Universiteit Leiden

Secure Multi-Party Logistic Regression for Detecting Unusual Medical Ordering Behaviour Across Hospitals

Michael Siswowijoto (s3081842)

Supervisors:
Dr. Dilek Öner Simsek & Dr. Eleftheria Makri

Bachelor Thesis– Informatica

Leiden Institute of Advanced Computer Science (LIACS)
liacs.leidenuniv.nl

June 30, 2026

Abstract

This thesis investigates whether secure multi-party computation can be used to train a logistic regression model for cross-hospital detection of unusual medical ordering behaviour. Hospitals may benefit from collaborative analysis, but patient-level data cannot easily be shared or centralized. This is due to privacy, legal, and governance constraints. To study this problem, the Myocardial Infarction Complications dataset is used as a healthcare case study. The dataset does not contain direct labels for guideline noncompliance. Therefore, anomaly labels are constructed during preprocessing using a distance-based procedure over treatment- and procedure-related features. This turns the problem into a supervised binary classification task.

The proposed pipeline horizontally partitions the processed dataset between hospitals and trains logistic regression securely using the MP-SPDZ framework. The results indicate that proper feature standardization was essential for stable secure training. With this preprocessing step, the best `semi2k` configuration achieved an accuracy of approximately 0.90, recall of approximately 0.92, and an F1 score of approximately 0.79. This shows that the secure model preserved strong predictive performance, especially for detecting anomalous records. However, this came with substantial overhead: about 4,600 seconds of runtime and 187 GB of communication per party. Compared with the plaintext baseline, the secure model achieved lower overall performance but maintained high recall. The main finding is that privacy-preserving cross-hospital anomaly detection is feasible in terms of predictive performance. Practical deployment depends strongly on runtime, communication cost, numerical stability, and protocol choice.

Contents

1 Introduction

Artificial intelligence is increasingly being applied in healthcare. This is being done to support clinical decision-making, improve efficiency or detect patterns in medical data. Hospitals collect large amounts of patient information. This includes diagnoses, treatments, procedures, medication decisions, and clinical outcomes. This data can help identify patterns in medical practice if properly analysed. One important use case is the analysis of diagnostic tests and medical procedures ordered for patients. In practice, some combinations of tests and procedures may deviate from common clinical patterns or established guidelines. In this thesis, such deviations are treated as potential signs of noncompliant clinical practice. This refers to clinical practice that does not follow expected guidelines or standard procedures [13].

Developing reliable healthcare models requires access to large and diverse datasets. A model trained on data from one hospital may mostly reflect the characteristics and practices of that specific institution. This can reduce the robustness and applicability of the model. Combining data from multiple hospitals can address this limitation. It will increase the number of patient records and include a wider range of clinical patterns [3]. This is especially important for detecting unusual medical ordering behaviour. These are patterns in ordered tests, treatments, procedures, or medication decisions that differ from what is normally seen. Such rare or atypical patterns may not be visible in data from one hospital alone.

However, heatlhcare data is highly sensitive. It contains private patient information and must be handled under strict, ethical, and organizational rules, including HIPAA and European privacy legislation [6]. As a result, it is difficult for hospitals to share or centralize raw patient data. Transferring patient data to one central location can create privacy risks, governance concerns, and security vulnerabilities [1]. This creates a conflict between collaborative healthcare research and the need to protect patient privacy.

Secure Multi-Party Computation (MPC) offers a possible solution to this problem. MPC is a cryptographic approach that allows multiple parties to jointly compute a function over their private inputs without revealing those inputs to each other [2, 18]. This means that multiple hospitals can collaborate on a shared computation, such as training a machine learning model, while keeping their local patient data private. Instead of exchanging raw data, the parties take part in a secure protocol that only reveals the intended output of the computation. This makes MPC a promising technique for privacy-preserving machine learning in cross-institutional medical research.

This thesis investigates how Secure Multi-Party Computation can be used to train a logistic regression model for detecting unusual medical ordering behaviour. In practice, hospitals may want to detect combinations of tests, treatments, or procedures that differ from normal clinical practice. Earlier work on clinical order sets shows that these outliers can point to possible deviations from expected practice [13]. In this thesis, these deviations are treated as signals that may point to potential guideline noncompliance.

The dataset used in this work does not directly state whether the ordered tests, treatments, or procedures followed clinical guidelines. Therefore, anomaly labels are constructed during preprocessing based on how strongly each procedure-related profile differs from the cohort average. These labels do not prove clinical noncompliance. They only indicate that the ordering pattern is unusual compared with the rest of the cohort. This provides a practical way to study unusual medical ordering as a supervised binary classification task.

Logistic regression is used as the predictive model because it fits binary classification tasks well and is still widely used in clinical prediction studies [17]. It is also easier to interpret than many more complex machine learning models. This is important in healthcare because model outputs may influence clinical interpretation or decision-making. Logistic regression is useful here because its predictions are easier to interpret than those of many more complex machine learning models [17]. It is a practical choice for secure computation. Most of its training can be expressed with arithmetic operations on shared values [12]. However, the sigmoid function is nonlinear. It therefore requires special handling and often needs an MPC-friendly approximation [12, 9]. In this thesis, the model is trained using the MP-SPDZ framework, which supports secure computation protocols, fixed-point arithmetic, and practical measurements such as runtime and communication cost [7].

The thesis considers a cross-hospital setting where each hospital keeps its own patient records. This allows hospitals to collaborate on model training without requiring them to centralize or directly share their raw data. In this work, the data is treated as horizontally partitioned. Each hospital has different patient records, but the same feature structure. This makes it possible to study collaborative model training without assuming that all data can be collected in one central place.

The secure computation setting is based on a semi-honest threat model. In this model, the participating hospitals are assumed to follow the protocol correctly. However, they may still try to learn extra information from the messages they observe [10]. This is a practical starting point for the thesis. It allows the evaluation to focus on whether privacy-preserving logistic regression is feasible in terms of predictive performance, runtime, and communication overhead, before considering stronger and more expensive malicious-security settings [10, 7].

A central challenge in this work is that secure machine learning is not only a privacy problem, but also a numerical and practical engineering problem. Unlike standard plaintext machine learning, secure training in MP-SPDZ relies on fixed-point arithmetic and requires communication between parties for many operations [7]. These factors make secure training sensitive to design choices such as feature scaling, fixed-point precision, learning rate, and training depth [7, 12]. Even small numerical choices can affect convergence, predictive performance, and communication overhead in secure computation [7, 9]. Therefore, this thesis evaluates not only whether secure logistic regression can be implemented, but also how well it performs compared with a plaintext baseline and how much additional runtime and communication overhead it introduces [9].

This thesis contributes a privacy-preserving pipeline for detecting unusual medical ordering behaviour across hospitals. The pipeline includes preprocessing of clinical data, construction of anomaly labels from procedure-related features, horizontal data partitioning between hospitals, secure logistic regression training in MP-SPDZ, and comparison against a plaintext logistic regression baseline. The model is evaluated using accuracy, precision, recall, and F1-score. Practical feasibility is assessed through runtime and communication overhead. Through this evaluation, the thesis studies the trade-off between privacy protection and practical usability in secure healthcare machine learning [9, 7].

1.1 The Situation

Healthcare institutions often operate in isolation when analysing their data, even when they face similar clinical and operational challenges. A single hospital may not have enough data to reliably detect unusual ordering patterns. This can make the resulting model less robust and less generalizable [9]. This is especially problematic for anomaly detection, because unusual or potentially noncompliant ordering behaviour may occur relatively infrequently. Detecting such patterns requires enough data to separate meaningful deviations from random variation.

However, directly gathering patient data from different hospitals is generally infeasible due to significant privacy concerns and the risk of data breaches [1]. Sharing medical data across institutions introduces substantial security and governance risks [1, 6]. As a result, there is an urgent need for privacy-preserving methods that support collaborative analysis without exposing raw patient records.

The thesis considers a collaborative setting in which multiple hospitals each hold their own local dataset of patient records. Each record is represented as a feature vector containing covariates and procedure-related variables. The aim is to identify records whose ordering patterns deviate statistically from normal practice. To achieve this in a privacy-preserving manner, the participating parties jointly train a logistic regression model using secure multi-party computation.

The central research question of this thesis is:

> *How can an MPC protocol for training logistic regression in the semi-honest setting be designed to enable cross-hospital detection of guideline noncompliant test and procedure ordering, while remaining practical in terms of predictive accuracy and computational and communication overhead?*

To answer this question, the thesis develops and evaluates a secure training pipeline for logistic regression. The focus is not only on whether the model can be trained securely, but also on whether the approach remains practical. This is important because MPC can add substantial runtime and communication overhead compared with standard machine learning.

1.2 Thesis Overview

The remainder of this thesis is structured as follows. Chapter 2 discusses relevant literature on privacy-preserving machine learning and healthcare applications. The necessary background on anomaly detection, logistic regression, and secure multi-party computation is introduced in Chapter 3. Chapter 4 presents the methodology and describes how the proposed MPC-based logistic regression approach is designed and implemented. The experimental setup and evaluation are discussed in Chapter 5, with a focus on predictive accuracy, runtime, and communication overhead. Chapter 6 outlines directions for future research. Finally, Chapter 7 concludes the thesis.

2 Related Work

This section reviews work related to the main parts of this thesis. It starts with clinical logistic regression, clinical decision support, and distance-based outlier detection. It then moves to privacy-preserving logistic regression and secure machine learning methods that are closer to the MPC setting studied here.

Zabor et al. [17] describe logistic regression as a standard method for modelling binary outcomes in clinical studies. They highlight that the model is useful because its results can be interpreted through odds ratios. At the same time, they stress that logistic regression must be specified carefully. Too many parameters or poorly chosen predictors can make the results less reliable. This work is relevant to the present thesis because it supports the use of logistic regression as a clinically meaningful and interpretable model, even when the training process is moved to a privacy-preserving MPC setting.

Montani and Striani [13] discuss the role of artificial intelligence in clinical decision support. Their work shows how AI can support clinical reasoning and help identify decisions that may deviate from expected practice. This connects to the motivation behind the anomaly-detection task. In this thesis, unusual combinations of tests, treatments, and procedures are treated as possible signals of atypical medical ordering behaviour. The paper does not focus on secure multi-party computation or distance-based anomaly detection directly. However, it provides clinical motivation for studying unusual ordering patterns in healthcare data.

Knorr and Ng [8] introduce distance-based outliers as a way to identify observations that differ strongly from the rest of a dataset. In their formulation, an object can be considered an outlier when it does not have enough neighbouring objects within a specified distance threshold. This idea is relevant to the present thesis because the anomaly labels are constructed using distances from the cohort norm. More specifically, each patient's procedure-related feature profile is compared with the average procedure profile of the cohort. Records with the largest distance scores are then labeled as anomalous. Knorr and Ng therefore provide the methodological basis for the distance-based labeling strategy used during preprocessing.

Mohassel and Zhang introduce SecureML [12], a system for scalable privacy-preserving machine learning. Their framework supports secure training for linear regression, logistic regression, and neural networks in a two-server setting. In this setting, data owners distribute their private records as secret shares between two non-colluding servers, which then jointly train the model without learning the raw data [12]. A key contribution of SecureML is its efficient handling of arithmetic on shared decimal values. Instead of relying only on expensive Boolean circuits, the system represents decimal values as shared integers in a finite field and uses secure multiplication with truncation to support fixed-point computation. This is relevant to the present thesis because secure logistic regression in MP-SPDZ also depends on fixed-point arithmetic and efficient secure arithmetic operations. SecureML also addresses the difficulty of nonlinear functions such as the sigmoid function, which is important for logistic regression. The authors propose MPC-friendly alternatives and use conversions between arithmetic sharing and Yao sharing to evaluate such operations more efficiently [12]. This work is closely related to the present thesis because both study secure logistic regression. It also highlights the same practical trade-off between privacy, numerical representation, and computational efficiency.

Shi et al. introduce SMAC-GLORE [16], an MPC-based framework for logistic regression across multiple institutions. It is designed for horizontally partitioned patient data. In this setting, each institution keeps its own patient records, but still contributes to one shared logistic regression model [16]. SMAC-GLORE protects more than only the raw patient data. It also protects intermediate values produced during model learning, such as score vectors and information matrices. This is important because intermediate values may still reveal sensitive information if they are shared in plaintext. The system uses a circuit-based MPC approach based on the GMW protocol [16]. This means that the computation is represented as a circuit and securely evaluated by the participating parties. SMAC-GLORE releases only the final learned model coefficients. This work is relevant to the present thesis because it addresses a similar healthcare setting: collaborative logistic regression without centralizing patient data. However, the technical approach is different. SMAC-GLORE relies on Boolean-circuit-based secure computation, while this thesis uses MP-SPDZ with secret-shared arithmetic. For that reason, SMAC-GLORE is mainly used as related work that shows how secure logistic regression has been applied in biomedical research, rather than as a direct experimental baseline.

Ghavamipour et al. [4] propose privacy-preserving protocols for training logistic regression models using secret-sharing-based Secure Multi-Party Computation. Their work focuses on medical classification tasks where data is spread across different institutions. Combining these datasets can make the analysis stronger, but it also raises privacy concerns. Their method estimates the model parameters with the Newton–Raphson method. The authors consider both honest-majority and dishonest-majority settings. In an honest-majority setting, most parties are assumed not to be corrupted. In a dishonest-majority setting, corrupted parties may form a majority. They also address the difficulty of nonlinear operations in logistic regression by considering both accurate sigmoid computation and a more efficient polynomial approximation. The protocols are evaluated on synthetic and real-world datasets and compared with ordinary logistic regression in terms of accuracy and efficiency. This work is relevant to the present thesis because it also studies secure logistic regression on horizontally distributed data, with a strong focus on medical data and secret sharing. It provides a useful comparison point for the MP-SPDZ-based implementation in this thesis, especially when discussing optimisation choices, sigmoid handling, and the trade-off between security assumptions and practical performance.

Liu et al. [9] propose an online-efficient protocol for privacy-preserving logistic regression based on function secret sharing (FSS). Their approach uses two non-colluding servers and a third-party dealer that provides correlated randomness. The main focus of the paper is reducing online-phase cost. This is important in practical deployments, where latency and communication overhead can become major bottlenecks. The authors also introduce MPC-friendly sigmoid approximations and compare their method with secret-sharing baselines, including MP-SPDZ-based implementations. Their results show improvements in online runtime and communication overhead. This paper is closely related to the present thesis because it studies secure logistic regression training and highlights the trade-off between privacy, efficiency, and practical deployment [9]. Although their FSS-based approach reports strong online efficiency, this thesis uses MP-SPDZ for a different reason. MP-SPDZ provides a practical framework for running different built-in secret-sharing protocols, such as `semi2k` and Shamir. It also reports runtime and communication costs, which are needed for the experimental evaluation [7].

The paper *Secure and Efficient Logistic Regression with Secret-Sharing MPC and Differential Privacy* proposes a privacy-preserving logistic regression method that combines secret-sharing-based MPC with differential privacy [11]. The approach uses a three-party MPC protocol with mini-batch gradient descent to reduce computational and communication overhead during training. After secure training, differential privacy is applied by adding noise to the gradients. This helps protect the final model against attacks such as membership inference and model inversion. Their paper studies secure logistic regression under practical efficiency constraints, while also considering privacy risks after training. The present thesis does not apply differential privacy, but the paper still serves as an important comparison point for MPC-based logistic regression approaches that focus on confidentiality during computation.

2.1 Position of this Thesis

The works discussed above show that privacy-preserving logistic regression is an active research topic, especially when sensitive data cannot be shared directly between institutions [4, 16]. Clinical logistic regression literature supports the choice of an interpretable binary classifier [17]. Clinical decision-support work motivates the focus on unusual ordering behaviour as a signal for possible deviations from expected practice [13]. Distance-based outlier detection provides the methodological basis for constructing anomaly labels from records that deviate from the cohort norm [8]. The SecureML system shows that logistic regression and other machine learning models can be trained securely using secret sharing and fixed-point arithmetic. It also discusses MPC-friendly handling of nonlinear functions, which is important for logistic regression [12]. SMAC-GLORE adds a healthcare-specific reference point, since it applies secure multi-party logistic regression to horizontally partitioned patient data across multiple institutions [16]. Logistic regression based on secret sharing has also been studied for medical data analysis. This work considers different security settings and optimisation choices [4]. Prior work has also proposed efficient secure logistic regression protocols and improved online-phase performance. Some approaches explore additional privacy protections, such as differential privacy [9, 11]. Together, these works provide the background for the main design choices in this thesis, including the use of logistic regression, secret-shared computation, fixed-point arithmetic, and the evaluation of runtime and communication overhead.

This thesis builds on these ideas, but applies them in a cross-hospital anomaly-detection setting. The focus is not only on the secure logistic regression protocol itself. Instead, the thesis evaluates a complete pipeline. This pipeline includes preprocessing clinical data, constructing distance-based anomaly labels from procedure-related features, horizontally partitioning the data between hospitals, training logistic regression securely in MP-SPDZ, and comparing the secure model with a plaintext baseline. The related work therefore informs both the technical and practical parts of the thesis. SecureML and related secure logistic regression work motivate the use of MPC-friendly arithmetic and attention to the sigmoid function [12, 9]. Clinical logistic regression literature supports the choice of an interpretable binary classifier [17]. Clinical decision-support work motivates the focus on unusual ordering behaviour as a signal for possible deviations from expected practice [13]. Distance-based outlier detection provides the methodological basis for constructing anomaly labels from records that deviate from the cohort norm [8]. Together, the related work connects the privacy-preserving training method with the healthcare task studied here. The thesis then evaluates whether secure cross-hospital anomaly detection can achieve useful prediction quality while remaining practical in terms of runtime and communication overhead.

3 Background

This chapter introduces the main concepts required for the rest of the thesis. It begins with secure multi-party computation (MPC) and explains why it is relevant for privacy-preserving collaboration. Next, it discusses secret sharing, including Shamir's secret sharing as one example used in MPC. The chapter then presents the threat model considered in this thesis, with particular attention to the semi-honest setting. Finally, the chapter introduces logistic regression, a statistical method for binary classification in clinical studies. Logistic regression serves as the primary model for detecting statistically unusual patterns in medical ordering behaviour.

3.1 MPC

Secure multi-party computation (MPC) is a cryptographic approach that allows multiple parties to jointly compute a function over their private inputs. The goal is to reveal only the intended output of the computation, not the private inputs themselves [2, 18]. The participants can obtain a shared result without directly disclosing their raw data to one another. This makes MPC especially useful in settings where data is sensitive and cannot easily be centralized.[2].

The central idea behind MPC is that data is not processed in plaintext by a trusted central authority. Instead, the participating parties carry out the computation together. Each party keeps its own data locally, but still contributes to the shared computation [2]. Different MPC protocols achieve this in different ways. The focus of this thesis is on protocols based on secret sharing. Such protocols are analysed under a specific adversarial model, such as the semi-honest, malicious, or covert setting [18].

Hospitals may want to train a shared machine learning model without directly sharing patient-level data [2, 9]. MPC fits the cross-hospital setting considered in this thesis. MP-SPDZ is used as the implementation framework because it supports the development and benchmarking of MPC protocols under different assumptions [7]. This thesis also makes practical use of the `semi2k` protocol within MP-SPDZ. This protocol is based on additive secret sharing over a ring and is used as the main semi-honest baseline for the implementation [7]. A ring is a mathematical structure that supports addition, subtraction, and multiplication within the same value space. Logistic regression training mainly consists of arithmetic operations, such as additions, multiplications, and parameter updates. These operations fit naturally with the arithmetic protocols supported by MP-SPDZ.

Why MP-SPDZ was chosen. MP-SPDZ was selected as the implementation framework because it provides a practical and flexible environment for experimenting with secure multi-party computation in machine learning settings [7, 9]. In particular, it supports multiple MPC protocol families within a single framework. This makes it possible to compare different secure computation approaches without having to reimplement the entire learning pipeline [7]. This was especially useful in the present thesis, where both the `semi2k` protocol and a Shamir-based configuration were considered. MP-SPDZ also offers built-in support for secure arithmetic types, fixed-point computation, and runtime and communication reporting, all of which are directly relevant to evaluating the practicality of privacy-preserving logistic regression training [7].

Why MPC was chosen instead of FHE. Another method considered was fully homomorphic encryption (FHE). Computations are directly carried out on encrypted data [18]. FHE is useful when the computation is outsourced to another party. In that setting, the external party can perform the computation without seeing plaintext data [18]. That is not the main setting of this thesis. The goal here is not to let one external party compute on encrypted hospital data, but to let multiple hospitals take part in the same training process. MPC fits this collaboration model more directly. Each hospital can contribute to the computation while keeping its own raw data private [18, 2]. This also makes MPC suitable for studying the practical cost of secure training, including runtime and communication overhead [9, 10].

3.1.1 Secret Sharing

Secret sharing is an important technique used in many MPC protocols [7, 18]. It works by splitting a private value into several pieces, called shares. These shares are distributed among the participating parties, and a single share does not reveal the original value by itself. The secret can only be reconstructed when a sufficient number of shares are combined [15].

One common form is additive secret sharing. The secret is split into shares that add up to the original value within a chosen arithmetic space [7, 18]. In the `semi2k` setting used in this thesis, this arithmetic is performed over a ring. This is suitable for the implementation because the logistic regression training mainly uses arithmetic operations on fixed-point values, such as additions, multiplications, and parameter updates [7]. Additive secret sharing is simple, parties can compute on shares without directly seeing the underlying private values.

Figure 1 shows a simplified example of additive secret sharing.

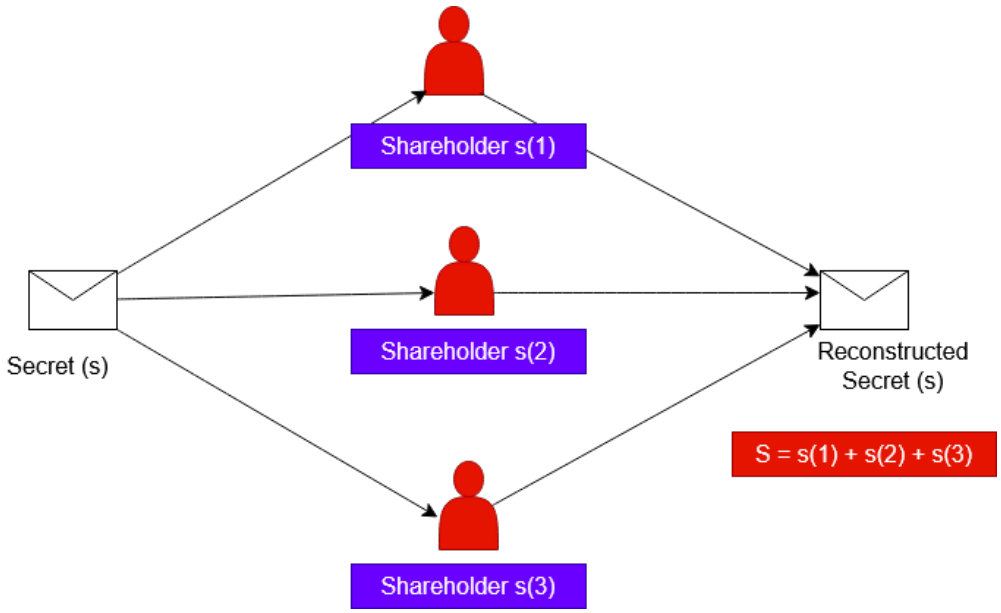

Figure 1: Illustration of additive secret sharing.

3.1.2 Shamir's Secret Sharing

An important example of this type of secret sharing scheme is *Shamir's secret sharing*. The secret is split into n shares. At least k shares are needed to reconstruct it, which is why the scheme is usually written as a (k, n) scheme. If fewer than k shares are available, the secret remains hidden [15].

The main idea is based on polynomials. The secret is placed as the constant term of a randomly chosen polynomial of degree $k - 1$. Each participant then receives one point on this polynomial as their share. Since a polynomial of degree $k - 1$ is uniquely determined by k points, at least k shares are required to recover the original secret. This also means that fewer than k shares are not enough to reconstruct the polynomial or learn any information about the secret [15].

Shamir's secret sharing is important in the context of MPC because it provides a natural way to represent private data in distributed form [18, 7]. Computation can then be performed on the shares rather than on the underlying secret values directly [7, 2]. Many MPC protocols based on honest-majority assumptions, such as those implemented in the MP-SPDZ framework, use variants of polynomial secret sharing derived from this idea [18, 7].

3.2 Threat Model

The security guarantees of an MPC protocol depend on the threat model. This model describes what kind of behaviour is expected from corrupted parties [7, 18]. In secure computation, two common cases are the semi-honest model and the malicious model [10, 7]. In the semi-honest model, corrupted parties still follow the protocol, but may try to infer extra information from the messages they observe [10, 18]. In the malicious model, corrupted parties may deviate from the protocol to learn private information or influence the result [10, 18].

The semi-honest model is often used as a practical baseline because the secure protocols in this setting are generally more efficient than those that are secure against malicious adversaries [10, 18]. This efficiency difference is especially relevant in privacy-preserving machine learning, where malicious security often introduces "huge communication and/or computation costs" [10]. For example, malicious security can significantly increase the bit-communication required for basic operations like multiplication compared to semi-honest variants [10]. At the same time, the malicious model provides stronger assurances, as it defends against participants who behave in an actively adversarial manner. [18, 10].

The present work focuses on the semi-honest threat model. The main reason for this choice is practicality. The goal is to evaluate whether privacy-preserving logistic regression training is feasible in terms of runtime, communication cost, and predictive performance in a cross-hospital setting [10]. The semi-honest model therefore provides a suitable starting point for implementation and evaluation, while stronger security models can be discussed as possible extensions [10]. This is also consistent with MP-SPDZ, which explicitly supports both semi-honest/passive and malicious/active protocol variants for benchmarking different trade-offs [7].

In an honest-majority setting, more than half of the participating parties are assumed to behave honestly. This assumption often allows more efficient MPC protocols, because the protocol can rely on the fact that a majority of parties are not corrupted [7, 18]. In a dishonest-majority setting, corrupted parties may form a majority. This provides stronger protection against collusion, but it usually requires more expensive cryptographic techniques and can increase runtime or communication cost [18, 10]. This distinction is relevant for the protocols considered in this thesis, because `semi2k` is used in a two-party setting, while the Shamir-based configuration follows an honest-majority setting with three parties.

3.3 Logistic Regression

Logistic regression is a statistical classification method. It is used to model the probability of a binary outcome as a function of one or more independent variables [17]. It is one of the most widely used approaches for binary prediction problems, including clinical prediction tasks and machine learning algorithms [17, 9]. Logistic regression does not output a continuous prediction directly. Instead, it uses the sigmoid function to turn a linear combination of the input features into a probability between 0 and 1 [17].

Given an input vector $x = (x_1, x_2, \ldots, x_p)$, logistic regression first computes a linear score

$$z = b + \sum_{j=1}^{p} w_j x_j,$$

where $w = (w_1, w_2, \ldots, w_p)$ is the weight vector and b is the bias term. This score is then transformed into a probability through the sigmoid function:

$$P(y = 1 \mid x) = \sigma(z) = \sigma(w^T x + b) = \frac{1}{1 + e^{-(w^T x + b)}}.$$

Here, $\sigma(\cdot)$ denotes the sigmoid function. The model parameters are usually estimated by minimizing a loss function based on the training data. This is often done with gradient-based optimisation methods [17].

Logistic regression is used in this thesis because it matches the anomaly-detection task in a practical way. The model is designed for binary classification. This matches the goal of predicting whether a patient record should be flagged as anomalous or non-anomalous [17, 9]. It is also more interpretable than many complex "black-box" machine learning models. This matters in healthcare, where understanding the relationship between predictors and outcomes is often just as important as the prediction itself [17, 13]. Logistic regression is also a practical choice for secure computation because most of its training can be expressed with arithmetic operations on shared values [12]. However, the sigmoid function remains challenging in MPC because it is nonlinear and may require approximation [9]. In addition, logistic regression remains a standard and well-established method in clinical prediction modelling [17, 13].

4 Approach

The approach consists of a privacy-preserving pipeline for detecting anomalies across hospitals. It first defines the system model. It then explains how anomaly labels are constructed from procedure-related features, how logistic regression is trained securely, and how MP-SPDZ is used to implement the secure computation.

4.1 System Model

This thesis considers a cross-hospital collaboration setting. Multiple hospitals each hold their own local patient records. They each wish to jointly train a predictive model without revealing raw data to one another. The system is therefore designed as a horizontally partitioned learning setting. Each party contributes different patient records. But they use the same set of features and jointly train a single model.

At a conceptual level, the system has three main stages. First, the original patient data is preprocessed into a feature representation suitable for secure computation. Then, an anomaly label is created from the procedure-related variables to identify records with unusual treatment or procedure patterns. Finally, the hospitals jointly train a logistic regression model on the labeled data without sharing their raw patient records.

The feature vector used for training contains 34 variables in total, containing 9 covariates and 25 procedure-related features. The target variable is an anomaly label stored as `ANOMALY_FLAG`. The predictive task of the system is binary classification. Each patient record is represented as a feature vector. For each patient record, the model receives the selected features as input. It then estimates the probability that the record is anomalous. A decision threshold is used to turn this probability into a final label.

The secure implementation is carried out in MP-SPDZ under a semi-honest threat model. The main setting uses two parties, referred to as Hospital A and Hospital B. Each hospital contributes its own local patient records to the secure training process. The secure computation is carried out between two data-contributing parties. Therefore, the runtime and communication results in the `semi2k` experiments reflect a two-party setup.

A Shamir-based configuration is also considered. In MP-SPDZ, the Shamir protocol used in this thesis follows a three-party honest-majority setting [7]. This means that at least three parties are needed, so that a majority of the parties can remain honest [18, 7]. For this reason, a third participant, Hospital C, is introduced for the Shamir-based experiments. Hospital C does not contribute additional patient records. It is included only to satisfy the protocol structure and to allow comparison with the two-party `semi2k` setting. Because the two protocols use different numbers of parties, their runtime and communication results are not affected only by the protocol arithmetic. They are also affected by the different party structure [7].

4.2 Anomaly Detection

The anomaly detection problem is treated as a supervised binary classification task. Rather than relying on pre-existing anomaly labels in the dataset, the preprocessing pipeline constructs the anomaly target directly from the procedure-related variables. The implemented method uses distance-based anomaly detection on the 25 procedure features. The demographic covariates are not used for constructing the anomaly label. This choice is intentional. The aim is to detect unusual treatment or procedure behaviour, not demographic differences between patients.
For each patient, the procedure-related features are compared with the average procedure profile of the cohort. This average profile represents the typical treatment and procedure pattern in the selected dataset. The method then calculates how far each patient is from this average profile. This is done using the squared distance over the 25 procedure features. The distance score is divided by the average distance. This makes the scores easier to compare between patient records. Patients with a score above the selected percentile threshold are labeled as anomalous.

The resulting anomaly label is not an externally annotated diagnosis of noncompliance. Instead, it is an operational label. It identifies records whose treatment and procedure patterns deviate strongly from the cohort norm. In the implemented version, the anomaly percentile was set to 20%. This means that the top 20% of records by distance score were labeled as anomalous. This threshold was chosen as a practical design decision. An earlier 5% setting produced too few positive examples, which made effective learning difficult for the implemented logistic regression model.

This setup fits the objective of the thesis. The purpose is not to predict a conventional clinical outcome. The aim is to identify unusual combinations of tests, treatments, and procedures. These combinations may indicate possible guideline noncompliance or atypical ordering behaviour. The constructed anomaly label makes it possible to train a classifier on these clinical variables.

Why distance-based anomaly detection was chosen. Distance-based anomaly detection was selected because the dataset does not contain direct labels for guideline noncompliance or anomalous ordering behaviour. This is a common challenge in clinical AI, where manual knowledge acquisition and formalization can become a development bottleneck [13]. Since no dedicated labels were available, an operational anomaly label had to be constructed before the problem could be treated as supervised binary classification [17]. A distance-based strategy was suitable for this purpose. It identifies records that are far from the cohort norm in the procedure-feature space. This follows the logic of distance-based outlier detection. Patient records with procedure profiles far from the rest of the cohort are treated as unusual [8]. In this thesis, the method identifies unusual treatment and procedure behaviour. These patterns may point to deviations from expected clinical practice [13]. The resulting labels can then be learned by a logistic regression model in the secure computation setting [9].

4.3 Secure Logistic Regression

Logistic regression is used as the predictive model for the anomaly-detection task defined during preprocessing. After the anomaly labels have been constructed through the distance-based labeling procedure, the problem becomes a supervised binary classification task. Each patient record must then be classified as anomalous or non-anomalous [17]. Logistic regression is then used to learn the relationship between the 34-dimensional patient feature vector and this binary anomaly label. In this sense, anomaly detection is the application objective, while logistic regression is the classification method used to achieve that objective in a secure computation setting [9].

The anomaly-detection procedure defines the target labels. Logistic regression then learns a decision boundary based on the available patient features. This boundary separates anomalous records from non-anomalous records [17]. The model does not discover anomalies in a fully unsupervised way. It is trained on labels that were already constructed during preprocessing, and it learns to reproduce this labeling scheme in a supervised manner.

In the implemented secure version, logistic regression is trained manually in MP-SPDZ using the anomaly labels generated during preprocessing. The model parameters consist of 34 feature weights and one scalar bias term. These parameters are initialized to zero before training. During training, the program computes a prediction for each patient record. It then compares this prediction with the corresponding anomaly label and accumulates gradients across all samples. These gradients are used to update the model parameters over multiple epochs. Each epoch corresponds to one full pass through all training records. Different training configurations are evaluated in the experiments, including variations in the number of epochs and the learning rate.

The learning process is carried out securely because all feature values and labels are represented as secret-shared values inside MP-SPDZ. The feature matrix is stored as an `sfix.Matrix`. In MP-SPDZ, `sfix` represents secret-shared fixed-point values, which are suitable for scaled numerical features. The labels are stored as a `sint.Array`, where `sint` represents secret-shared integers. This is suitable for the binary anomaly labels, which take the values 0 and 1. As a result, the participating hospitals can jointly train the model without revealing their raw patient records to one another. This makes logistic regression suitable not only because it matches the binary classification task, but also because it can be implemented in a relatively direct way within the secure computation framework [9, 7].

After training, the model is evaluated by converting predicted probabilities into binary decisions and computing the confusion matrix together with accuracy, precision, recall, and F1-score. A decision threshold is used for this binary classification step. Although 0.5 is a common default threshold, the final implementation uses a threshold of 0.456 after threshold tuning during the experiments. This lower threshold was selected to improve the balance between detecting anomalous records and limiting false positives. To provide a reference for comparison, the preprocessing script also trains a plaintext logistic regression baseline in scikit-learn on the same processed dataset.

4.4 MP-SPDZ

The secure implementation in this thesis is carried out using the MP-SPDZ framework [7]. MP-SPDZ is used to express logistic regression training and evaluation in a secure computation setting. In this workflow, each party provides its data in secret-shared form, while the framework executes the required arithmetic operations jointly across the parties. Within the present system, MP-SPDZ serves as the practical environment for privacy-preserving training. It supports the secure operations required by logistic regression, including computations on feature values, parameter updates, and evaluation of the trained model [9]. It also reports runtime and communication overhead, which are needed to evaluate the practical feasibility of the approach [7].

5 Experiments

To support reproducibility, the source code used for preprocessing and secure logistic regression training is available in a public GitHub repository:

https://github.com/mikesiswo/secure-cross-hospital-logistic-regression-mpc.

5.1 Setup

The experiments were conducted in a Linux virtual machine environment, which was the platform ultimately used to successfully compile and run the MP-SPDZ framework for this project. The secure computation was carried out under the semi-honest threat model using the `semi2k` and `shamir` protocols.

In all secure computation experiments, the dataset was horizontally partitioned to represent a cross-hospital collaboration setting. The main data-contributing parties represent two hospitals, with each party contributing 800 samples. This results in a total of 1600 patient records. For the `semi2k` configuration, the data is distributed between these two parties. For the Shamir-based configuration, the same horizontally partitioned dataset is used, but an additional third party is included to match the three-party honest-majority structure required by this protocol family in MP-SPDZ [7]. This third party does not contribute additional patient rows. The feature vector contains 34 variables in total. The first secure training configuration used 50 epochs with a learning rate of 0.01. To support fixed-point computation in MP-SPDZ, the input features were scaled before being written to the player input files.

The experiments evaluated the model from two perspectives. Predictive performance was measured using the confusion matrix, accuracy, precision, recall, and F1-score. Practical overhead was assessed using runtime and communication statistics reported by MP-SPDZ. The secure implementation was also compared with a plaintext scikit-learn baseline to provide a non-secure reference for performance and cost.

5.2 Dataset

The experiments in this thesis use the Myocardial Infarction Complications dataset [5]. This dataset contains patient records related to myocardial infarction and associated complications, and it provides a suitable basis for studying predictive modelling in a healthcare setting.

For the purposes of this thesis, the dataset was not used directly in its original form. A smaller set of variables were selected for the secure logistic regression model. In total, 34 features were used. These consisted of 9 covariates and 25 procedure-related variables. These features were converted to numeric values. Missing values were replaced with the mean of the corresponding column, after which the values were rounded to integers.

To make the data suitable for the anomaly-detection setting considered in this work, a binary anomaly label was constructed during preprocessing. This label was based on the distance between each patient's procedure-related feature vector and the mean procedure profile of the cohort. Records with distance scores in the top 20% were labeled as anomalous, while the remaining records were labeled as non-anomalous. After preprocessing and labeling, 1600 samples were selected and divided equally across the two parties in the secure computation setting.

Why this dataset was chosen. The Myocardial Infarction Complications dataset was chosen because it provides a realistic healthcare setting for studying predictive modelling on sensitive patient data [5]. The dataset is publicly available through the UCI Machine Learning Repository and can be reused with proper citation. It does not contain record identifiers that can be linked back to individual patients. It includes treatment- and procedure-related information, which makes it suitable for constructing feature representations of medical ordering behaviour. This is important for the present thesis, because the objective is not simply to predict a conventional clinical endpoint. The aim is to detect statistically unusual combinations of tests, treatments, and procedures. The dataset also contains missing values and heterogeneous clinical variables, making it more realistic than a highly simplified benchmark dataset. For these reasons, it serves as a suitable case study for cross-hospital anomaly detection under secure computation.

5.3 Results and Discussion

This section presents the results of the secure logistic regression experiments. The results are organized around the main design choices that affected the implementation: preprocessing, training depth, fixed-point precision, and protocol choice. This makes it possible to evaluate the final model performance. It also helps explain why certain configurations were more stable or practical than others.

5.3.1 Experimental Setup

The experiments were conducted on a subset of 1,600 patient records derived from the UCI Myocardial Infarction Complications dataset [5]. The data is horizontally partitioned across two hospitals to simulate a cross-institutional collaboration scenario:

- Total samples: 1600
- Hospital A: 800 samples
- Hospital B: 800 samples
- Features: 34 clinical and procedure-related attributes
- Positive class (noncompliant cases): 320 (20%)
- Negative class (compliant cases): 1280 (80%)

The class distribution reflects a realistic imbalance scenario, where noncompliant procedure ordering form a minority but clinically significant subset.

All secure training experiments were performed under the `semi2k` protocol unless otherwise specified. The `semi2k` protocol implements a two-party semi-honest secure computation model using additive secret sharing over a ring.

The following configuration was used for `semi2k` experiments:

- Ring size: 192 bits (`-R 192`)
- Fixed-point precision: $(f = 32, k = 63)$ unless otherwise specified
- Learning rate: $\eta = 0.005$
- Training epochs: 50 and 150 (evaluated comparatively)
- Optimisation method: secure full-batch gradient descent

For the `semi2k` experiments, the option `-R 192` was used to set the ring size. A ring defines the value space in which additions, subtractions, and multiplications are performed. With `-R 192`, secure arithmetic is performed in a 192-bit ring. This gives the computation a large range of possible values. This range helps avoid overflow when fixed-point values and intermediate training results grow during secure logistic regression training [7].

Model parameters were optimised using secure full-batch gradient descent, where gradients were computed over all samples in each epoch using secret-shared fixed-point arithmetic. No intermediate gradients or model parameters were revealed during training.

To evaluate the impact of protocol design, additional experiments were conducted using the `Shamir` protocol, which implements Shamir's secret sharing in a three-party honest-majority setting. In these experiments, a third party (Hospital C) was introduced solely to enable comparison between two-party additive sharing and three-party Shamir-based secret sharing. No additional data samples were assigned to Hospital C.

Unless explicitly stated, all reported results correspond to the `semi2k` protocol with a 192-bit ring configuration.

5.3.2 Initial Secure Training Behaviour

The first secure training experiments were conducted using simple feature scaling, where all input values were divided by a constant factor. The model was trained using secure full-batch gradient descent with fixed-point precision set to $(f = 16, k = 31)$ and 50 training epochs.

Under this configuration, the model completed training but did not converge to a useful decision boundary. The predicted probabilities stayed close to each other, mostly between 0.45 and 0.50. This suggests that the model was not meaningfully separating compliant cases from noncompliant cases.

The predicted probabilities clustered tightly in the range of approximately 0.47–0.48:

```
Prob 0 = 0.474548
Prob 1 = 0.475433
Prob 2 = 0.475586
Prob 3 = 0.475204
Prob 4 = 0.473465
```

This narrow probability distribution indicates that the model learned an almost constant decision boundary, failing to meaningfully separate compliant and noncompliant cases.

The resulting confusion matrix is shown as Table 1:

	Predicted Positive	Predicted Negative
Actual Positive	320 (TP)	0 (FN)
Actual Negative	1280 (FP)	0 (TN)

Table 1: Initial secure training results before proper feature standardization.

The derived performance metrics are presented as Table 2:

Metric	Value
Accuracy	0.2000
Precision	0.2000
Recall	1.0000
F1 Score	0.3334

Table 2: Performance metrics before feature standardization.

Although recall reached 1.0, this result was misleading. The model simply predicted every sample as noncompliant, resulting in zero true negatives. The apparent behaviour demonstrates that the optimisation process collapsed toward a trivial solution rather than learning meaningful separation.

In other early experiments (with slightly different scaling), the model instead predicted only the majority class, producing accuracy values close to 80% but near-zero recall. This further highlights that raw accuracy alone is insufficient for evaluating imbalanced classification problems.

The failure was not caused by protocol incorrectness or cryptographic limitations. Instead, it stemmed from optimisation instability under secure fixed-point arithmetic. The gradients were poorly conditioned due to insufficient feature normalization, preventing effective convergence.

5.3.3 Impact of Feature Standardization

After observing unstable convergence under simple division-based scaling, feature preprocessing was revised. In the previous implementation, all input values were divided by a constant. In the revised version, the features were standardized using `StandardScaler`. This method subtracts the mean of each feature and then divides by its standard deviation, giving each feature zero mean and unit variance before secure training.

This change made the input features better scaled for fixed-point secure arithmetic. In the experiments, this was the decisive preprocessing change that allowed the secure model to learn a meaningful decision boundary.

Unlike the previous configuration where probabilities clustered tightly around 0.45–0.50, the standardized model produced a wider and more meaningful distribution:

```
Prob 0 = 0.269935
Prob 1 = 0.305759
Prob 2 = 0.278290
Prob 3 = 0.316763
Prob 4 = 0.422372
```

The wider spread indicates that the model learned a non-trivial decision boundary and successfully separated compliant and noncompliant cases. Table 3 shows the resulting confusion matrix after feature standardization.

	Predicted Positive	Predicted Negative
Actual Positive	295 (TP)	25 (FN)
Actual Negative	129 (FP)	1151 (TN)

Table 3: Secure training results after feature standardization (150 epochs, high precision).

Classification Results Table 4 reports the classification metrics computed from the confusion matrix.

Metric	Value
Accuracy	0.9038
Precision	0.6958
Recall	0.9219
F1 Score	0.7930

Table 4: Classification performance after feature standardization.

Compared with the initial unstable configuration, the standardized model no longer collapsed into predicting almost all records as the same class. Recall reached approximately 0.92, while the F1 score increased from 0.33 in the initial configuration to approximately 0.79. This shows that the model was not only detecting most noncompliant cases, but also producing a better balance between false positives and false negatives.

Computation and Communication Overhead Table 5 summarizes the computation and communication overhead for the standardized secure training configuration.

Metric	Value
Training time	4573.63 s
Training rounds	40,687,011
Evaluation time	26.92 s
Evaluation rounds	293,786
Total runtime	4600.64 s
Data sent per party	187,918 MB
Global data sent	375,671 MB

Table 5: Secure training overhead after feature standardization.

The dramatic improvement demonstrates that optimisation instability, not cryptographic limitations, was the primary cause of earlier failures. Secure full-batch gradient descent under fixed-point arithmetic is highly sensitive to feature scale. Without proper standardization, the training updates can become unstable and the model may fail to learn a useful decision boundary.

Feature standardization was therefore the decisive fix that enabled stable training, meaningful probability separation, and strong classification performance under secure computation.

Secure gradient-based optimisation requires proper feature conditioning. While floating-point plaintext training can often tolerate suboptimal scaling, secure fixed-point arithmetic amplifies numerical instability. Proper standardization was essential for convergence and constitutes the primary methodological improvement in this work.

5.3.4 Effect of Training Depth (Epochs)

To evaluate the impact of optimisation depth on secure training quality, experiments were conducted using 50 and 150 epochs under identical protocol and precision settings. The 50-epoch setting was used as the initial baseline. It provided a reasonable starting point for full-batch gradient descent, while keeping secure runtime manageable. After this setting showed unstable behaviour and many false positives, the number of epochs was increased to 150. This tested whether additional training iterations would improve convergence. Both configurations used the `semi2k` protocol, ring size `-R 192`, and fixed-point precision ($f = 32, k = 63$).

50 Epochs Table 6 shows the confusion matrix after 50 epochs of secure training.

	Predicted Positive	Predicted Negative
Actual Positive	320 (TP)	0 (FN)
Actual Negative	721 (FP)	559 (TN)

Table 6: Secure training results after 50 epochs.

Table 7 reports the corresponding performance metrics for the 50-epoch configuration.

Metric	Value
Accuracy	0.5494
Precision	0.3074
Recall	1.0000
F1 Score	0.4702

Table 7: Performance metrics after 50 epochs.

After 50 epochs, the model achieved recall of 1.0, meaning that no anomalous records were missed. This came at the cost of 721 false positives. The decision boundary was therefore still unstable and biased toward predicting the positive class. Although all noncompliant cases were identified, the excessive number of false alarms reduced precision and overall model reliability.

The total secure runtime for this configuration was 1,592.89 seconds, with 62,321 MB communicated per party.

150 Epochs Table 8 shows the confusion matrix after increasing the training depth to 150 epochs.

	Predicted Positive	Predicted Negative
Actual Positive	295 (TP)	25 (FN)
Actual Negative	129 (FP)	1151 (TN)

Table 8: Secure training results after 150 epochs.

The performance metrics for the 150-epoch configuration are shown in Table 9.

Metric	Value
Accuracy	0.9038
Precision	0.6958
Recall	0.9219
F1 Score	0.7930

Table 9: Performance metrics after 150 epochs.

With 150 epochs, the classifier became less biased toward the positive class. False positives decreased from 721 to 129, while recall stayed high at approximately 0.92. The F1 score increased from 0.47 to 0.79, which shows that the improvement was not limited to accuracy alone.

This improvement came at a computational cost. Total runtime increased to 4,626.45 seconds, and communication per party increased to 187,918 MB. This reflects the relationship between training iterations and communication complexity under secure computation.

The comparison shows that optimisation depth strongly affects convergence under secure full-batch gradient descent. With only 50 epochs, the model still predicted too many records as noncompliant. With 150 epochs, the additional training iterations reduced this bias and produced a more balanced classifier.

The plaintext reference model was trained with L-BFGS optimisation. L-BFGS, short for limited-memory Broyden–Fletcher–Goldfarb–Shanno, is a quasi-Newton method. It uses an approximation of curvature information to guide the parameter updates, while standard gradient descent only uses first-order gradient information [14]. Because of this, L-BFGS can often reach a stable solution in fewer iterations than basic gradient descent [14]. The secure implementation used full-batch gradient descent. This helps explain why the number of epochs had such a strong effect on convergence and predictive performance [12, 4].

Training depth is therefore an important factor in secure learning. In this experiment, 50 epochs were not enough for the model to settle into a reliable decision boundary. Increasing the number of epochs improved predictive quality. However, it also increased runtime and communication cost.

5.3.5 Effect of Fixed-Point Precision

To evaluate the influence of numerical precision on secure training stability, experiments were conducted using two fixed-point configurations:

- Low precision: $(f = 16, k = 31)$
- High precision: $(f = 32, k = 63)$

All other parameters were held constant. Both configurations used 150 epochs, the `semi2k` protocol, and ring size $-R$ 192.

High Precision: $(32, 63)$ Table 10 shows the confusion matrix for the high-precision configuration.

	Predicted Positive	Predicted Negative
Actual Positive	295 (TP)	25 (FN)
Actual Negative	129 (FP)	1151 (TN)

Table 10: Secure training results with high precision $(32, 63)$.

Table 11 reports the corresponding performance metrics for the high-precision configuration.

Metric	Value
Accuracy	0.9038
Precision	0.6958
Recall	0.9219
F1 Score	0.7930

Table 11: Performance metrics with high precision $(32, 63)$.

The total runtime for the high-precision configuration was 4,626.45 seconds, with 187,918 MB communicated per party.

Lower Precision: $(16, 31)$ Table 12 shows the confusion matrix for the lower-precision configuration.

	Predicted Positive	Predicted Negative
Actual Positive	305 (TP)	15 (FN)
Actual Negative	207 (FP)	1073 (TN)

Table 12: Secure training results with lower precision $(16, 31)$.

Table 13 reports the performance metrics for the lower-precision configuration.

Metric	Value
Accuracy	0.8613
Precision	0.5957
Recall	0.9531
F1 Score	0.7331

Table 13: Performance metrics with lower precision $(16, 31)$.

The runtime decreased to 3,841.89 seconds, and communication per party decreased to 154,582 MB.

Comparing Table 12 with Table 10 shows that increasing fixed-point precision from $(16, 31)$ to $(32, 63)$ reduced false positives from 207 to 129. This improved precision from 0.60 to 0.70 and increased the F1 score from 0.73 to 0.79, as shown in Table 13 and Table 11. Recall decreased slightly from 0.95 to 0.92, which suggests that higher precision produced a more conservative but more stable decision boundary.

The class separability itself did not fundamentally change between the two configurations. Instead, fixed-point precision mainly influenced boundary stability and gradient accumulation accuracy under secure arithmetic.

This improvement came at a substantial cost. Increasing precision raised runtime by approximately 20% and increased communication volume by more than 30 GB per party. This reflects the fact that higher precision requires larger numeric representations and more expensive truncation operations.

Fixed-point precision affects numerical stability in secure gradient descent. Higher precision helped refine the decision boundary and reduced false positives. But it also increased computation and communication overhead. This shows that improving numerical stability can make secure training more expensive in practice.

5.3.6 Comparison with Plaintext Baseline

To evaluate the practical impact of secure computation on predictive performance, the secure logistic regression model was compared with a plaintext baseline implemented using `scikit-learn`. The plaintext model was trained on the same standardized dataset using the L-BFGS optimiser.

Plaintext Baseline Performance Table 14 shows the confusion matrix for the plaintext logistic regression baseline.

	Predicted Positive	Predicted Negative
Actual Positive	274 (TP)	46 (FN)
Actual Negative	26 (FP)	1254 (TN)

Table 14: Confusion matrix for plaintext logistic regression.

Table 15 reports the corresponding performance metrics for the plaintext baseline.

Metric	Value
Accuracy	0.95
Precision	0.91
Recall	0.86
F1 Score	0.88
AUC-ROC	0.9878

Table 15: Plaintext logistic regression performance.

The plaintext model achieved strong predictive performance, with an F1 score of approximately 0.88. The AUC-ROC score of 0.9878 also indicates strong separability between anomalous and non-anomalous records, across different classification thresholds.

Secure Model Performance The best secure configuration used 150 epochs and fixed-point precision ($f = 32, k = 63$). Its classification results were reported earlier in Table 8 and Table 9. This configuration achieved an accuracy of approximately 0.90, recall of approximately 0.92, and an F1 score of approximately 0.79.

The secure model therefore retained high recall and correctly identified the majority of anomalous records. Precision was lower than in the plaintext baseline, but the overall classification quality remained strong. This shows that the secure computation setting did not prevent the model from learning a meaningful decision boundary.

Performance Gap Analysis Comparing Table 15 with Table 9 shows that there is still a performance gap between the plaintext and secure models. The plaintext model achieved an F1 score of approximately 0.88, while the best secure model achieved an F1 score of approximately 0.79.

This difference can be explained by several factors. The plaintext model uses L-BFGS, a limited-memory quasi-Newton optimiser that can converge faster than basic gradient descent [14]. The secure implementation, in contrast, relies on first-order full-batch gradient descent. The secure implementation also represents values with fixed-point arithmetic instead of standard floating-point precision. This makes optimisation more sensitive to feature scaling and numerical precision [7, 12].

Importantly, secure computation itself did not prevent high recall. The secure model still achieved recall of approximately 0.92, which shows that meaningful classification boundaries can be learned despite the constraints introduced by MPC.

Overhead Comparison The plaintext baseline required only 4.12 milliseconds for training and operated on 0.448 MB of raw data. In contrast, the best secure configuration required approximately 4,600 seconds of runtime and transmitted approximately 187 GB per party.

This difference shows that the main cost of the secure implementation is not predictive failure, but computational and communication overhead. The runtime and communication volume are several orders of magnitude larger than in the plaintext baseline.

The comparison shows that privacy-preserving cross-hospital anomaly detection is feasible in terms of predictive performance. The best secure configuration achieved an accuracy of around 0.90 and an F1 score around 0.79, while still preserving high recall. However, this performance came with substantial overhead: approximately 4,600 seconds of runtime and 187 GB of communication per party under the `semi2k` protocol. Earlier experiments showed the same trade-off when changing the number of epochs and fixed-point precision. More epochs improved convergence, and higher precision reduced false positives, but both increased runtime and communication cost. The main practical limitation is therefore not that the secure model fails to learn. Rather, better predictive quality requires more computation and communication. In real cross-hospital deployments, feasibility would depend on the communication budget and latency that the participating institutions can accept.

5.3.7 Comparison Between `semi2k` and Shamir

To evaluate the effect of MPC protocol choice, the Shamir-based configuration was compared with the best `semi2k` configuration. The comparison used the same anomaly-detection task, the same 34-feature input representation, and the same classification metrics.

Table 16 compares the predictive performance of both protocols.

Metric	Best `semi2k`	Best Shamir
Accuracy	0.9038	0.8613
Precision	0.6958	0.5957
Recall	0.9219	0.9531
F1 Score	0.7930	0.7332

Table 16: Predictive performance comparison between the best `semi2k` and Shamir configurations.

The Shamir-based configuration achieved slightly higher recall than the best `semi2k` configuration. This means it missed fewer anomalous records. This came with more false positives, which reduced precision from 0.6958 to 0.5957. As a result, the F1 score was lower for Shamir, decreasing from 0.7930 to 0.7332. In this experiment, `semi2k` therefore produced the better overall balance between precision and recall.

Table 17 compares the computation and communication overhead of both protocols.

Metric	Best `semi2k`	Best Shamir
Training time	4573.63 s	11628.9 s
Evaluation time	26.92 s	63.49 s
Total runtime	4600.64 s	11692.6 s
Data sent per party	187,918 MB	177,837 MB
Global data sent	375,671 MB	531,403 MB
Training rounds	40,687,011	24,134,908
Evaluation rounds	293,786	167,857

Table 17: Computation and communication comparison between the best `semi2k` and Shamir configurations.

The overhead results show a different trade-off. The Shamir configuration used slightly less data per party than the best `semi2k` configuration, but it required much more total runtime. Its global communication was also higher because the Shamir setting involved three parties instead of two. The total runtime increased from approximately 4,600 seconds for `semi2k` to approximately 11,700 seconds for Shamir.

The comparison between `semi2k` and Shamir also points to an important direction for further research. In this thesis, the Shamir-based configuration achieved high recall, but it also produced more false positives and required substantially more runtime than the best `semi2k` configuration. The exact cause of this difference was not isolated in the experiments. It may be related to the three-party structure used for Shamir, the additional protocol overhead, differences in secure arithmetic, or the way the model parameters were optimised under that protocol. A useful next step would therefore be to investigate these factors separately. This could show whether Shamir's stronger security assumptions can be combined with better tuning, improved preprocessing, or alternative optimisation settings to obtain a more accurate and practical secure learning system.

6 Future Work

The comparison between `semi2k` and `Shamir` points to an important direction for future research. In this thesis, the Shamir-based configuration achieved high recall, but it also produced more false positives and required substantially more runtime than the best `semi2k` configuration. The exact reason for this difference was not isolated in the experiments. A controlled comparison could help determine whether the observed performance and overhead differences are mainly caused by the three-party setting, stronger security assumptions, numerical precision, protocol overhead, or implementation-specific factors. If Shamir's additional security can be combined with better tuning, improved preprocessing, or alternative optimisation settings, it may become a more practical option for secure healthcare collaboration.

Future work could also explore more efficient optimisation methods for secure logistic regression. This could include mini-batch gradient descent, momentum-based methods, or secure versions of more advanced optimisation techniques. These methods may reduce runtime while preserving model quality. Another useful direction is to investigate better sigmoid approximations or alternative loss functions that are more efficient under secure computation.

The anomaly-labeling method could also be improved. In this thesis, anomaly labels are constructed using a distance-based procedure over procedure-related features. This provides a practical operational label, but it is not the same as expert-confirmed guideline noncompliance. Future work could compare the distance-based labels with clinician annotations, medical guidelines, or alternative anomaly-detection methods. This would make it possible to evaluate whether the detected anomalies are not only statistically unusual, but also clinically meaningful.

7 Conclusions

This thesis investigated how secure multi-party computation can be used to train logistic regression for cross-hospital detection of unusual medical ordering behaviour. The goal was to evaluate whether hospitals could jointly train a predictive model without sharing raw patient records. The model also had to maintain useful classification performance. To study this, a complete pipeline was implemented. It included clinical data preprocessing, distance-based anomaly labeling, horizontal partitioning between hospitals, secure logistic regression training in MP-SPDZ, and comparison with a plaintext logistic regression baseline.

The results show that secure logistic regression is feasible for this anomaly-detection task, but only when the numerical setup is handled carefully. The initial secure training attempts were unstable. In some cases, the model predicted almost all records as anomalous. In other cases, it mainly predicted the majority class. These outcomes showed that accuracy could be misleading on its own. A model could appear accurate while still failing to detect the anomalous records. For that reason, precision, recall, and F1 score were needed to judge the results properly. The early failures were mainly caused by optimisation instability under secure fixed-point arithmetic, rather than by the cryptographic protocol itself.

Feature standardization was the most important improvement. After replacing simple division-based scaling with standardization, the secure model produced a wider probability distribution and learned a more meaningful decision boundary. Under the best `semi2k` configuration, using 150 epochs and fixed-point precision ($f = 32, k = 63$), the model achieved an accuracy of approximately 0.90, precision of approximately 0.70, recall of approximately 0.92, and an F1 score of approximately 0.79. These results show that secure computation did not prevent the model from identifying most anomalous records.

The experiments also showed that training depth and fixed-point precision strongly influenced the final result. Increasing the number of epochs from 50 to 150 improved convergence. It also reduced false positives and produced a better balance between precision and recall. Higher fixed-point precision also improved stability and F1 score, but both changes increased runtime and communication cost. These results show that secure training is not determined by the MPC protocol alone. Numerical choices such as scaling, precision, learning rate, and training depth also have a strong influence on the final model.

Compared with the plaintext scikit-learn baseline, the secure model achieved lower overall performance but preserved high recall. The plaintext baseline reached an F1 score of approximately 0.88, while the best secure configuration reached approximately 0.79. However, the secure implementation was several orders of magnitude more expensive. The best `semi2k` configuration required approximately 4,600 seconds of runtime and 187 GB of communication per party. The main practical limitation is therefore not that the secure model fails to learn, but that secure training introduces substantial computation and communication overhead.

The comparison between `semi2k` and Shamir showed that protocol choice also matters. The best Shamir-based configuration achieved high recall, but it produced more false positives, a lower F1 score, and much longer runtime than the best `semi2k` configuration. In this thesis, `semi2k` therefore provided the better practical balance between predictive performance and secure computation overhead. A more detailed analysis of the Shamir results was outside the scope of this thesis, but it remains an interesting direction for future work. Such work could investigate whether the observed differences were mainly caused by the three-party setup, protocol overhead, numerical settings, or other implementation-specific factors.

The main conclusion is that privacy-preserving cross-hospital anomaly detection with logistic regression is possible, but practical deployment remains challenging. The secure model can learn meaningful decision boundaries and preserve high recall, which is important for detecting unusual medical ordering patterns. At the same time, the runtime and communication overhead are still substantial. In a real cross-hospital setting, feasibility would depend on the available infrastructure, acceptable latency, communication budget, and security requirements of the participating institutions.

References

[1] Karim Abouelmehdi, Abderrahim Beni-Hessane, and Hayat Khaloufi. Big healthcare data: preserving security and privacy. *Journal of big data*, 5(1):1, 2018.

[2] Wajdi Alghamdi, Reda Salama, M Sirija, Ahmed Radie Abbas, and Kholmurodova Dilnoza. Secure multi-party computation for collaborative data analysis. In *E3S Web of Conferences*, volume 399, page 04034. EDP Sciences, 2023.

[3] Flora Cornish, Alex Gillespie, and Tania Zittoun. Collaborative analysis of qualitative data. *Handbook of qualitative data analysis, SAGE, 2013/6//79-93*, 2013.

[4] Ali Reza Ghavamipour, Fatih Turkmen, and Xiaoqian Jiang. Privacy-preserving logistic regression with secret sharing. *BMC medical informatics and decision making*, 22(1):89, 2022.

[5] Shulman V.A. Rossiev D.A. Shesternya P.A. Nikulina S.Yu. Orlova Yu.V. Golovenkin, S.E. and V.F. Voino-Yasenetsky. Myocardial infarction complications. UCI Machine Learning Repository, 2020. DOI: https://doi.org/10.24432/C53P5M.

[6] IT Healthcare. Health information at risk: Successful strategies for healthcare security and privacy.

[7] Marcel Keller. Mp-spdz: A versatile framework for multi-party computation. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 1575–1590, 2020.

[8] Edwin M. Knorr and Raymond T. Ng. Algorithms for mining distance-based outliers in large datasets. In Ashish Gupta, Oded Shmueli, and Jennifer Widom, editors, *VLDB'98, Proceedings of 24rd International Conference on Very Large Data Bases, August 24-27, 1998, New York City, New York, USA*, pages 392–403. Morgan Kaufmann, 1998.

[9] Jing Liu, Jamie Cui, and Cen Chen. Online efficient secure logistic regression based on function secret sharing. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 1597–1606, New York, NY, USA, 2023. Association for Computing Machinery.

[10] Yibiao Lu, Bingsheng Zhang, and Kui Ren. Maliciously secure mpc from semi-honest 2pc in the server-aided model. *IEEE Transactions on Dependable and Secure Computing*, 21(4):3109–3125, 2024.

[11] Seyma Selcan Magara, Ipek Motorcu, Emin Sahin Mektepli, Cem Ata Baykara, Ali Burak Ünal, and Mete Akgün. Secure and efficient logistic regression with secret-sharing mpc and differential privacy. *IEEE Access*, 2025.

[12] Payman Mohassel and Yupeng Zhang. Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 19–38, 2017.

[13] Stefania Montani and Manuel Striani. Artificial intelligence in clinical decision support: a focused literature survey. *Yearbook of Medical Informatics*, 28(1):120–127, 2019.

[14] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 2006.

[15] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, November 1979.

[16] Haoyi Shi, Chao Jiang, Wenrui Dai, Xiaoqian Jiang, Yuzhe Tang, Lucila Ohno-Machado, and Shuang Wang. Secure multi-party computation grid logistic regression (smac-glore). *BMC medical informatics and decision making*, 16(Suppl 3):89, 2016.

[17] Emily C Zabor, Chandana A Reddy, Rahul D Tendulkar, and Sujata Patil. Logistic regression in clinical studies. *International Journal of Radiation Oncology* Biology* Physics*, 112(2):271–277, 2022.

[18] Chuan Zhao, Shengnan Zhao, Minghao Zhao, Zhenxiang Chen, Chong-Zhi Gao, Hongwei Li, and Yu-an Tan. Secure multi-party computation: theory, practice and applications. *Information Sciences*, 476:357–372, 2019.