	0.609 (±0.00)	0.977 (±0.00)	0.823 (±0.02)
	Federated	**0.994^{+}(±0.00)**	**0.909^{+}(±0.00)**	**0.993^{+}(±0.00)**	**0.911^{+}(±0.00)**	**0.995^{+}(±0.00)**	**0.908^{+}(±0.02)**	**0.994^{+}(±0.00)**	**0.918^{+}(±0.01)**
	Synthesis	0.950^{-} (±0.01)	0.503^{-} (±0.04)	0.968 (±0.01)	0.612^{--} (±0.09)	0.989^{+}(±0.00)	0.806^{+}(±0.01)	0.974 (±0.01)	0.681^{-} (±0.01)
50	Local	0.972 (±0.01)	0.804 (±0.01)	0.969 (±0.00)	0.798 (±0.00)	0.838 (±0.00)	0.601 (±0.01)	0.972 (±0.01)	0.807 (±0.01)
	Federated	**0.993^{+}(±0.00)**	**0.899^{+}(±0.01)**	**0.993^{+}(±0.00)**	**0.898^{+}(±0.00)**	**0.994^{+}(±0.00)**	**0.905^{+}(±0.02)**	**0.993^{+}(±0.00)**	**0.893^{+}(±0.01)**
	Synthesis	0.935^{-} (±0.01)	0.455^{-} (±0.06)	0.957 (±0.01)	0.574^{-} (±0.07)	0.978^{+}(±0.01)	0.704^{+}(±0.06)	0.965 (±0.02)	0.667^{--} (±0.09)

Table A.13: Results for DevNet using Credit Card data set: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		**Feature-based non-i.i.d.**	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.953 (±0.01)	0.683 (±0.07)	-	-
2	Local	0.955 (±0.03)	0.662 (±0.05)	0.966 (±0.01)	0.687 (±0.03)
	Federated	**0.985 (±0.00)**	**0.708 (±0.05)**	**0.975 (±0.01)**	**0.707 (±0.06)**
	Synthesis	0.908 (±0.06)	0.487 (±0.18)	0.962 (±0.01)	0.597$^-$ (±0.03)
5	Local	0.952 (±0.02)	0.660 (±0.06)	0.959 (±0.02)	0.636 (±0.07)
	Federated	**0.984^{++}(±0.01)**	**0.711 (±0.05)**	**0.980 (±0.01)**	**0.712 (±0.06)**
	Synthesis	0.947 (±0.01)	0.409 (±0.21)	0.851 (±0.11)	0.295 (±0.28)
10	Local	0.959 (±0.02)	0.648 (±0.06)	0.957 (±0.01)	0.659 (±0.03)
	Federated	**0.984^{++}(±0.01)**	**0.707 (±0.05)**	**0.981^{+}(±0.01)**	**0.712 (±0.06)**
	Synthesis	0.958 (±0.01)	0.645 (±0.11)	0.915 (±0.08)	0.540 (±0.13)
20	Local	0.959 (±0.01)	0.635 (±0.06)	0.959 (±0.01)	0.622 (±0.04)
	Federated	**0.985^{+}(±0.00)**	**0.708 (±0.06)**	**0.978^{+}(±0.00)**	**0.713^{++}(±0.05)**
	Synthesis	0.924$^-$ (±0.01)	0.417 (±0.27)	0.959 (±0.01)	0.690 (±0.07)
30	Local	0.958 (±0.01)	0.626 (±0.05)	0.956 (±0.01)	0.638 (±0.05)
	Federated	**0.984^{+}(±0.00)**	**0.708 (±0.05)**	**0.978^{+}(±0.01)**	**0.711 (±0.06)**
	Synthesis	0.939 (±0.02)	0.620 (±0.03)	0.953 (±0.01)	0.503$^-$ (±0.06)
40	Local	0.958 (±0.01)	0.641 (±0.06)	0.954 (±0.01)	0.623 (±0.07)
	Federated	**0.984^{+}(±0.00)**	**0.711 (±0.05)**	**0.974^{+}(±0.01)**	**0.708 (±0.06)**
	Synthesis	0.945 (±0.02)	0.334$^-$ (±0.02)	0.918$^-$ (±0.01)	0.365 (±0.21)
50	Local	0.955 (±0.01)	0.633 (±0.05)	0.955 (±0.01)	0.635 (±0.06)
	Federated	**0.986^{+}(±0.00)**	**0.711 (±0.06)**	**0.976^{+}(±0.00)**	**0.710 (±0.05)**
	Synthesis	0.907 (±0.07)	0.394 (±0.32)	0.873 (±0.08)	0.185$^-$ (±0.18)

Table A.14: Results for DevNet using Probe data set: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		**Feature-based non-i.i.d.**	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.933 (±0.09)	0.877 (±0.12)	-	-
2	Local	0.955 (±0.04)	0.892 (±0.05)	0.925 (±0.08)	0.862 (±0.08)
	Federated	**0.993 (±0.00)**	**0.957^{++}(±0.01)**	**0.993 (±0.00)**	**0.959 (±0.00)**
	Synthesis	0.925 (±0.10)	0.879 (±0.11)	0.964 (±0.02)	0.912 (±0.01)
5	Local	0.915 (±0.07)	0.838 (±0.06)	0.941 (±0.04)	0.869 (±0.06)
	Federated	**0.994 (±0.00)**	**0.963^{+}(±0.00)**	**0.994^{++}(±0.00)**	**0.966^{+}(±0.00)**
	Synthesis	0.988 (±0.00)	0.942^{++}(±0.02)	0.954 (±0.03)	0.872 (±0.06)
10	Local	0.933 (±0.03)	0.859 (±0.04)	0.933 (±0.02)	0.863 (±0.02)
	Federated	**0.994^{+}(±0.00)**	**0.962^{+}(±0.00)**	**0.993^{+}(±0.00)**	**0.960^{+}(±0.00)**
	Synthesis	0.975 (±0.01)	0.923^{++}(±0.01)	0.932 (±0.09)	0.858 (±0.12)
20	Local	0.953 (±0.01)	0.884 (±0.02)	0.942 (±0.01)	0.872 (±0.02)
	Federated	**0.994^{+}(±0.00)**	**0.966^{+}(±0.00)**	**0.995^{+}(±0.00)**	**0.967^{+}(±0.00)**
	Synthesis	0.943 (±0.03)	0.859 (±0.07)	0.929 (±0.08)	0.817 (±0.11)
30	Local	0.930 (±0.00)	0.865 (±0.01)	0.942 (±0.01)	0.879 (±0.01)
	Federated	**0.994^{+}(±0.00)**	**0.962^{+}(±0.01)**	**0.994^{+}(±0.00)**	**0.962^{+}(±0.00)**
	Synthesis	0.971^{+}(±0.01)	0.886 (±0.04)	0.964 (±0.03)	0.850 (±0.10)
40	Local	0.944 (±0.01)	0.882 (±0.01)	0.929 (±0.02)	0.859 (±0.03)
	Federated	**0.994^{+}(±0.00)**	**0.966^{+}(±0.00)**	**0.995^{+}(±0.00)**	**0.967^{+}(±0.00)**
	Synthesis	0.967^{++}(±0.01)	0.861 (±0.06)	0.961^{++}(±0.01)	0.837 (±0.05)
50	Local	0.948 (±0.01)	0.885 (±0.02)	0.923 (±0.01)	0.857 (±0.01)
	Federated	**0.994^{+}(±0.00)**	**0.962^{+}(±0.01)**	**0.994^{+}(±0.00)**	**0.961^{+}(±0.01)**
	Synthesis	0.919 (±0.04)	0.666^{-}(±0.04)	0.970^{+}(±0.01)	0.833 (±0.06)

Table A.15: Results for DevNet using R2L data set: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		**Feature-based non-i.i.d.**	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.805 (±0.01)	0.444 (±0.02)	-	-
2	Local	0.685 (±0.09)	0.318 (±0.21)	0.698 (±0.02)	0.223 (±0.03)
	Federated	**0.939^{+}(±0.03)**	**0.436 (±0.13)**	**0.964^{+}(±0.00)**	0.451^{+}(±0.03)
	Synthesis	0.796 (±0.05)	0.357 (±0.03)	0.822^{+}(±0.05)	**0.464^{+}(±0.02)**
5	Local	0.709 (±0.08)	0.249 (±0.15)	0.761 (±0.06)	0.273 (±0.12)
	Federated	**0.913^{+}(±0.02)**	0.288 (±0.05)	**0.967^{+}(±0.01)**	**0.476^{++}(±0.04)**
	Synthesis	0.726 (±0.09)	**0.323 (±0.01)**	0.810 (±0.10)	0.371 (±0.04)
10	Local	0.756 (±0.05)	0.268 (±0.08)	0.755 (±0.09)	0.284 (±0.16)
	Federated	**0.930^{+}(±0.03)**	**0.354 (±0.10)**	**0.965^{+}(±0.01)**	**0.442 (±0.03)**
	Synthesis	0.723 (±0.08)	0.293 (±0.11)	0.782 (±0.02)	0.221 (±0.01)
20	Local	0.762 (±0.08)	0.280 (±0.15)	0.762 (±0.05)	0.278 (±0.13)
	Federated	**0.915^{+}(±0.01)**	**0.336 (±0.05)**	**0.968 (±0.00)**	**0.503 (±0.00)**
	Synthesis	0.712 (±0.08)	0.220 (±0.09)	0.750 (±0.08)	0.282 (±0.16)
30	Local	0.755 (±0.03)	0.269 (±0.12)	0.758 (±0.04)	0.265 (±0.12)
	Federated	**0.921^{+}(±0.02)**	**0.297 (±0.03)**	**0.970 (±0.00)**	**0.453 (±0.00)**
	Synthesis	0.623^{-}(±0.01)	0.233 (±0.05)	0.630 (±0.13)	0.252 (±0.14)
40	Local	0.778 (±0.04)	0.290 (±0.12)	0.752 (±0.04)	0.265 (±0.13)
	Federated	**0.921^{+}(±0.02)**	**0.306 (±0.07)**	**0.963 (±0.00)**	**0.478 (±0.00)**
	Synthesis	0.655^{-}(±0.06)	0.154 (±0.05)	0.698 (±0.13)	0.234 (±0.06)
50	Local	0.762 (±0.05)	0.270 (±0.13)	0.762 (±0.05)	0.270 (±0.12)
	Federated	**0.926^{+}(±0.00)**	**0.290 (±0.01)**	**0.965 (±0.00)**	**0.422 (±0.00)**
	Synthesis	0.703 (±0.04)	0.169 (±0.09)	0.519^{-}(±0.11)	0.081^{--}(±0.02)

Table A.16: Results for DevNet using Annthyroid data set: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		**Feature-based non-i.i.d.**	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.688 (±0.13)	0.220 (±0.19)	-	-
2	Local	0.754 (±0.03)	0.271 (±0.02)	0.741 (±0.06)	0.300 (±0.09)
	Federated	**0.780 (±0.07)**	**0.455^{+}(±0.06)**	0.789 (±0.07)	**0.472^{++}(±0.10)**
	Synthesis	0.741 (±0.02)	0.243 (±0.07)	**0.819 (±0.09)**	0.400 (±0.10)
5	Local	0.759 (±0.00)	0.316 (±0.01)	0.737 (±0.01)	0.287 (±0.04)
	Federated	0.747 (±0.05)	**0.462^{+}(±0.02)**	**0.809^{+}(±0.01)**	**0.488^{+}(±0.01)**
	Synthesis	**0.762 (±0.00)**	0.274^{-} (±0.03)	0.773^{++}(±0.02)	0.337 (±0.04)
10	Local	**0.744 (±0.01)**	0.300 (±0.03)	0.742 (±0.04)	0.294 (±0.07)
	Federated	0.724 (±0.04)	**0.444^{+}(±0.04)**	**0.759 (±0.02)**	**0.478^{+}(±0.03)**
	Synthesis	0.738 (±0.04)	0.279 (±0.05)	0.722 (±0.06)	0.244 (±0.11)
20	Local	0.740 (±0.00)	0.301 (±0.02)	0.734 (±0.03)	0.286 (±0.05)
	Federated	**0.763 (±0.09)**	**0.463^{+}(±0.04)**	**0.784 (±0.05)**	**0.455^{+}(±0.03)**
	Synthesis	0.684^{-} (±0.03)	0.247^{-} (±0.02)	0.691 (±0.15)	0.254 (±0.14)
30	Local	0.729 (±0.02)	0.291 (±0.04)	0.736 (±0.01)	0.283 (±0.01)
	Federated	**0.737 (±0.06)**	**0.449^{+}(±0.04)**	**0.752 (±0.07)**	**0.454^{+}(±0.02)**
	Synthesis	0.705 (±0.06)	0.257 (±0.09)	0.684 (±0.06)	0.205 (±0.13)
40	Local	**0.731 (±0.01)**	0.271 (±0.04)	0.723 (±0.02)	0.273 (±0.04)
	Federated	0.729 (±0.03)	**0.462^{+}(±0.03)**	**0.738 (±0.05)**	**0.455^{+}(±0.02)**
	Synthesis	0.685 (±0.07)	0.231 (±0.09)	0.685 (±0.21)	0.206 (±0.18)
50	Local	0.723 (±0.02)	0.273 (±0.05)	0.721 (±0.01)	0.268 (±0.03)
	Federated	**0.747 (±0.07)**	**0.428^{+}(±0.02)**	**0.783^{+}(±0.02)**	**0.439^{+}(±0.04)**
	Synthesis	0.683 (±0.10)	0.252 (±0.13)	0.648 (±0.15)	0.185 (±0.11)

Table A.17: Results for Autoencoder using Credit Card data set: Significance of federated learning and synthetic-based learning results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		Feature-based non-i.i.d.	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.938 (±0.01)	0.459 (±0.11)	-	-
2	Local	**0.950 (±0.02)**	**0.407 (±0.07)**	**0.941 (±0.01)**	**0.378 (±0.06)**
	Federated	0.941 (±0.02)	0.363 (±0.06)	0.937 (±0.01)	0.363 (±0.05)
	Synthesis	0.922 (±0.02)	0.081$^-$ (±0.03)	0.909 (±0.03)	0.062$^-$ (±0.04)
5	Local	**0.944 (±0.02)**	**0.375 (±0.06)**	0.940 (±0.02)	0.361 (±0.06)
	Federated	0.941 (±0.02)	0.363 (±0.06)	0.941 (±0.01)	**0.366 (±0.06)**
	Synthesis	0.937 (±0.01)	0.092$^-$ (±0.01)	**0.944 (±0.01)**	0.082$^-$ (±0.02)
10	Local	0.941 (±0.02)	0.361 (±0.06)	0.939 (±0.02)	0.358 (±0.06)
	Federated	**0.941 (±0.02)**	**0.362 (±0.06)**	0.941 (±0.02)	**0.364 (±0.06)**
	Synthesis	0.933 (±0.01)	0.068$^-$ (±0.02)	**0.943 (±0.01)**	0.098$^-$ (±0.05)
20	Local	0.941 (±0.02)	0.359 (±0.06)	0.939 (±0.02)	0.357 (±0.06)
	Federated	0.941 (±0.02)	**0.362 (±0.06)**	**0.941 (±0.02)**	**0.366 (±0.06)**
	Synthesis	**0.943 (±0.02)**	0.229 (±0.18)	0.936 (±0.01)	0.129$^-$ (±0.05)
30	Local	0.941 (±0.02)	0.360 (±0.06)	0.940 (±0.02)	0.359 (±0.06)
	Federated	0.941 (±0.02)	**0.360 (±0.06)**	0.941 (±0.02)	**0.362 (±0.06)**
	Synthesis	**0.951 (±0.02)**	0.184$^-$ (±0.07)	**0.945 (±0.02)**	0.239 (±0.19)
40	Local	0.941 (±0.02)	0.361 (±0.06)	0.938 (±0.02)	0.359 (±0.06)
	Federated	0.941 (±0.02)	0.361 (±0.06)	0.940 (±0.02)	0.356 (±0.06)
	Synthesis	**0.953 (±0.01)**	**0.377 (±0.06)**	**0.949 (±0.02)**	**0.362 (±0.01)**
50	Local	0.941 (±0.02)	0.360 (±0.06)	0.938 (±0.02)	0.357 (±0.06)
	Federated	0.941 (±0.02)	0.361 (±0.06)	0.940 (±0.02)	0.359 (±0.06)
	Synthesis	**0.950 (±0.02)**	**0.399 (±0.06)**	**0.951 (±0.02)**	**0.367 (±0.05)**

Table A.18: Results for Autoencoder using Probe data set: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		Feature-based non-i.i.d.	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.961 (±0.00)	0.826 (±0.01)	-	-
2	Local	**0.968 (±0.00)**	**0.840 (±0.00)**	**0.974 (±0.01)**	**0.865 (±0.03)**
	Federated	0.966 (±0.00)	0.834 (±0.01)	0.944 (±0.03)	0.791^{--} (±0.05)
	Synthesis	0.805^{-} (±0.03)	0.374^{-} (±0.03)	0.844^{-} (±0.07)	0.469^{-} (±0.06)
5	Local	**0.969 (±0.00)**	**0.834 (±0.00)**	**0.978 (±0.00)**	**0.872 (±0.01)**
	Federated	0.962^{-} (±0.00)	0.817^{--} (±0.01)	0.954^{-} (±0.01)	0.793^{-} (±0.02)
	Synthesis	0.761^{-} (±0.05)	0.345^{-} (±0.04)	0.862^{-} (±0.04)	0.461^{-} (±0.03)
10	Local	**0.974 (±0.00)**	**0.850 (±0.01)**	**0.970 (±0.00)**	0.833 (±0.00)
	Federated	0.970 (±0.00)	0.841 (±0.02)	0.967 (±0.01)	**0.834 (±0.05)**
	Synthesis	0.832^{-} (±0.03)	0.395^{-} (±0.03)	0.901^{-} (±0.01)	0.502^{-} (±0.02)
20	Local	**0.973 (±0.00)**	0.845 (±0.01)	0.954 (±0.00)	0.731 (±0.02)
	Federated	0.969^{-} (±0.00)	**0.846 (±0.01)**	**0.968^{++}(±0.01)**	**0.844^{+}(±0.03)**
	Synthesis	0.696^{-} (±0.01)	0.310^{-} (±0.01)	0.856^{-} (±0.01)	0.470^{-} (±0.02)
30	Local	0.960 (±0.00)	0.781 (±0.01)	0.950 (±0.00)	0.709 (±0.02)
	Federated	**0.962 (±0.00)**	**0.795 (±0.00)**	**0.967 (±0.02)**	**0.830^{+}(±0.04)**
	Synthesis	0.604^{-} (±0.05)	0.271^{-} (±0.02)	0.713^{-} (±0.10)	0.327^{-} (±0.08)
40	Local	0.954 (±0.00)	0.719 (±0.01)	0.947 (±0.00)	0.690 (±0.02)
	Federated	**0.961^{+}(±0.00)**	**0.799^{+}(±0.01)**	**0.964 (±0.02)**	**0.824^{+}(±0.05)**
	Synthesis	0.612^{-} (±0.05)	0.271^{-} (±0.02)	0.863^{-} (±0.03)	0.451^{-} (±0.04)
50	Local	0.955 (±0.00)	0.729 (±0.01)	0.924 (±0.01)	0.651 (±0.03)
	Federated	**0.969^{+}(±0.01)**	**0.822^{+}(±0.01)**	**0.964^{+}(±0.02)**	**0.821^{+}(±0.05)**
	Synthesis	0.827^{-} (±0.02)	0.432^{-} (±0.02)	0.891 (±0.03)	0.520^{-} (±0.04)

Table A.19: Results for Autoencoder using R2L data set: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		**Feature-based non-i.i.d.**	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.928 (±0.01)	0.454 (±0.03)	-	-
2	Local	0.888 (±0.01)	0.264 (±0.03)	**0.907 (±0.01)**	**0.221 (±0.02)**
	Federated	**0.908^{++}(±0.01)**	**0.348^{+}(±0.03)**	0.880^{-}(±0.01)	0.213 (±0.02)
	Synthesis	0.819^{-}(±0.02)	0.125^{-}(±0.02)	0.816^{-}(±0.03)	0.118^{-}(±0.02)
5	Local	**0.895 (±0.01)**	0.225 (±0.02)	0.878 (±0.00)	0.165 (±0.01)
	Federated	0.880^{--}(±0.00)	**0.270^{+}(±0.01)**	**0.906^{+}(±0.01)**	**0.263^{+}(±0.03)**
	Synthesis	0.805^{-}(±0.04)	0.117^{-}(±0.02)	0.811^{-}(±0.02)	0.121^{-}(±0.02)
10	Local	**0.896 (±0.00)**	0.186 (±0.01)	0.822 (±0.00)	0.129 (±0.01)
	Federated	0.874^{-}(±0.01)	**0.231^{+}(±0.00)**	**0.920^{+}(±0.01)**	**0.234^{+}(±0.01)**
	Synthesis	0.785^{-}(±0.03)	0.112^{-}(±0.03)	0.769^{--}(±0.03)	0.097^{-}(±0.01)
20	Local	0.853 (±0.01)	0.147 (±0.01)	0.776 (±0.02)	0.099 (±0.01)
	Federated	**0.908^{+}(±0.01)**	**0.330^{+}(±0.02)**	**0.886^{+}(±0.01)**	**0.169^{+}(±0.02)**
	Synthesis	0.798^{--}(±0.04)	0.111^{--}(±0.02)	0.826^{+}(±0.01)	0.125^{+}(±0.01)
30	Local	0.799 (±0.01)	0.111 (±0.01)	0.770 (±0.01)	0.095 (±0.01)
	Federated	**0.915^{+}(±0.00)**	**0.240^{+}(±0.02)**	**0.864^{+}(±0.01)**	**0.157^{+}(±0.02)**
	Synthesis	0.787 (±0.06)	0.109 (±0.02)	0.775 (±0.05)	0.111 (±0.02)
40	Local	0.784 (±0.01)	0.102 (±0.01)	0.769 (±0.01)	0.096 (±0.01)
	Federated	**0.905^{+}(±0.00)**	**0.194^{+}(±0.01)**	**0.861^{+}(±0.01)**	**0.158^{+}(±0.01)**
	Synthesis	0.775 (±0.06)	0.108 (±0.02)	0.812^{+}(±0.02)	0.120^{+}(±0.01)
50	Local	0.777 (±0.01)	0.098 (±0.01)	0.764 (±0.01)	0.094 (±0.01)
	Federated	**0.897^{+}(±0.01)**	**0.184^{+}(±0.02)**	**0.889^{+}(±0.01)**	**0.313^{+}(±0.06)**
	Synthesis	0.802^{+}(±0.01)	0.117^{+}(±0.01)	0.795 (±0.09)	0.121 (±0.02)

Table A.20: Results for Autoencoder using Annthyroid data set: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		Feature-based non-i.i.d.	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.856 (±0.02)	0.295 (±0.08)	-	-
2	Local	0.663 (±0.02)	0.131 (±0.01)	0.663 (±0.02)	0.131 (±0.01)
	Federated	**0.844^{+}(±0.01)**	**0.302^{+}(±0.02)**	0.679 (±0.06)	0.146 (±0.01)
	Synthesis	0.670 (±0.17)	0.153 (±0.09)	**0.762 (±0.11)**	**0.176 (±0.06)**
5	Local	0.634 (±0.03)	0.114 (±0.01)	0.634 (±0.03)	0.114 (±0.01)
	Federated	0.704^{++}(±0.04)	**0.166^{+}(±0.01)**	**0.702^{+}(±0.02)**	**0.165^{++}(±0.04)**
	Synthesis	**0.734^{+}(±0.03)**	0.141 (±0.03)	0.694 (±0.15)	0.144 (±0.05)
10	Local	0.613 (±0.02)	0.103 (±0.00)	0.613 (±0.02)	0.103 (±0.00)
	Federated	0.717^{+}(±0.02)	**0.181^{+}(±0.02)**	**0.658^{++}(±0.02)**	**0.139^{+}(±0.01)**
	Synthesis	**0.774^{+}(±0.01)**	0.164^{+}(±0.03)	0.642 (±0.12)	0.126 (±0.02)
20	Local	0.584 (±0.02)	0.093 (±0.00)	0.584 (±0.02)	0.093 (±0.00)
	Federated	**0.707^{+}(±0.01)**	**0.172^{+}(±0.02)**	**0.654^{+}(±0.02)**	**0.148^{+}(±0.01)**
	Synthesis	0.528 (±0.11)	0.090 (±0.02)	0.595 (±0.16)	0.120 (±0.04)
30	Local	0.559 (±0.02)	0.086 (±0.00)	0.559 (±0.02)	0.086 (±0.00)
	Federated	0.685^{+}(±0.04)	**0.156^{+}(±0.03)**	**0.659^{+}(±0.04)**	**0.145^{+}(±0.03)**
	Synthesis	**0.720^{+}(±0.10)**	0.147^{++}(±0.05)	0.608 (±0.17)	0.112 (±0.04)
40	Local	0.535 (±0.01)	0.083 (±0.00)	0.535 (±0.01)	0.083 (±0.00)
	Federated	**0.666^{+}(±0.02)**	**0.125^{+}(±0.00)**	0.642^{+}(±0.02)	**0.118^{+}(±0.00)**
	Synthesis	0.631 (±0.13)	0.110 (±0.03)	**0.653^{++}(±0.08)**	0.112^{+}(±0.01)
50	Local	0.534 (±0.01)	0.082 (±0.00)	0.534 (±0.01)	0.082 (±0.00)
	Federated	**0.687^{+}(±0.02)**	**0.162^{+}(±0.02)**	**0.681^{+}(±0.03)**	**0.143^{+}(±0.02)**
	Synthesis	0.672^{+}(±0.08)	0.142^{++}(±0.05)	0.612 (±0.09)	0.104^{++}(±0.02)

Table A.21: Results for Isolation Forest using Credit Card data set: Significance of federated learning and synthetic-based learning results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		**Feature-based non-i.i.d.**		**Label-based non-i.i.d.**		**LDP non-i.i.d.**	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.947 (±0.01)	0.213 (±0.04)	-	-	-	-	-	-
2	Local	0.948 (±0.01)	0.213 (±0.04)	0.949 (±0.01)	0.215 (±0.04)	0.930 (±0.02)	0.221 (±0.04)	0.929 (±0.02)	0.204 (±0.03)
	Federated	0.948 (±0.01)	0.212 (±0.04)	0.949 (±0.01)	0.215 (±0.04)	0.934 (±0.02)	**0.225 (±0.05)**	0.937 (±0.02)	0.208 (±0.03)
	Synthesis	**0.951 (±0.01)**	**0.216 (±0.04)**	**0.950 (±0.01)**	**0.219 (±0.05)**	**0.949 (±0.01)**	0.216 (±0.05)	**0.948 (±0.01)**	**0.217 (±0.03)**
5	Local	0.946 (±0.01)	0.210 (±0.04)	0.947 (±0.01)	0.211 (±0.04)	0.926 (±0.01)	0.207 (±0.01)	0.929 (±0.01)	0.193 (±0.03)
	Federated	**0.947 (±0.01)**	0.211 (±0.04)	**0.948 (±0.01)**	0.210 (±0.04)	0.936 (±0.01)	0.214 (±0.03)	0.939 (±0.01)	0.203 (±0.03)
	Synthesis	0.946 (±0.01)	**0.221 (±0.05)**	0.948 (±0.01)	**0.220 (±0.04)**	**0.946 (±0.01)**	**0.216 (±0.04)**	**0.947 (±0.01)**	**0.226 (±0.04)**
10	Local	0.947 (±0.01)	0.215 (±0.05)	0.947 (±0.01)	0.215 (±0.04)	0.928 (±0.02)	0.179 (±0.03)	0.930 (±0.01)	0.189 (±0.04)
	Federated	0.948 (±0.01)	0.214 (±0.05)	**0.948 (±0.01)**	0.214 (±0.05)	0.942 (±0.01)	0.190 (±0.03)	0.940 (±0.01)	0.200 (±0.04)
	Synthesis	**0.949 (±0.01)**	**0.219 (±0.04)**	0.947 (±0.01)	**0.223 (±0.05)**	**0.949 (±0.01)**	**0.221 (±0.03)**	**0.949 (±0.01)**	**0.214 (±0.04)**
20	Local	0.947 (±0.01)	0.216 (±0.04)	0.947 (±0.01)	0.214 (±0.04)	0.927 (±0.01)	0.176 (±0.04)	0.930 (±0.01)	0.187 (±0.04)
	Federated	0.947 (±0.01)	0.214 (±0.04)	0.947 (±0.01)	0.214 (±0.05)	0.937 (±0.01)	0.193 (±0.05)	0.942 (±0.01)	0.205 (±0.04)
	Synthesis	**0.948 (±0.01)**	**0.231 (±0.05)**	**0.950 (±0.01)**	**0.227 (±0.04)**	0.950^{++}(±0.01)	**0.230 (±0.04)**	**0.945 (±0.01)**	**0.233 (±0.05)**
30	Local	0.947 (±0.01)	0.218 (±0.05)	0.946 (±0.01)	0.218 (±0.05)	0.930 (±0.01)	0.168 (±0.03)	0.929 (±0.01)	0.176 (±0.04)
	Federated	0.947 (±0.01)	0.218 (±0.05)	0.947 (±0.01)	0.217 (±0.05)	0.941 (±0.01)	0.186 (±0.05)	0.940 (±0.01)	0.198 (±0.06)
	Synthesis	**0.948 (±0.01)**	**0.231 (±0.05)**	**0.949 (±0.01)**	**0.240 (±0.05)**	**0.948 (±0.01)**	**0.233 (±0.04)**	**0.949 (±0.01)**	**0.242 (±0.04)**
40	Local	0.946 (±0.01)	0.216 (±0.05)	0.946 (±0.01)	0.218 (±0.05)	0.928 (±0.01)	0.166 (±0.04)	0.929 (±0.01)	0.168 (±0.04)
	Federated	0.947 (±0.01)	0.215 (±0.05)	0.947 (±0.01)	0.220 (±0.05)	0.939 (±0.01)	0.182 (±0.05)	0.942 (±0.01)	0.183 (±0.05)
	Synthesis	**0.949 (±0.01)**	**0.240 (±0.04)**	**0.951 (±0.01)**	**0.242 (±0.05)**	**0.948 (±0.01)**	**0.237 (±0.05)**	0.950^{++}(±0.01)	0.240^{++}(±0.04)
50	Local	0.947 (±0.01)	0.216 (±0.05)	0.946 (±0.01)	0.219 (±0.05)	0.929 (±0.01)	0.155 (±0.04)	0.930 (±0.01)	0.170 (±0.04)
	Federated	0.947 (±0.01)	0.216 (±0.05)	0.947 (±0.01)	0.220 (±0.05)	0.941 (±0.01)	0.170 (±0.06)	0.941 (±0.01)	0.190 (±0.06)
	Synthesis	**0.949 (±0.01)**	**0.255 (±0.06)**	**0.948 (±0.01)**	**0.245 (±0.04)**	0.948^{++}(±0.01)	0.239^{++}(±0.04)	**0.946 (±0.01)**	0.241^{++}(±0.04)

Table A.22: Results for Isolation Forest using Probe data set: Significance of federated learning and synthetic-based learning results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		Feature-based non-i.i.d.		Label-based non-i.i.d.		LDP non-i.i.d.	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.885 (±0.02)	0.443 (±0.05)	-	-	-	-	-	-
2	Local	0.887 (±0.02)	0.446 (±0.03)	0.882 (±0.02)	0.442 (±0.04)	0.774 (±0.08)	0.458 (±0.08)	0.809 (±0.06)	0.386 (±0.09)
	Federated	0.888 (±0.02)	0.447 (±0.04)	0.886 (±0.02)	0.445 (±0.03)	0.883 (±0.04)	**0.518 (±0.12)**	0.824 (±0.05)	0.394 (±0.09)
	Synthesis	**0.895 (±0.01)**	**0.469 (±0.02)**	**0.903 (±0.01)**	**0.484 (±0.03)**	**0.896^{++}(±0.01)**	0.473 (±0.03)	**0.897^{++}(±0.02)**	**0.474 (±0.03)**
5	Local	0.889 (±0.02)	0.460 (±0.04)	0.884 (±0.01)	0.453 (±0.03)	0.818 (±0.05)	0.457 (±0.08)	0.832 (±0.03)	0.413 (±0.07)
	Federated	0.891 (±0.02)	0.462 (±0.04)	0.890 (±0.02)	0.458 (±0.03)	0.885^{++}(±0.03)	**0.508 (±0.10)**	0.855 (±0.03)	0.433 (±0.08)
	Synthesis	**0.899 (±0.01)**	**0.480 (±0.02)**	**0.903 (±0.01)**	**0.490 (±0.03)**	**0.900^{+}(±0.01)**	0.482 (±0.03)	**0.897^{+}(±0.01)**	**0.474 (±0.03)**
10	Local	0.888 (±0.01)	0.461 (±0.02)	0.885 (±0.01)	0.463 (±0.03)	0.833 (±0.01)	0.455 (±0.04)	0.822 (±0.02)	0.413 (±0.06)
	Federated	0.891 (±0.01)	0.463 (±0.03)	0.892 (±0.02)	0.467 (±0.04)	0.878^{+}(±0.02)	**0.495 (±0.07)**	0.852 (±0.02)	0.437 (±0.07)
	Synthesis	**0.892 (±0.01)**	**0.472 (±0.03)**	**0.898 (±0.01)**	**0.478 (±0.02)**	**0.898^{+}(±0.02)**	0.485 (±0.04)	**0.899^{+}(±0.01)**	**0.483 (±0.02)**
20	Local	0.891 (±0.01)	0.475 (±0.02)	0.884 (±0.01)	0.474 (±0.02)	0.823 (±0.02)	0.446 (±0.07)	0.833 (±0.02)	0.419 (±0.06)
	Federated	0.893 (±0.01)	0.476 (±0.02)	0.894 (±0.01)	0.480 (±0.03)	0.871^{++}(±0.02)	**0.481 (±0.09)**	0.853 (±0.01)	0.443 (±0.07)
	Synthesis	**0.900 (±0.01)**	**0.498 (±0.02)**	**0.909^{+}(±0.01)**	**0.517^{++}(±0.02)**	**0.893^{+}(±0.00)**	0.476 (±0.01)	**0.901^{+}(±0.01)**	**0.494 (±0.02)**
30	Local	0.895 (±0.01)	0.491 (±0.02)	0.886 (±0.01)	0.486 (±0.03)	0.837 (±0.02)	0.456 (±0.06)	0.832 (±0.03)	0.429 (±0.08)
	Federated	**0.897 (±0.01)**	**0.493 (±0.03)**	0.897 (±0.01)	0.492 (±0.03)	0.876^{+}(±0.01)	0.487 (±0.08)	0.853 (±0.02)	0.450 (±0.09)
	Synthesis	0.895 (±0.01)	0.483 (±0.01)	**0.902 (±0.01)**	**0.502 (±0.02)**	**0.903^{+}(±0.01)**	**0.496 (±0.02)**	**0.906^{+}(±0.02)**	**0.504 (±0.04)**
40	Local	0.895 (±0.01)	0.496 (±0.03)	0.889 (±0.01)	0.494 (±0.02)	0.830 (±0.03)	0.453 (±0.07)	0.838 (±0.02)	0.441 (±0.08)
	Federated	0.897 (±0.01)	**0.498 (±0.03)**	0.898 (±0.01)	0.501 (±0.03)	0.870 (±0.01)	0.481 (±0.09)	0.857 (±0.01)	0.463 (±0.09)
	Synthesis	**0.900 (±0.01)**	0.492 (±0.02)	**0.906^{+}(±0.01)**	**0.506 (±0.02)**	**0.903^{+}(±0.01)**	**0.498 (±0.03)**	**0.903^{+}(±0.01)**	**0.504 (±0.02)**
50	Local	0.896 (±0.01)	0.504 (±0.03)	0.891 (±0.01)	0.502 (±0.03)	0.838 (±0.02)	0.463 (±0.07)	0.836 (±0.03)	0.445 (±0.09)
	Federated	0.899 (±0.01)	**0.507 (±0.03)**	0.901 (±0.01)	**0.512 (±0.03)**	0.873^{++}(±0.01)	0.490 (±0.09)	0.857 (±0.01)	0.467 (±0.09)
	Synthesis	**0.900 (±0.01)**	0.490 (±0.02)	**0.903 (±0.01)**	0.499 (±0.02)	**0.904^{+}(±0.01)**	**0.501 (±0.01)**	**0.904^{+}(±0.01)**	**0.503 (±0.02)**

Table A.23: Results for Isolation Forest using R2L data set: Significance of federated learning and synthetic-based learning results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		Feature-based non-i.i.d.		Label-based non-i.i.d.		LDP non-i.i.d.	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.818 (±0.00)	0.112 (±0.00)	-	-	-	-	-	-
2	Local	0.815 (±0.02)	0.111 (±0.01)	0.813 (±0.01)	0.110 (±0.01)	0.778 (±0.06)	**0.119 (±0.03)**	0.778 (±0.05)	0.105 (±0.02)
	Federated	0.815 (±0.01)	0.111 (±0.01)	0.815 (±0.01)	0.110 (±0.01)	0.794 (±0.07)	0.110 (±0.03)	0.789 (±0.05)	0.105 (±0.03)
	Synthesis	**0.831 (±0.00)**	**0.120 (±0.00)**	0.831^{+}(±0.00)	**0.120 (±0.00)**	**0.815 (±0.00)**	0.109 (±0.00)	**0.830 (±0.01)**	**0.119 (±0.01)**
5	Local	0.816 (±0.01)	0.111 (±0.01)	0.813 (±0.01)	0.110 (±0.01)	0.799 (±0.04)	**0.128 (±0.04)**	0.791 (±0.02)	0.106 (±0.02)
	Federated	0.817 (±0.00)	0.111 (±0.01)	0.819 (±0.01)	0.112 (±0.01)	0.822 (±0.04)	0.121 (±0.03)	0.805 (±0.02)	0.107 (±0.02)
	Synthesis	**0.833^{+}(±0.01)**	**0.121 (±0.01)**	**0.835^{+}(±0.00)**	**0.122^{++}(±0.00)**	**0.834 (±0.00)**	0.120 (±0.00)	**0.830^{+}(±0.00)**	**0.119 (±0.00)**
10	Local	0.818 (±0.01)	0.112 (±0.01)	0.813 (±0.01)	0.110 (±0.01)	0.785 (±0.02)	0.109 (±0.01)	0.774 (±0.01)	0.099 (±0.01)
	Federated	0.819 (±0.01)	0.112 (±0.01)	0.818 (±0.01)	0.111 (±0.01)	0.807 (±0.02)	0.108 (±0.01)	0.792 (±0.01)	0.101 (±0.01)
	Synthesis	**0.831 (±0.01)**	**0.121 (±0.01)**	**0.832^{+}(±0.00)**	**0.120^{++}(±0.00)**	**0.832 (±0.00)**	**0.118 (±0.00)**	**0.834^{+}(±0.00)**	**0.121^{+}(±0.00)**
20	Local	0.815 (±0.01)	0.110 (±0.01)	0.808 (±0.01)	0.108 (±0.01)	0.780 (±0.01)	0.108 (±0.01)	0.788 (±0.02)	0.107 (±0.01)
	Federated	0.816 (±0.01)	0.110 (±0.01)	0.816 (±0.01)	0.110 (±0.01)	0.803^{+}(±0.01)	0.107 (±0.01)	0.805 (±0.02)	0.108 (±0.01)
	Synthesis	**0.835^{+}(±0.00)**	**0.122^{++}(±0.00)**	**0.837^{+}(±0.00)**	**0.123^{+}(±0.00)**	**0.833 (±0.00)**	**0.119 (±0.00)**	**0.832^{+}(±0.00)**	**0.120 (±0.00)**
30	Local	0.814 (±0.01)	0.109 (±0.01)	0.808 (±0.01)	0.107 (±0.01)	0.783 (±0.02)	0.108 (±0.01)	0.791 (±0.00)	0.109 (±0.01)
	Federated	0.816 (±0.01)	0.110 (±0.01)	0.816 (±0.01)	0.110 (±0.01)	0.804 (±0.02)	0.108 (±0.01)	0.806^{+}(±0.01)	0.109 (±0.01)
	Synthesis	**0.840^{+}(±0.00)**	**0.125^{+}(±0.01)**	**0.842^{+}(±0.00)**	**0.126^{+}(±0.00)**	**0.851 (±0.00)**	**0.130 (±0.00)**	**0.843^{+}(±0.00)**	**0.127^{+}(±0.01)**
40	Local	0.813 (±0.01)	0.109 (±0.01)	0.808 (±0.01)	0.107 (±0.01)	0.771 (±0.00)	0.102 (±0.01)	0.780 (±0.01)	0.104 (±0.01)
	Federated	0.815 (±0.01)	0.109 (±0.01)	0.816 (±0.01)	0.110 (±0.01)	0.791^{+}(±0.01)	0.100 (±0.01)	0.798 (±0.01)	0.104 (±0.01)
	Synthesis	**0.841^{+}(±0.00)**	**0.126^{+}(±0.00)**	**0.840^{+}(±0.01)**	**0.125^{+}(±0.01)**	**0.837 (±0.00)**	**0.121 (±0.00)**	**0.843^{+}(±0.00)**	**0.127^{+}(±0.00)**
50	Local	0.812 (±0.01)	0.108 (±0.01)	0.806 (±0.01)	0.106 (±0.01)	0.780 (±0.00)	0.107 (±0.01)	0.783 (±0.01)	0.106 (±0.00)
	Federated	0.815 (±0.01)	0.109 (±0.01)	0.815 (±0.01)	0.109 (±0.01)	0.799^{+}(±0.01)	0.104 (±0.00)	0.799^{++}(±0.01)	0.105 (±0.01)
	Synthesis	**0.838^{+}(±0.01)**	**0.124^{+}(±0.00)**	**0.840^{+}(±0.00)**	**0.125^{+}(±0.01)**	**0.837 (±0.00)**	**0.121 (±0.00)**	**0.845^{+}(±0.00)**	**0.129^{+}(±0.00)**

Table A.24: Results for Isolation Forest using Annthyroid data set: Significance of federated learning and synthetic-based learning results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		Feature-based non-i.i.d.		Label-based non-i.i.d.		LDP non-i.i.d.	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.789 (±0.02)	0.291 (±0.02)	-	-	-	-	-	-
2	Local	0.787 (±0.01)	0.283 (±0.03)	0.787 (±0.02)	0.284 (±0.03)	0.700 (±0.12)	0.219 (±0.10)	0.648 (±0.11)	0.169 (±0.07)
	Federated	0.790 (±0.01)	0.283 (±0.03)	0.790 (±0.02)	**0.284 (±0.03)**	0.720 (±0.14)	0.228 (±0.10)	0.659 (±0.13)	0.178 (±0.08)
	Synthesis	**0.800 (±0.02)**	**0.284 (±0.04)**	**0.793 (±0.01)**	0.283 (±0.03)	**0.793 (±0.02)**	**0.294 (±0.03)**	**0.787 (±0.02)**	**0.287^{++}(±0.04)**
5	Local	0.789 (±0.01)	0.279 (±0.03)	0.790 (±0.01)	0.277 (±0.03)	0.673 (±0.13)	0.194 (±0.09)	0.653 (±0.15)	0.181 (±0.09)
	Federated	0.794 (±0.01)	0.280 (±0.03)	**0.796 (±0.01)**	0.281 (±0.03)	0.705 (±0.16)	0.208 (±0.10)	0.671 (±0.19)	0.193 (±0.10)
	Synthesis	**0.802 (±0.02)**	**0.288 (±0.04)**	0.789 (±0.01)	**0.290 (±0.03)**	**0.798 (±0.01)**	**0.297 (±0.04)**	**0.786 (±0.01)**	**0.279 (±0.02)**
10	Local	0.783 (±0.00)	0.267 (±0.03)	0.782 (±0.01)	0.270 (±0.03)	0.682 (±0.14)	0.198 (±0.09)	0.681 (±0.14)	0.192 (±0.10)
	Federated	**0.790^{+}(±0.00)**	0.271 (±0.03)	**0.789 (±0.01)**	0.272 (±0.04)	0.722 (±0.17)	0.219 (±0.11)	0.714 (±0.19)	0.215 (±0.12)
	Synthesis	0.787 (±0.02)	**0.284 (±0.04)**	0.786 (±0.01)	**0.281 (±0.04)**	**0.791 (±0.03)**	**0.283 (±0.03)**	**0.796 (±0.01)**	**0.279 (±0.03)**
20	Local	0.778 (±0.00)	0.261 (±0.03)	0.774 (±0.01)	0.258 (±0.03)	0.693 (±0.15)	0.204 (±0.10)	0.702 (±0.15)	0.205 (±0.11)
	Federated	**0.785^{++}(±0.00)**	0.266 (±0.03)	0.782 (±0.01)	0.263 (±0.03)	0.734 (±0.19)	0.232 (±0.12)	0.737 (±0.19)	0.235 (±0.14)
	Synthesis	0.784 (±0.01)	**0.276 (±0.02)**	**0.789 (±0.02)**	**0.285 (±0.03)**	**0.790 (±0.01)**	**0.283 (±0.04)**	**0.800 (±0.02)**	**0.296 (±0.04)**
30	Local	0.778 (±0.01)	0.263 (±0.03)	0.780 (±0.01)	0.264 (±0.03)	0.700 (±0.16)	0.204 (±0.10)	0.706 (±0.16)	0.212 (±0.11)
	Federated	0.785 (±0.01)	0.268 (±0.03)	0.788 (±0.01)	0.269 (±0.03)	0.740 (±0.20)	0.233 (±0.13)	0.740 (±0.20)	0.240 (±0.14)
	Synthesis	**0.790 (±0.02)**	**0.293 (±0.04)**	**0.796 (±0.01)**	**0.288 (±0.03)**	**0.795 (±0.00)**	**0.288 (±0.03)**	**0.791 (±0.01)**	**0.277 (±0.03)**
40	Local	0.779 (±0.01)	0.265 (±0.03)	0.776 (±0.00)	0.261 (±0.03)	0.700 (±0.16)	0.204 (±0.10)	0.709 (±0.16)	0.212 (±0.11)
	Federated	0.787 (±0.01)	0.272 (±0.04)	0.785^{++}(±0.00)	0.269 (±0.04)	0.739 (±0.20)	0.237 (±0.13)	0.743 (±0.20)	0.242 (±0.14)
	Synthesis	**0.792 (±0.02)**	**0.298 (±0.03)**	0.794^{++}(±0.01)	**0.293 (±0.04)**	**0.792 (±0.01)**	**0.286 (±0.05)**	**0.800 (±0.00)**	**0.285 (±0.03)**
50	Local	0.778 (±0.01)	0.263 (±0.03)	0.777 (±0.01)	0.264 (±0.03)	0.693 (±0.16)	0.196 (±0.10)	0.710 (±0.16)	0.212 (±0.11)
	Federated	0.786 (±0.01)	0.273 (±0.03)	0.787 (±0.01)	0.276 (±0.03)	0.731 (±0.20)	0.231 (±0.13)	0.742 (±0.20)	0.245 (±0.14)
	Synthesis	**0.792 (±0.01)**	**0.290 (±0.03)**	0.793^{++}(±0.01)	**0.293 (±0.03)**	**0.792 (±0.01)**	**0.285 (±0.03)**	**0.799 (±0.01)**	**0.296 (±0.03)**

Table A.25: Results for REPEN using Credit Card data set: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		**Feature-based non-i.i.d.**		**Label-based non-i.i.d.**		**LDP non-i.i.d.**	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.946 (±0.02)	0.511 (±0.04)	-	-	-	-	-	-
2	Local	0.929 (±0.02)	0.251 (±0.15)	0.924 (±0.03)	0.184 (±0.08)	0.923 (±0.02)	0.209 (±0.03)	0.932 (±0.02)	0.206 (±0.03)
	Federated	0.932 (±0.01)	0.280 (±0.19)	0.929 (±0.02)	0.201 (±0.02)	0.933 (±0.02)	0.189 (±0.01)	0.932 (±0.02)	0.184 (±0.01)
	Synthesis	**0.944 (±0.01)**	**0.676^{+}(±0.07)**	**0.939 (±0.02)**	**0.664^{+}(±0.07)**	**0.934 (±0.02)**	**0.680^{+}(±0.05)**	**0.938 (±0.04)**	**0.680^{+}(±0.05)**
5	Local	0.929 (±0.02)	0.478 (±0.07)	0.928 (±0.01)	0.334 (±0.04)	0.927 (±0.02)	0.354 (±0.03)	0.921 (±0.02)	0.342 (±0.11)
	Federated	0.935 (±0.02)	0.633^{++}(±0.07)	0.929 (±0.02)	**0.426 (±0.06)**	0.925 (±0.01)	0.468 (±0.10)	0.919 (±0.03)	0.403 (±0.19)
	Synthesis	**0.946 (±0.01)**	**0.676^{+}(±0.06)**	**0.941 (±0.02)**	0.424 (±0.32)	**0.937 (±0.02)**	**0.681^{+}(±0.06)**	**0.938 (±0.02)**	**0.493 (±0.25)**
10	Local	0.929 (±0.02)	0.508 (±0.08)	0.932 (±0.01)	0.485 (±0.01)	0.923 (±0.02)	0.421 (±0.12)	0.932 (±0.01)	0.481 (±0.02)
	Federated	0.939 (±0.02)	0.659^{++}(±0.08)	0.929 (±0.02)	**0.631^{+}(±0.02)**	0.886 (±0.05)	0.459 (±0.08)	0.920 (±0.03)	0.616^{+}(±0.02)
	Synthesis	**0.945 (±0.01)**	**0.680^{+}(±0.06)**	**0.933 (±0.01)**	0.451 (±0.35)	**0.936 (±0.02)**	**0.554 (±0.23)**	**0.936 (±0.03)**	**0.669^{+}(±0.06)**
20	Local	0.929 (±0.02)	0.547 (±0.07)	0.929 (±0.02)	0.553 (±0.02)	0.926 (±0.02)	0.517 (±0.11)	**0.930 (±0.02)**	0.537 (±0.04)
	Federated	0.927 (±0.03)	**0.681^{++}(±0.06)**	0.922 (±0.03)	**0.677^{+}(±0.06)**	0.885 (±0.03)	0.583 (±0.01)	0.916 (±0.02)	0.666^{+}(±0.05)
	Synthesis	**0.950 (±0.02)**	0.600 (±0.07)	**0.937 (±0.02)**	0.659^{+}(±0.05)	**0.934 (±0.04)**	**0.592 (±0.07)**	0.930 (±0.04)	**0.674^{+}(±0.07)**
30	Local	**0.932 (±0.02)**	0.569 (±0.08)	**0.932 (±0.02)**	0.561 (±0.05)	**0.927 (±0.02)**	0.488 (±0.02)	0.928 (±0.02)	0.588 (±0.05)
	Federated	0.923 (±0.03)	**0.681 (±0.06)**	0.917 (±0.03)	**0.676^{++}(±0.06)**	0.872^{--} (±0.03)	**0.607^{+}(±0.04)**	0.915 (±0.02)	**0.675 (±0.05)**
	Synthesis	0.925 (±0.02)	0.482 (±0.27)	0.922 (±0.03)	0.632 (±0.11)	0.924 (±0.04)	0.459 (±0.39)	**0.945 (±0.01)**	0.469 (±0.38)
40	Local	**0.931 (±0.02)**	0.562 (±0.06)	0.931 (±0.02)	0.565 (±0.03)	0.924 (±0.02)	0.475 (±0.03)	**0.930 (±0.01)**	0.574 (±0.06)
	Federated	0.919 (±0.02)	**0.678^{++}(±0.06)**	0.916 (±0.03)	**0.672^{+}(±0.05)**	0.884 (±0.02)	0.600^{+}(±0.01)	0.911 (±0.03)	**0.677 (±0.06)**
	Synthesis	0.928 (±0.01)	0.269 (±0.29)	**0.943 (±0.01)**	0.394 (±0.25)	**0.932 (±0.03)**	**0.657^{+}(±0.07)**	0.922 (±0.03)	0.419 (±0.33)
50	Local	0.932 (±0.02)	0.569 (±0.07)	**0.931 (±0.02)**	0.588 (±0.06)	0.926 (±0.02)	0.478 (±0.02)	0.932 (±0.01)	0.574 (±0.02)
	Federated	0.917 (±0.02)	**0.675 (±0.06)**	0.915 (±0.02)	**0.674 (±0.06)**	0.875^{--} (±0.02)	0.599^{+}(±0.00)	0.901 (±0.03)	0.663^{++}(±0.05)
	Synthesis	**0.943 (±0.01)**	0.414 (±0.14)	0.924 (±0.04)	0.377 (±0.31)	**0.944 (±0.03)**	**0.680^{+}(±0.07)**	**0.934 (±0.04)**	**0.679^{+}(±0.06)**

Table A.26: Results for REPEN using Probe data set: Significance of federated learning and synthetic-based learning results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		**Feature-based non-i.i.d.**		**Label-based non-i.i.d.**		**LDP non-i.i.d.**	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.945 (±0.02)	0.731 (±0.05)	-	-	-	-	-	-
2	Local	**0.929 (±0.01)**	0.571 (±0.01)	**0.933 (±0.01)**	**0.591 (±0.04)**	0.934 (±0.02)	0.659 (±0.02)	0.923 (±0.01)	0.566 (±0.02)
	Federated	0.898 (±0.05)	0.516 (±0.16)	0.903 (±0.07)	0.579 (±0.15)	**0.948 (±0.01)**	**0.676 (±0.04)**	0.859 (±0.08)	0.448 (±0.13)
	Synthesis	0.926 (±0.03)	**0.588 (±0.10)**	0.884 (±0.05)	0.488 (±0.11)	0.872^{-}(±0.02)	0.440^{-}(±0.03)	**0.939 (±0.02)**	**0.615 (±0.10)**
5	Local	0.916 (±0.00)	0.554 (±0.01)	**0.902 (±0.01)**	**0.533 (±0.02)**	0.895 (±0.01)	0.578 (±0.01)	**0.921 (±0.01)**	0.576 (±0.04)
	Federated	0.824 (±0.11)	0.467 (±0.11)	0.895 (±0.01)	0.469^{-}(±0.01)	0.854 (±0.04)	0.440^{-}(±0.08)	0.869^{--}(±0.03)	0.425^{-}(±0.05)
	Synthesis	**0.920 (±0.02)**	**0.652^{+}(±0.01)**	0.881 (±0.07)	0.526 (±0.17)	**0.930^{+}(±0.01)**	**0.670^{++}(±0.06)**	0.919 (±0.02)	**0.662^{++}(±0.05)**
10	Local	**0.918 (±0.01)**	**0.559 (±0.01)**	0.901 (±0.01)	0.552 (±0.02)	**0.872 (±0.04)**	0.559 (±0.03)	0.886 (±0.02)	0.520 (±0.04)
	Federated	0.899^{--}(±0.01)	0.521 (±0.03)	0.839 (±0.09)	0.455 (±0.14)	0.861 (±0.06)	0.552 (±0.07)	0.743^{-}(±0.04)	0.329^{-}(±0.09)
	Synthesis	0.825 (±0.10)	0.405 (±0.17)	**0.948^{+}(±0.01)**	**0.667^{+}(±0.06)**	0.865 (±0.15)	**0.578 (±0.29)**	**0.921 (±0.04)**	**0.637 (±0.11)**
20	Local	0.914 (±0.00)	0.574 (±0.01)	0.906 (±0.00)	**0.573 (±0.00)**	0.877 (±0.03)	0.565 (±0.04)	**0.904 (±0.01)**	**0.564 (±0.03)**
	Federated	0.900 (±0.06)	0.564 (±0.13)	**0.908 (±0.02)**	0.549^{-}(±0.00)	0.880 (±0.08)	0.544 (±0.15)	0.851 (±0.13)	0.497 (±0.18)
	Synthesis	**0.951^{+}(±0.01)**	**0.746^{+}(±0.04)**	0.899 (±0.03)	0.544 (±0.16)	**0.957^{+}(±0.01)**	**0.685^{++}(±0.08)**	0.881 (±0.05)	0.475 (±0.14)
30	Local	**0.919 (±0.00)**	**0.599 (±0.01)**	0.916 (±0.01)	0.599 (±0.01)	0.873 (±0.03)	**0.555 (±0.03)**	0.897 (±0.01)	0.565 (±0.02)
	Federated	0.913 (±0.00)	0.547 (±0.04)	**0.935^{++}(±0.01)**	**0.660^{++}(±0.05)**	**0.896 (±0.03)**	0.554 (±0.13)	0.926^{+}(±0.01)	0.602 (±0.04)
	Synthesis	0.899 (±0.04)	0.501 (±0.14)	0.924 (±0.05)	0.630 (±0.18)	0.882 (±0.11)	0.550 (±0.25)	**0.936^{+}(±0.00)**	**0.610 (±0.05)**
40	Local	0.919 (±0.01)	0.602 (±0.02)	0.911 (±0.00)	**0.596 (±0.01)**	0.876 (±0.01)	0.571 (±0.01)	**0.903 (±0.00)**	**0.587 (±0.01)**
	Federated	**0.936 (±0.02)**	**0.680^{+}(±0.02)**	**0.917 (±0.03)**	0.569 (±0.11)	0.833^{-}(±0.01)	0.506 (±0.05)	0.877 (±0.07)	0.490 (±0.12)
	Synthesis	0.818^{-}(±0.06)	0.386^{-}(±0.06)	0.905 (±0.06)	0.589 (±0.21)	**0.936^{+}(±0.03)**	**0.648 (±0.17)**	0.892 (±0.05)	0.559 (±0.03)
50	Local	0.920 (±0.00)	0.613 (±0.01)	0.911 (±0.00)	0.609 (±0.01)	0.878 (±0.00)	0.576 (±0.02)	**0.903 (±0.01)**	**0.608 (±0.02)**
	Federated	**0.922 (±0.02)**	**0.644^{++}(±0.02)**	**0.918 (±0.02)**	**0.632 (±0.07)**	0.861 (±0.03)	0.585 (±0.05)	0.898 (±0.03)	0.549 (±0.08)
	Synthesis	0.860 (±0.07)	0.462 (±0.19)	0.904 (±0.05)	0.570 (±0.17)	**0.936^{+}(±0.03)**	**0.648 (±0.15)**	0.877 (±0.08)	0.540 (±0.18)

Table A.27: Results for REPEN using R2L data set: Significance of federated learning and synthetic-based learning results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		Feature-based non-i.i.d.		Label-based non-i.i.d.		LDP non-i.i.d.	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.807 (±0.01)	0.107 (±0.01)	-	-	-	-	-	-
2	Local	0.807 (±0.05)	0.118 (±0.03)	**0.805 (±0.01)**	**0.122 (±0.01)**	**0.814 (±0.03)**	**0.129 (±0.02)**	0.812 (±0.03)	0.128 (±0.02)
	Federated	**0.838 (±0.02)**	**0.135 (±0.02)**	0.773^{-} (±0.01)	0.108 (±0.00)	0.748 (±0.10)	0.104 (±0.04)	0.805 (±0.02)	0.123 (±0.02)
	Synthesis	0.759 (±0.06)	0.103 (±0.04)	0.763 (±0.05)	0.094^{--} (±0.02)	0.806 (±0.03)	0.108 (±0.01)	**0.822 (±0.05)**	**0.138 (±0.03)**
5	Local	**0.802 (±0.01)**	**0.128 (±0.02)**	**0.819 (±0.01)**	**0.155 (±0.02)**	**0.816 (±0.03)**	**0.146 (±0.03)**	**0.795 (±0.02)**	0.131 (±0.03)
	Federated	0.735 (±0.06)	0.100 (±0.02)	0.719^{--} (±0.07)	0.102^{--} (±0.03)	0.737 (±0.06)	0.098^{-} (±0.01)	0.770 (±0.03)	0.117 (±0.01)
	Synthesis	0.772 (±0.09)	0.121 (±0.05)	0.761 (±0.05)	0.143 (±0.09)	0.803 (±0.02)	0.128 (±0.04)	0.741 (±0.06)	**0.134 (±0.09)**
10	Local	0.779 (±0.00)	**0.126 (±0.01)**	0.796 (±0.01)	**0.132 (±0.01)**	**0.802 (±0.02)**	**0.145 (±0.01)**	**0.794 (±0.00)**	**0.124 (±0.01)**
	Federated	0.724 (±0.11)	0.098 (±0.03)	**0.819 (±0.04)**	0.130 (±0.02)	0.681 (±0.15)	0.100 (±0.05)	0.599 (±0.21)	0.082 (±0.04)
	Synthesis	**0.809 (±0.04)**	0.119 (±0.02)	0.747 (±0.10)	0.091^{--} (±0.03)	0.773 (±0.09)	0.103 (±0.03)	0.789 (±0.01)	0.103^{-} (±0.01)
20	Local	0.770 (±0.01)	**0.123 (±0.01)**	**0.784 (±0.00)**	0.129 (±0.00)	**0.782 (±0.01)**	**0.129 (±0.01)**	0.788 (±0.01)	**0.130 (±0.02)**
	Federated	0.705 (±0.15)	0.111 (±0.02)	0.778 (±0.04)	**0.130 (±0.03)**	0.709 (±0.09)	0.112 (±0.03)	0.687^{--} (±0.07)	0.095^{--} (±0.02)
	Synthesis	**0.781 (±0.07)**	0.107 (±0.03)	0.766 (±0.12)	0.112 (±0.05)	0.774 (±0.05)	0.099^{--} (±0.02)	**0.799 (±0.04)**	0.108 (±0.02)
30	Local	0.773 (±0.02)	0.125 (±0.01)	0.777 (±0.01)	**0.126 (±0.01)**	0.780 (±0.01)	0.127 (±0.01)	0.778 (±0.01)	0.132 (±0.01)
	Federated	0.796 (±0.03)	**0.131 (±0.02)**	0.722 (±0.09)	0.110 (±0.03)	0.724 (±0.12)	0.097 (±0.03)	0.785 (±0.03)	**0.134 (±0.03)**
	Synthesis	**0.817 (±0.05)**	0.126 (±0.03)	**0.788 (±0.02)**	0.124 (±0.03)	**0.829^{++}(±0.03)**	**0.146 (±0.04)**	**0.803 (±0.09)**	0.128 (±0.07)
40	Local	0.777 (±0.01)	0.127 (±0.00)	0.775 (±0.02)	0.125 (±0.00)	0.783 (±0.00)	0.127 (±0.00)	**0.775 (±0.00)**	**0.122 (±0.01)**
	Federated	0.752 (±0.05)	**0.127 (±0.03)**	0.782 (±0.06)	0.128 (±0.03)	0.773 (±0.09)	**0.146^{+}(±0.01)**	0.718 (±0.05)	0.103 (±0.01)
	Synthesis	**0.810 (±0.03)**	0.127 (±0.03)	**0.811^{++}(±0.01)**	**0.172 (±0.11)**	**0.804 (±0.03)**	0.110 (±0.01)	0.771 (±0.03)	0.099^{--} (±0.02)
50	Local	0.775 (±0.01)	0.120 (±0.01)	0.770 (±0.02)	0.121 (±0.00)	**0.777 (±0.01)**	**0.126 (±0.00)**	0.773 (±0.01)	0.121 (±0.00)
	Federated	0.728 (±0.05)	0.107 (±0.02)	0.731 (±0.10)	0.099 (±0.03)	0.704^{-} (±0.02)	0.104 (±0.02)	0.758 (±0.08)	**0.133 (±0.05)**
	Synthesis	**0.799 (±0.09)**	**0.197 (±0.18)**	**0.828^{++}(±0.04)**	**0.148 (±0.06)**	0.730 (±0.12)	0.092 (±0.03)	**0.804 (±0.03)**	0.117 (±0.02)

Table A.28: Results for REPEN using Annthyroid data set: Significance of **federated learning** and **synthetic-based learning** results compared to local learning is given with + for $p < 0.05$ and ++ for $p < 0.1$

		i.i.d.		**Feature-based non-i.i.d.**		**Label-based non-i.i.d.**		**LDP non-i.i.d.**	
N	Scenario	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC
-	Centralized	0.638 (±0.02)	0.168 (±0.02)	-	-	-	-	-	-
2	Local	0.549 (±0.03)	0.102 (±0.02)	**0.608 (±0.05)**	**0.142 (±0.04)**	0.562 (±0.04)	0.115 (±0.00)	0.593 (±0.05)	0.122 (±0.02)
	Federated	0.571 (±0.03)	0.104 (±0.02)	0.544 (±0.09)	0.117 (±0.06)	**0.601 (±0.03)**	**0.140 (±0.02)**	**0.593 (±0.02)**	**0.129 (±0.02)**
	Synthesis	0.674^{+}(±0.03)	0.227^{+}(±0.05)	0.553 (±0.12)	0.129 (±0.09)	0.574 (±0.07)	0.138 (±0.07)	0.533 (±0.08)	0.113 (±0.04)
5	Local	0.583 (±0.04)	0.158 (±0.04)	0.590 (±0.04)	0.141 (±0.03)	0.574 (±0.03)	0.135 (±0.03)	0.603 (±0.02)	0.151 (±0.02)
	Federated	**0.648 (±0.10)**	**0.206 (±0.07)**	**0.647^{++}(±0.02)**	**0.181 (±0.04)**	**0.669^{++}(±0.06)**	**0.173 (±0.05)**	**0.678^{+}(±0.03)**	**0.199 (±0.05)**
	Synthesis	0.613 (±0.05)	0.169 (±0.04)	0.572 (±0.05)	0.131 (±0.03)	0.551 (±0.03)	0.121 (±0.03)	0.605 (±0.05)	0.167 (±0.02)
10	Local	0.581 (±0.02)	0.142 (±0.01)	0.567 (±0.00)	0.137 (±0.01)	0.596 (±0.02)	**0.148 (±0.01)**	0.602 (±0.03)	0.166 (±0.03)
	Federated	**0.664^{+}(±0.03)**	**0.219 (±0.08)**	**0.670 (±0.08)**	**0.236 (±0.14)**	**0.623 (±0.03)**	0.146 (±0.05)	**0.693^{++}(±0.06)**	**0.248 (±0.12)**
	Synthesis	0.609 (±0.04)	0.153 (±0.06)	0.593^{++}(±0.02)	0.181^{++}(±0.03)	0.611 (±0.06)	0.133 (±0.06)	0.635 (±0.02)	0.185 (±0.03)
20	Local	0.584 (±0.01)	0.154 (±0.02)	0.567 (±0.02)	0.137 (±0.01)	0.580 (±0.02)	0.135 (±0.01)	0.571 (±0.02)	0.138 (±0.01)
	Federated	**0.655^{+}(±0.01)**	**0.175 (±0.02)**	**0.640^{+}(±0.01)**	**0.158 (±0.02)**	**0.637^{+}(±0.02)**	0.145 (±0.04)	**0.627^{+}(±0.02)**	0.134 (±0.03)
	Synthesis	0.612 (±0.07)	0.168 (±0.10)	0.571 (±0.10)	0.116 (±0.05)	0.588 (±0.04)	**0.166 (±0.07)**	0.609 (±0.10)	**0.190 (±0.13)**
30	Local	0.585 (±0.03)	0.161 (±0.03)	0.580 (±0.01)	0.149 (±0.01)	0.585 (±0.02)	0.143 (±0.01)	0.566 (±0.01)	0.136 (±0.01)
	Federated	**0.656^{+}(±0.02)**	**0.192 (±0.03)**	**0.623^{+}(±0.02)**	0.150 (±0.01)	**0.628 (±0.04)**	0.136 (±0.02)	**0.656^{+}(±0.04)**	**0.149 (±0.03)**
	Synthesis	0.617 (±0.07)	0.175 (±0.05)	0.565 (±0.10)	**0.182 (±0.10)**	0.596 (±0.10)	**0.147 (±0.05)**	0.589 (±0.03)	0.145 (±0.05)
40	Local	0.589 (±0.01)	0.154 (±0.01)	0.584 (±0.01)	**0.149 (±0.01)**	0.589 (±0.03)	0.150 (±0.03)	0.580 (±0.02)	0.142 (±0.01)
	Federated	**0.676^{+}(±0.02)**	**0.190 (±0.04)**	**0.627^{++}(±0.03)**	0.145 (±0.03)	0.632 (±0.02)	0.170 (±0.01)	**0.655^{+}(±0.03)**	0.153 (±0.01)
	Synthesis	0.622 (±0.06)	0.176 (±0.09)	0.552 (±0.03)	0.125 (±0.06)	**0.645 (±0.07)**	**0.188 (±0.07)**	0.624 (±0.03)	**0.162 (±0.05)**
50	Local	0.583 (±0.01)	0.151 (±0.01)	0.577 (±0.02)	0.149 (±0.01)	0.577 (±0.02)	0.146 (±0.02)	0.586 (±0.01)	0.154 (±0.01)
	Federated	**0.651^{+}(±0.01)**	**0.166 (±0.02)**	**0.624^{+}(±0.01)**	0.135 (±0.01)	0.562 (±0.13)	0.106 (±0.05)	0.615 (±0.06)	0.128 (±0.04)
	Synthesis	0.625 (±0.04)	0.150 (±0.04)	0.571 (±0.07)	**0.152 (±0.04)**	0.643^{++}(±0.04)	**0.183 (±0.03)**	**0.624 (±0.09)**	**0.200 (±0.11)**

List of Figures

List of Tables

Acronyms

AdaBoost Adaptive Boosting. 28

Adam Adaptive Moment Estimation. 48, 49

AE Autoencoder. 15, 23, 26, 34, 39, 49, 50, 115

BCELoss Binary Cross Entropy Loss. 48

CART Classification and Regression Trees. 33, 34

CBLOF Cluster-based Local Outlier Factor. 15, 19, 21

COF Connectivity-based Outlier Factor. 15, 20

DAGMM Deep Autoencoding Gaussian Mixture Model. 15, 26, 113

DBSCAN Density-Based Spatial Clustering of Applications with Noise. 15, 19

DevNet Deviation Network. 15, 25, 49, 50, 71, 72, 80, 81, 96–99, 113, 115, 116

EMD Earth Mover's Distance. 52–54, 82, 113

FedAvg Federated averaging. 34–36, 57

FedSGD Federated Stochastic Gradient Descent. 57

FFNN Feedforward Neural Network. 15, 21, 22, 47–49, 60–63, 65, 66, 69, 70, 73, 80, 81, 84–87, 113–115

FN False Negative. 10

FNR False Negative Rate. 11, 13

FP False Positive. 10

FPR False Positive Rate. 11, 13

FSVRG Federated Stochastic Variance Reduced Gradient. 57

GAN Generative Adversarial Networks. 34

GB Gradient Boosting. 28, 29, 113

GMM Gaussian Mixture Model. 26, 34, 39

GPU Graphics Processing Unit. 48

i.i.d. independent and identically distributed. 3, 4, 36–38, 41, 51–53, 59–65, 67, 68, 71–73, 75–79, 81, 82, 114, 115

IF Isolation Forest. 15, 27, 34, 35, 39, 50, 57, 75

LDP Label-based Dirichlet Partition. 52, 62, 63, 65, 66, 68–70, 76–78, 114, 115

LOF Local Outlier Factor. 15, 20, 21

LReLU Leaky Rectified Linear Unit. 49, 50

LSTM Long Short-Term Memory. 39, 40

MLP Multilayer Perceptron. 21, 39

MSE Mean Squared Error. 49

non-i.i.d. non independent and identically distributed. 2–4, 37, 38, 41, 51–53, 59–73, 75–79, 81, 82, 113–115

NPV Negative Predictive Value. 11

OC-SVM One-Class Support Vector Machine. 15, 21, 26, 40

PCA Principal Component Analysis. 44, 45, 82

PPML Privacy-Preserving Machine Learning. 32, 33

PPV Positive Predictive Value. 11

PR Precision-Recall. 13, 14, 53, 63, 83, 113, 114, 118

PR AUC Area under the PR curve. 14, 54, 59–73, 75, 77, 78, 80, 81, 114

ReLU Rectified Linear Activation Unit. 48, 49

REPEN REPresentations for a random nEarest Neighbor. 15, 24, 50, 51, 75, 77, 78, 108–111, 113, 115, 116

ROC Receiver Operator Characteristic. 13, 14, 53, 83, 113, 118

ROC AUC Area under the ROC curve. 14, 40, 54, 59, 61–63, 65, 67, 68, 72, 73, 75, 77, 78, 80, 81

SDV Synthetic Data Vault. 33, 34

SGD Stochastic Gradient Descent. 48, 49

SMOTE Synthetic Minority Oversampling Technique. 58, 70, 81

SVM Support vector machine. 9

TN True Negative. 10

TP True Positive. 10

TPR True Positive Rate. 11, 13

VAE Variational Autoencoder. 15, 34, 40

XGBoost Extreme Gradient Boosting. 15, 29, 35, 47, 49, 57, 67–69, 80, 93–95, 115, 116

Bibliography

[AA19] Sridhar Alla and Suman Kalyan Adari. *Beginning Anomaly Detection Using Python-Based Deep Learning: With Keras and PyTorch.* Apress, Berkeley, CA, 2019.

[AB15] Dal Pozzolo Andrea and Gianluca Bontempi. *Adaptive Machine Learning for Credit Card Fraud Detection.* PhD thesis, Université Libre de Bruxelles, 2015.

[AC15] Jinwon An and Sungzoon Cho. Variational Autoencoder based Anomaly Detection using Reconstruction Probability. In *Special Lecture on IE (2015)*, 2015.

[ADE20] Arwa Aldweesh, Abdelouahid Derhab, and Ahmed Z. Emam. Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowledge-Based Systems*, 189:105124, February 2020.

[Agg13] Charu C. Aggarwal. Outlier ensembles: Position paper. *SIGKDD Explor. Newsl.*, 14(2):49–58, April 2013.

[AMM$^+$21] Redhwan Al-amri, Raja Kumar Murugesan, Mustafa Man, Alaa Fareed Abdulateef, Mohammed A. Al-Sharafi, and Ammar Ahmed Alkahtani. A Review of Machine Learning and Deep Learning Techniques for Anomaly Detection in IoT Data. *Applied Sciences*, 11(12):5320, June 2021.

[ASA$^+$21] Shaashwat Agrawal, Sagnik Sarkar, Ons Aouedi, Gokul Yenduri, Kandaraj Piamrat, Sweta Bhattacharya, Praveen Kumar Reddy Maddikunta, and Thippa Reddy Gadekallu. Federated Learning for Intrusion Detection System: Concepts, Challenges and Future Directions, June 2021.

[Bab15] V. Kishore Babu. Detection of Probe Attacks Using Machine Learning Techniques. In *International Journal of Research Studies in Computer Science and Engineering*, 2015.

[BCM$^+$18] Theodora S. Brisimi, Ruidi Chen, Theofanie Mela, Alex Olshevsky, Ioannis Ch. Paschalidis, and Wei Shi. Federated learning of predictive models from federated Electronic Health Records. *International Journal of Medical Informatics*, 112:59–67, April 2018.

[BDR18] Steven M. Bellovin, Preetam K. Dutta, and Nathan Reitinger. Privacy and Synthetic Datasets. *SSRN Journal*, 2018.

[BFOS17] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification And Regression Trees.* Routledge, first edition, October 2017.

[BKL$^+$21] Saikiran Bulusu, Bhavya Kailkhura, Bo Li, Pramod K. Varshney, and Dawn Song. Anomalous Example Detection in Deep Learning: A Survey, February 2021.

[BKNS00] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. LOF: Identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104, June 2000.

[BKW20] Thomas Bäck, Jiawen Kong, and Kowalczyk Wojtek. Improving Imbalanced Classification by Anomaly Detection. In Mike Preuß, André Deutz, Hao Wang, Carola Doerr, Michael Emmerich, and Heike Trautmann, editors, *Parallel Problem Solving from Nature - PPSN XVI. Part 1*, number 12269 in Lecture Notes in Computer Science Theoretical Computer Science and General Issues, pages 512–523. Springer, Cham, 2020.

[Bot10] Léon Bottou. Large-Scale Machine Learning with Stochastic Gradient Descent. In Yves Lechevallier and Gilbert Saporta, editors, *Proceedings of COMPSTAT'2010*, pages 177–186. Physica-Verlag HD, Heidelberg, 2010.

[Bro16] J. Brownlee. *XGBoost With Python: Gradient Boosted Trees with XGBoost and Scikit-Learn*. Machine Learning Mastery, 2016.

[CBK09] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):1–58, July 2009.

[CC19] Raghavendra Chalapathy and Sanjay Chawla. Deep Learning for Anomaly Detection: A Survey, January 2019.

[CG16] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, August 2016.

[CJ20] Davide Chicco and Giuseppe Jurman. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1):6, December 2020.

[CLY17] Long Cheng, Fang Liu, and Danfeng Daphne Yao. Enterprise data breach: Causes, challenges, prevention, and future directions: Enterprise data breach. *WIREs Data Mining Knowl Discov*, 7(5):e1211, September 2017.

[CM22] Florencia Cavallin and Rudolf Mayer. Anomaly Detection from Distributed Data Sources via Federated Learning. In Leonard Barolli, Farookh Hussain, and Tomoya Enokido, editors, *Advanced Information Networking and Applications*, volume 450, pages 317–328. Springer International Publishing, Cham, 2022.

[CP09] Peter Christen and Agus Pudjijono. Accurate Synthetic Generation of Realistic Personal Information. In Thanaruk Theeramunkong, Boonserm Kijsirikul, Nick Cercone, and Tu-Bao Ho, editors, *Advances in Knowledge Discovery and Data Mining*, volume 5476, pages 507–514. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[CRS20] Peter Christen, Thilina Ranbaduge, and Rainer Schnell. *Linking Sensitive Data: Methods and Techniques for Practical Privacy-Preserving Information Sharing*. Springer, Cham, 2020.

[DBD+05] Jesse Davis, Elizabeth S. Burnside, Inês de Castro Dutra, David Page, Raghu Ramakrishnan, Vítor Santos Costa, and Jude W. Shavlik. View Learning for Statistical Relational Learning: With an Application to Mammography. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005*, pages 677–683. Professional Book Center, 2005.

[DDRF+22] Isaac Martín De Diego, Ana R. Redondo, Rubén R. Fernández, Jorge Navarro, and Javier M. Moguerza. General Performance Score for classification problems. *Appl Intell*, 52(10):12049–12063, August 2022.

[DG06] Jesse Davis and Mark Goadrich. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd International Conference on Machine Learning - ICML '06*, pages 233–240, Pittsburgh, Pennsylvania, 2006. ACM Press.

[DG17] Dheeru Dua and Casey Graff. UCI Machine Learning Repository, 2017.

[Die98] Thomas G. Dietterich. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, 10(7):1895–1923, October 1998.

[DKP+20] Georgios Drainakis, Konstantinos V. Katsaros, Panagiotis Pantazopoulos, Vasilis Sourlas, and Angelos Amditis. Federated vs. Centralized Machine Learning under Privacy-elastic Users: A Comparative Analysis. In *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*, pages 1–8, Cambridge, MA, USA, November 2020. IEEE.

[Don93] Bruce Rubin Donald. Discussion: Statistical Disclosure Limitation. *Journal of Official Statistics*, Vol. 9. No. 2:461–468, 1993.

[DPS+18] Abhishek Divekar, Meet Parekh, Vaibhav Savla, Rudra Mishra, and Mahesh Shirole. Benchmarking datasets for Anomaly-based Network Intrusion Detection: KDD CUP 99 alternatives. In *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*, pages 1–8, Kathmandu, October 2018. IEEE.

[Dwo08] Cynthia Dwork. Differential Privacy: A Survey of Results. In Manindra Agrawal, Dingzhu Du, Zhenhua Duan, and Angsheng Li, editors, *Theory and Applications of Models of Computation*, volume 4978, pages 1–19. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Knowledge Discovery and Data Mining*, 1996.

[FGG+18] Alberto Fernández, Mikel Galar, Salvador García, Francisco Herrera, Bartosz Krawczyk, and Ronaldo C. Prati. *Learning from Imbalanced Data Sets*. Springer International Publishing : Imprint: Springer, Cham, 1st ed. 2018 edition, 2018.

[Fri02] Jerome H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, February 2002.

[Gab10] Mohamed Medhat Gaber. *Knowledge Discovery from Sensor Data: Second International Workshop, Sensor-KDD 2008, Las Vegas, NV, USA, August 24-27, 2008: Revised Selected Papers*. Number 5840 in Lecture Notes in Computer Science. Springer, Berlin ; New York, 2010.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Adaptive Computation and Machine Learning. The MIT Press, Cambridge, Massachusetts, 2016.

[Gee15] Sunder Gee. *Fraud and Fraud Detection: A Data Analytics Approach*. Wiley Corporate F&A Series. Wiley, Hoboken, New Jersey, 2015.

[Gér19] Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc, Beijing [China] ; Sebastopol, CA, second edition edition, 2019.

[GGAH14] Manish Gupta, Jing Gao, Charu Aggarwal, and Jiawei Han. *Outlier Detection for Temporal Data*. Synthesis Lectures on Data Mining and Knowledge Discovery. Springer International Publishing, Cham, 2014.

[GPM+14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks, June 2014.

[GR12] Manasi Gyanchandani and J. L. Rana. Taxonomy of Anomaly Based Intrusion Detection System: A Review. In *International Journal of Scientific and Research Publications*, volume Volume 2, Issue 12, 2012.

[GU16] Markus Goldstein and Seiichi Uchida. A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. *PLoS ONE*, 11(4):e0152173, April 2016.

[GZW+20] Rui Guo, Zhiqian Zhao, Tao Wang, Guangheng Liu, Jingyi Zhao, and Dianrong Gao. Degradation State Recognition of Piston Pump Based on ICEEMDAN and XGBoost. *Applied Sciences*, 10(18):6593, September 2020.

[Han82] Lars Peter Hansen. Large Sample Properties of Generalized Method of Moments Estimators. *Econometrica*, 50(4):1029, July 1982.

[HBL+21] Truong Thu Huong, Ta Phuong Bac, Dao Minh Long, Tran Duc Luong, Nguyen Minh Dan, Le Anh Quang, Le Thanh Cong, Bui Doan Thang, and Kim Phuc Tran. Detecting cyberattacks using anomaly detection in industrial control systems: A Federated Learning approach. *Computers in Industry*, 132:103509, November 2021.

[HC13] T. Ryan Hoens and Nitesh V. Chawla. Imbalanced Datasets: From Sampling to Classifiers. In Haibo He and Yunqian Ma, editors, *Imbalanced Learning*, pages 43–59. John Wiley & Sons, Inc., Hoboken, NJ, USA, June 2013.

[HHH+22] Songqiao Han, Xiyang Hu, Hailiang Huang, Mingqi Jiang, and Yue Zhao. ADBench: Anomaly Detection Benchmark, September 2022.

[HKP12] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Elsevier, third edition edition, 2012.

[Hub19] Franz Huber. *A Logical Introduction to Probability and Induction*. Oxford Universty Press, New York, NY, 2019.

[HXD02] Zengyou He, Xiaofei Xu, and Shengchun Deng. Squeezer: An efficient algorithm for clustering categorical data. *J. Comput. Sci. & Technol.*, 17(5):611–624, September 2002.

[HXD03] Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9-10):1641–1650, June 2003.

[JLLK20] Chunggil Jung, Yonggwan Lee, Jiwan Lee, and Seongjoon Kim. Performance Evaluation of the Multiple Quantile Regression Model for Estimating Spatial Soil Moisture after Filtering Soil Moisture Outliers. *Remote Sensing*, 12(10):1678, May 2020.

[KB17] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017.

[KMA+21] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and Open Problems in Federated Learning, March 2021.

[KW22] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes, December 2022.

[LDCH21] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated Learning on Non-IID Data Silos: An Experimental Study, October 2021.

[LHD+20] Yunlong Lu, Xiaohong Huang, Yueyue Dai, Sabita Maharjan, and Yan Zhang. Differentially Private Asynchronous Federated Learning for Mobile Edge Computing in Urban Informatics. *IEEE Trans. Ind. Inf.*, 16(3):2134–2143, March 2020.

[Llo82] S. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inform. Theory*, 28(2):129–137, March 1982.

[LM22] Majlinda Llugiqi and Rudolf Mayer. An Empirical Analysis of Synthetic-Data-Based Anomaly Detection. In Andreas Holzinger, Peter Kieseberg, A Min Tjoa, and Edgar Weippl, editors, *Machine Learning and Knowledge Extraction*, volume 13480, pages 306–327. Springer International Publishing, Cham, 2022.

[LMX+19] Wenqi Li, Fausto Milletarì, Daguang Xu, Nicola Rieke, Jonny Hancox, Wentao Zhu, Maximilian Baust, Yan Cheng, Sébastien Ourselin, M. Jorge Cardoso, and Andrew Feng. Privacy-Preserving Federated Brain Tumour Segmentation. In Heung-Il

Suk, Mingxia Liu, Pingkun Yan, and Chunfeng Lian, editors, *Machine Learning in Medical Imaging*, volume 11861, pages 133–141. Springer International Publishing, Cham, 2019.

[LNA16] Guillaume Lemaitre, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning, September 2016.

[Loc75] Robert H. Lochner. A Generalized Dirichlet Distribution in Bayesian Life Testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 37(1):103–113, September 1975.

[LSTS20] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Process. Mag.*, 37(3):50–60, May 2020.

[LTJZ20] Guodong Long, Yue Tan, Jing Jiang, and Chengqi Zhang. Federated Learning for Open Banking. In Qiang Yang, Lixin Fan, and Han Yu, editors, *Federated Learning*, volume 12500, pages 240–254. Springer International Publishing, Cham, 2020.

[LTZ08] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation Forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422, Pisa, Italy, December 2008. IEEE.

[MKP+22] Viraaji Mothukuri, Prachi Khare, Reza M. Parizi, Seyedamin Pouriyeh, Ali Dehghantanha, and Gautam Srivastava. Federated-Learning-Based Anomaly Detection for IoT Security Attacks. *IEEE Internet Things J.*, 9(4):2545–2554, February 2022.

[MMH17] Kishan G. Mehrotra, Chilukuri K. Mohan, and HuaMing Huang. *Anomaly Detection Principles and Algorithms*. Terrorism, Security, and Computation. Springer International Publishing, Cham, 2017.

[MMR+17] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data, February 2017.

[MN21] David Meyer and Thomas Nagler. Synthia: Multidimensional synthetic data generation in Python. *JOSS*, 6(65):2863, September 2021.

[MS99] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Mass, 1999.

[MV17] Ricardo Mendes and Joao P. Vilela. Privacy-Preserving Data Mining: Methods, Metrics, and Applications. *IEEE Access*, 5:10562–10582, 2017.

[NRD16] Beata Nowok, Gillian M. Raab, and Chris Dibben. **Synthpop** : Bespoke Creation of Synthetic Data in *R*. *J. Stat. Soft.*, 74(11), 2016.

[NSU+18] Adrian Nilsson, Simon Smith, Gregor Ulm, Emil Gustavsson, and Mats Jirstrand. A Performance Evaluation of Federated Learning Algorithms. In *Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning*, pages 1–8, Rennes France, December 2018. ACM.

[NVP22] Mirko Nardi, Lorenzo Valerio, and Andrea Passarella. Anomaly Detection through Unsupervised Federated Learning, September 2022.

[PCCL18] Guansong Pang, Longbing Cao, Ling Chen, and Huan Liu. Learning Representations of Ultrahigh-dimensional Data for Random Distance-based Outlier Detection. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2041–2050, July 2018.

[PCJB15] Andrea Dal Pozzolo, Olivier Caelen, Reid A. Johnson, and Gianluca Bontempi. Calibrating Probability with Undersampling for Unbalanced Classification. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 159–166, Cape Town, South Africa, December 2015. IEEE.

[PF97] Foster Provost and Tom Fawcett. Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, KDD'97, pages 43–48. AAAI Press, 1997.

[PFK98] Foster J. Provost, Tom Fawcett, and Ron Kohavi. The Case against Accuracy Estimation for Comparing Induction Algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 445–453, San Francisco, CA, USA, July 1998. Morgan Kaufmann Publishers Inc.

[PSCvdH22] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton van den Hengel. Deep Learning for Anomaly Detection: A Review. *ACM Comput. Surv.*, 54(2):1–38, March 2022.

[PSH17] Haoyue Ping, Julia Stoyanovich, and Bill Howe. DataSynthesizer: Privacy-Preserving Synthetic Datasets. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, pages 1–5, Chicago IL USA, June 2017. ACM.

[PSvdH19] Guansong Pang, Chunhua Shen, and Anton van den Hengel. Deep Anomaly Detection with Deviation Networks, November 2019.

[PTE$^+$20] Harsh Purohit, Ryo Tanabe, Takashi Endo, Kaori Suefusa, Yuki Nikaido, and Yohei Kawaguchi. Deep Autoencoding GMM-based Unsupervised Anomaly Detection in Acoustic Signals and its Hyper-parameter Optimization, September 2020.

[PWV16] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. The Synthetic Data Vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 399–410, Montreal, QC, Canada, October 2016. IEEE.

[PZJL22] Jiaming Pei, Kaiyang Zhong, Mian Ahmad Jan, and Jinhai Li. Personalized federated learning framework for network traffic anomaly detection. *Computer Networks*, 209:108906, May 2022.

[RBB$^+$20] Debbie Rankin, Michaela Black, Raymond Bond, Jonathan Wallace, Maurice Mulvenna, and Gorka Epelde. Reliability of Supervised Machine Learning Using Synthetic Data in Health Care: Model to Preserve Privacy for Data Sharing. *JMIR Med Inform*, 8(7):e18910, July 2020.

[RBJ89] Vijay Raghavan, Peter Bollmann, and Gwang S. Jung. A critical investigation of recall and precision as measures of retrieval system performance. *ACM Trans. Inf. Syst.*, 7(3):205–229, July 1989.

[RD22] Statista Research Department. Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2025. https://www.statista.com/statistics/871513/worldwide-data-created/, May 2022.

[Rei05] Jerome P. Reiter. Using CART to generate partially synthetic public use microdata. *Journal of Official Statistics*, 21:441–462, 2005.

[RND18] Gillian M Raab, Beata Nowok, and Chris Dibben. Practical Data Synthesis for Large Samples. *JPC*, 7(3):67–97, February 2018.

[Rok19] Lior Rokach. *Ensemble Learning: Pattern Classification Using Ensemble Methods*. Number vol. 85 in Series in Machine Perception and Artificial Intelligence. World Scientific, New Jersey London Singapore Beijing Shanghai Hong Kong Taipei Chennai, second edition edition, 2019.

[RSMMA19] N. N. R. Ranga Suri, Narasimha Murty M, and G. Athithan. *Outlier Detection: Techniques and Applications: A Data Mining Perspective*, volume 155 of *Intelligent Systems Reference Library*. Springer International Publishing, Cham, 2019.

[Sch99] Robert E. Schapire. A Brief Introduction to Boosting. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'99, pages 1401–1406, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[Sch15] Juergen Schmidhuber. Deep Learning in Neural Networks: An Overview. *Neural Networks*, 61:85–117, January 2015.

[SH21] Raed Abdel Sater and A. Ben Hamza. A Federated Learning Approach to Anomaly Detection in Smart Buildings. *ACM Trans. Internet Things*, 2(4):1–23, November 2021.

[SR15] Takaya Saito and Marc Rehmsmeier. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLoS ONE*, 10(3):e0118432, March 2015.

[SWS$^+$99] Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support Vector Method for Novelty Detection. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999.

[SWW$^+$99] S.J. Stolfo, Wei Fan, Wenke Lee, A. Prodromidis, and P.K. Chan. Cost-based modeling for fraud and intrusion detection: Results from the JAM project. In *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*, volume 2, pages 130–144, Hilton Head, SC, USA, 1999. IEEE Comput. Soc.

[SZA$^+$20] Adil Hussain Seh, Mohammad Zarour, Mamdouh Alenezi, Amal Krishna Sarkar, Alka Agrawal, Rajeev Kumar, and Raees Ahmad Khan. Healthcare Data Breaches: Insights and Implications. *Healthcare*, 8(2):133, May 2020.

[TBJS20] Srikanth Thudumu, Philip Branch, Jiong Jin, and Jugdutt Singh. A comprehensive survey of anomaly detection techniques for high dimensional big data. *J Big Data*, 7(1):42, December 2020.

[TBLG09] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani. A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pages 1–6, Ottawa, ON, Canada, July 2009. IEEE.

[TCFC02] Jian Tang, Zhixiang Chen, Ada Wai-chee Fu, and David W. Cheung. Enhancing Effectiveness of Outlier Detections for Low Density Patterns. In G. Goos, J. Hartmanis, J. van Leeuwen, Ming-Syan Chen, Philip S. Yu, and Bing Liu, editors, *Advances in Knowledge Discovery and Data Mining*, volume 2336, pages 535–548. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[THX22] Zhongyun Tang, Haiyang Hu, and Chonghuan Xu. A federated learning method for network intrusion detection. *Concurrency and Computation*, 34(10), May 2022.

[TMKD17] Matthias Templ, Bernhard Meindl, Alexander Kowarik, and Olivier Dupriez. Simulation of Synthetic Complex Data: The *R* Package **simPop**. *J. Stat. Soft.*, 79(10), 2017.

[TSKK19] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, and Anuj Karpatne. *Introduction to Data Mining, Global Edition.* Pearson Education Limited, Harlow, United Kingdom, 2nd ed edition, 2019.

[Tuk77] John Wilder Tukey. *Exploratory Data Analysis.* Addison-Wesley Series in Behavioral Science. Addison-Wesley Pub. Co, Reading, Mass, 1977.

[Wes68] Alan Furman Westin. *Privacy and Freedom.* IG Publishing, New York, new edition edition, 1968.

[WXDL22] Hexu Wang, Fei Xie, Qun Duan, and Jing Li. Federated Learning for Supply Chain Demand Forecasting. *Mathematical Problems in Engineering*, 2022:1–8, November 2022.

[XBJ21] Runhua Xu, Nathalie Baracaldo, and James Joshi. Privacy-Preserving Machine Learning: Methods, Challenges and Directions, September 2021.

[XGS$^+$21] Jie Xu, Benjamin S. Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. Federated Learning for Healthcare Informatics. *J Healthc Inform Res*, 5(1):1–19, March 2021.

[XRY$^+$15] Dan Xu, Elisa Ricci, Yan Yan, Jingkuan Song, and Nicu Sebe. Learning Deep Representations of Appearance and Motion for Anomalous Event Detection, October 2015.

[XWW$^+$21] Wenchao Xia, Wanli Wen, Kai-Kit Wong, Tony Q.S. Quek, Jun Zhang, and Hongbo Zhu. Federated-Learning-Based Client Scheduling for Low-Latency Wireless Communications. *IEEE Wireless Commun.*, 28(2):32–38, April 2021.

[YZY$^+$19] Wensi Yang, Yuhang Zhang, Kejiang Ye, Li Li, and Cheng-Zhong Xu. FFD: A Federated Learning Based Method for Credit Card Fraud Detection. In Keke Chen, Sangeetha Seshadri, and Liang-Jie Zhang, editors, *Big Data – BigData 2019*, volume 11514, pages 18–32. Springer International Publishing, Cham, 2019.

[ZLL$^+$18] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated Learning with Non-IID Data. *Computing Research Repository (CoRR)*, 2018.

[ZXLJ21] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated Learning on Non-IID Data: A Survey. *Neurocomputing*, Volume 465, 2021.