The University of Hong Kong

School of Computing and Data Science

COMP7705

Project Report

A Dynamic Credit Risk Prediction System Based on Federated Learning

Submitted in partial fulfillment of the requirements for the admission to the degree of Master of Science in Computer Science

By

Huang Ruifan (3036381046)
He Fuzhi (3036382894)
He Yanze (3036274683)
Liu Dahai (3036383044)

Mentor: Dr. Ting, Hing Fung

Date of submission: 01/08/2025

Abstract

This project presents a dynamic credit risk prediction system based on Federated Learning (FL). Credit assessment plays a crucial role in financial institutions' risk management, but data privacy and security concerns pose significant challenges for multi-party collaboration. Federated Learning, with its decentralized approach, addresses these challenges by allowing institutions to collaboratively train models without sharing sensitive data. This research explores the application of both Horizontal Federated Learning (HFL) and Vertical Federated Learning (VFL) to improve the accuracy and robustness of credit risk prediction models using three distinct datasets: German Credit, Loan Data, and MIT Credit Ranking. We implement FL techniques using the Flower framework, comparing the performance of federated models with standalone models. Our results show that FL significantly enhances model performance, improving classification accuracy, F1-score, and AUC. The study demonstrates the potential of FL in financial applications, ensuring privacy while optimizing prediction models. This work contributes to the development of more secure, efficient, and accurate credit risk assessment systems for financial institutions, supporting better decision-making and resource allocation.

Declaration

We, the undersigned, declare that this project report is our own original work and has been prepared in accordance with the guidelines provided by the University of Hong Kong. We have fully acknowledged and cited any sources used in the completion of this project. This report has not been submitted for any other academic award or degree at this or any other institution.

We further declare that all data, research, and analysis presented in this report are accurate to the best of our knowledge. We have adhered to all ethical standards in the conduct of our research and have followed the university's policies on academic integrity.

Acknowledgments

We would like to express our sincere gratitude to our supervisor, Dr. Ting Hing Fung, for his invaluable guidance and unwavering support throughout this project. Dr. Ting not only provided insightful academic advice but also offered patience and encouragement whenever we faced challenges. Working with him has significantly enhanced our research skills and provided us with invaluable learning experiences.

We would also like to extend our thanks to the staff at the Master Programme Office. Their administrative support and assistance throughout the project, especially with the submission of the project report, data handling, and other administrative matters, were crucial in ensuring the smooth completion of our project.

We are deeply grateful to all those who supported us during this journey. Your help has been instrumental in allowing us to complete this research successfully and gain valuable knowledge and experience.

Table of Contents

1 Introduction

Credit assessment is a core aspect of a financial institution's risk management system. First, credit assessment predicts the probability of default by quantitatively analyzing data on a borrower's historical credit history, financial condition and debt level, thus helping banks to identify high-risk customers and take risk-mitigation measures (such as raising interest rates, requesting guarantees, or refusing to extend credit) to significantly reduce the rate of non-performing loans**.** The accumulation of systemic credit risk may trigger a financial crisis, while credit assessment reduces the risk of a market-wide chain reaction of defaults and maintains the overall stability of the financial system by screening high-quality customers and optimizing the allocation of credit resources. Second, according to Basel III and other regulatory requirements, financial institutions are required to calculate risk-weighted assets (RWA) based on the quantitative results of credit assessment (e.g., credit scores, risk ratings) to ensure that capital adequacy ratios are in line with regulatory standards and to avoid compliance risks arising from inadequate capitalization.

At this stage, the application scenarios of credit assessment have expanded from traditional finance to diversified scenarios such as supply chain, government programs and inclusive finance. In addition, behavioral scoring technology dynamically updates credit scores by analyzing users' real-time behavioral data (e.g. credit card repayment habits, e-commerce consumption records).With the advancement of time and technology, and based on massive data, credit assessment can usually replace traditional manual approval through standardized models (e.g., machine learning algorithms), reducing subjective judgment bias and improving decision-making efficiency and fairness. For example, an intelligent approval system can shorten SME loan approval time from weeks to hours. This is driving an era of more comprehensive and diversified credit analysis and assessment.

In the field of credit assessment, relying on a single data source (e.g., traditional credit reports) has significant limitations, as it is difficult to fully capture the dynamic risk characteristics of individuals. Fusion of heterogeneous data from multiple sources has become the core path to improve the comprehensiveness and accuracy of assessment. For example, financial behavior data (bank flows, insurance payment records) can quantify short-term solvency; public affairs data (social security, tax payments) reflect social credit stability; business scenario data (e-commerce transactions, logistics fulfillment) reveal consumption habits and contractual spirit; and alternative data (device usage, social relationships) can be used to mine potential risk associations through machine learning. With the development of digital economy, integrating heterogeneous data from multiple sources has become an inevitable choice to improve the credit assessment system.

Admittedly, the interplay of multi-faceted and multi-dimensional data and algorithmic models does greatly increase the accuracy and efficiency of evaluating a user's credit rating, but the different sources of different data are often accompanied by a series of issues such as data privacy as well as security. Although these data are collected through legal authorization and work together to build a multidimensional credit portrait, there are strict privacy protection requirements for data sharing between different institutions. In most

cases, financial institutions and third-party data providers need to make data "available but not visible" through privacy computing technology to ensure that the original data does not leave the organization. This poses a challenge in credit assessment scenarios.

In credit assessment scenarios, the privacy protection of data sources and model performance requirements form a seemingly contradictory dual requirement. On the one hand, the accuracy of assessment is highly dependent on the cross-validation of multi-dimensional data - banks need consumption data from e-commerce platforms to verify the stability of users' income and expenditure, e-commerce platforms need repayment records from financial institutions to assess users' credit, and government data serves as the basic credit anchor point. These data dimensions are indispensable, but due to data security and personal information protection restrictions, organizations can not directly share raw data. This "data silo" phenomenon makes traditional centralized modeling difficult to implement: if you rely on data from a single institution, the model will have "blind spots" due to the lack of feature dimensions (e.g., you cannot identify the risk of sudden consumption downgrade using only bank data); if you force data aggregation, you will face difficulties in the conditions of data sharing and the risk of leakage. What's more, credit evaluation is directly related to the flow of large sums of money, which has almost harsh requirements on model performance - the need to simultaneously meet the comprehensiveness of multi-source data fusion (covering a large number of dimensions, such as lending, consumption, taxation, etc.), the broad dimensionality of feature processing (structured data and unstructured text/image analysis), the efficiency of real-time decision-making (milliseconds response to the loan), and the need to provide a comprehensive and comprehensive modeling framework for the credit evaluation. Efficient in real-time decision-making (millisecond response to loan applications), and robust against attacks (preventing counter samples such as forged water flow from black production).This paradigm of "both wants and needs" constitutes a natural testing ground for federated learning technology.

Federal Learning achieves Pareto improvement in privacy protection and model performance through the architectural design of "data does not move, model moves". The core breakthrough lies in: first, the use of distributed feature engineering, where each organization extracts key features locally and shares only the results of feature computation instead of the original data; second, the use of encrypted gradient aggregation, so that the global model can learn the data patterns of all the participants, but can not inversely infer the sensitive information of specific users.

2 Related work

In 2020, with the upturn of big data and artificial intelligence technologies, big data-driven models, such as various types of machine learning models, start to be applied in the field of financial risk management, especially credit scoring and rating. Under the premise of protecting data privacy, Fanglan Zheng [1] et al. proposed a projected gradient method based on bounded constrained logistic regression, FL-LRBC, for traditional scorecards in a vertical federated learning framework. This is a successful application of federated learning and traditional machine learning models interacting with each other in the field of credit scoring. In the coming year, 2021, also from Fanglan Zheng [2] and other scholars improve on previous research and propose a logistic regression vertical joint learning (VFL) structure with constraints for traditional scorecards, which allows multiple organizations to jointly obtain an optimized scorecard model in a single training. FL-LRBC can form scorecard models with positive coefficients, which more specifically proves the robustness and interpretability of the model while avoiding the time-consuming parameter tuning process. After that, federal learning has gradually gained a firm foothold in the field of credit assessment, and more and more scholars have expanded the training idea of FL, perfected it, and applied it in the field of financial risk management and credit assessment. In 2024, Abdollah Rida [3] first brought the latest research on machine learning and deep learning in finance, credit scoring models, to a highly regulated environment such as a bankuse, which has never been done to date. With the feasibility of federated learning in the field of credit risk assessment proven, in the same year Shuyao Zhang, Jordan Tay, Pedro Baiz [4] explored two neural network architectures on three different datasets - multilayer perceptron (MLP) and Long Short-Term Memory (LSTM), and a tree ensemble architecture, Extreme Gradient Boosting (XGBoost).It is demonstrated that the joint model consistently outperforms the local model on non-dominant clients with smaller datasets.

In addition to the feasibility of federated learning with the performance gains it brings in credit assessment scenarios relative to traditional machine learning, the unique privacy computing properties of federated learning algorithms are equally well-tested in the field. In 2022, Xue Jiang, Xuebing Zhou, Jens Grossklags [5] generalized vertical federated learning reconstruction attack framework based on gradient was designed for flexible application to simple logistic regression models and multi-layer neural networks. In the same year, in order to address privacy leakage in HE (homomorphic encryption) based protocols, Yuzheng Hu [6]and other scholars developed a simple but effective differential privacy (DP)-based countermeasure and provided utility and privacy guarantees for the updated algorithm. And during our exploration, we retrieved Daniel J. Beutel, Taner Topal (Beutel, Daniel J., et al. "Flower: A friendly federated learning framework." (2022).)and other scholars involved in the research and development of the Flower framework, their proposed Flower framework is a very comprehensive federated learning framework that provides new capabilities for performing large-scale FL experiments and takes into account rich and heterogeneous FL devices, as well as providing very important insights and contributions to privacy computation methods and cryptographic protocols for federated learning.

Comprehensively, the Flowers framework supports the horizontal and vertical learning methods of federated learning, that is, it supports data splitting computation based on samples and features, respectively. It also supports local environment placement in heterogeneous systems (supporting PyTorch, TensorFlow) Meanwhile, the Flower framework utilizes a secure aggregation scheme to implement the SecAgg+ protocol (Bell et al., 2020) to support semi-honest models. The combined approach of client-side local training and encrypted aggregation of model updates (individual updates are not available to the server) provides a double privacy guarantee for the implementation of private computing. Its performance optimization for linear computation and communication overhead also contributes significantly.

Inspired by past research in this scenario, we believe that federated learning can indeed be very helpful in the scenario of credit assessment of users by various organizations. In previous studies, the research method using neural network structure is not compatible with the federated learning theory, and the local training models used in the research results of federated learning methods are based on simple machine learning models such as linear regression, and can only deal with binary classification scenarios. In the context of personal credit assessment, a person's credit rating should not only be classified into two categories, good and bad, and a more multifaceted credit rating assessment can often be more helpful to an organization's decision making. Linear regression methods often show great limitations in the complex nonlinear relationships of multi-categorization, while neural networks transform the inputs through nonlinear activation functions (e.g., ReLU, Sigmoid, tanh), so that the outputs of each layer become a combination of nonlinear functions, which usually achieves higher performance if encountering multi-categorization scenarios and the end-to-end self-learning capability can also omit a lot of feature engineering steps, which makes it easier to learn the complex laws in the data and has better model generalization ability. On the other hand, existing studies based on federated learning methods for analyzing and predicting individual credit ratings are implemented with vertical federation, whereas in credit assessment scenarios, horizontal federation learning (HFL) and vertical federation learning (VFL) each have their own applicability. For example, HFL requires participants to have the same characteristics but different user groups (e.g., banks in different regions) without the need for complex sample alignment operations. VFL, on the other hand, requires matching overlapping user samples through cryptographic techniques (e.g., PSI protocol), which has a high computational overhead and tends to be a performance bottleneck. The property of horizontal federation learning that only requires aggregation parameters also allows for faster convergence and iteration, lower communication costs, and higher privacy security. In other words, the two federated learning modes have their own usage scenarios, and the federated learning mode should be chosen according to the needs of the actual application.

In summary, we expect to design neural networks, xgboost, and other models with more generalization ability to deal with data in binary and multiclassification scenarios respectively, and to subdivide the assessment of individual credit rating into more levels. At the same time, based on Flower framework, FedAvg and other methods, we demonstrate the

performance improvement of vertical federation learning method and horizontal federation learning method compared with single machine learning and single client training model when dealing with different datasets, features and scenarios in the task of user's credit rating evaluation.

2.1 Horizontal federated learning

Horizontal Federated Learning (HFL), also referred to as sample-based federated learning, is a decentralized machine learning approach in which multiple participants collaboratively train a model without sharing their local datasets. In this method, all participants have the same feature space but different samples. This makes HFL especially useful in scenarios where organizations hold similar types of data for different individuals and wish to build a shared model while preserving privacy. Each organization or institution can train its own model using local data and then share model updates rather than raw data. This collaborative model training enhances the overall model's performance without exposing sensitive data.

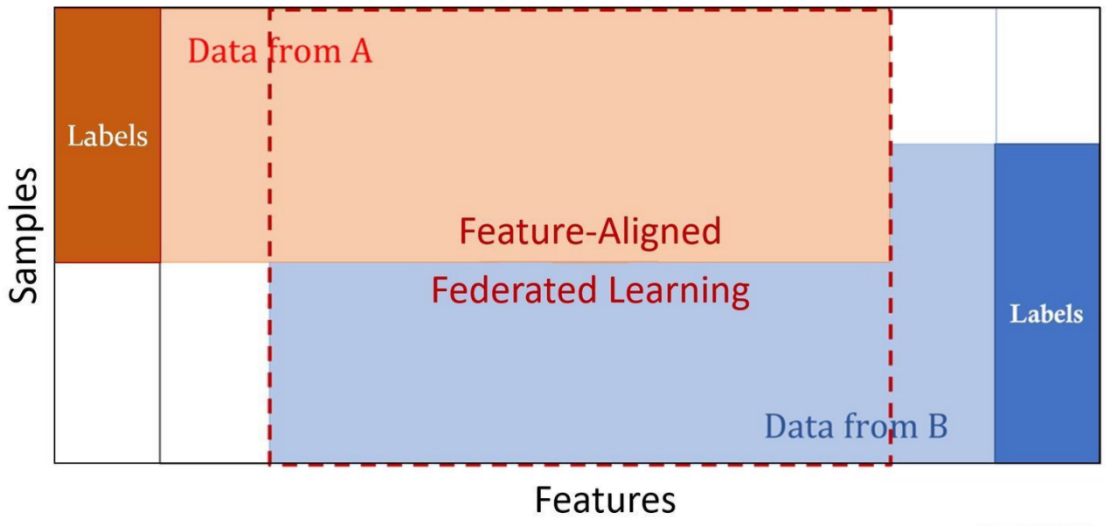

Figure 1: Horizontal Federated Learning Architecture

The concept of HFL was introduced by McMahan et al. in 2016 [7] as part of their work on federated learning, where data is kept local and only the model updates are exchanged. This approach ensures data privacy because the raw data never leaves the local devices or systems. Instead, only aggregated model updates are sent to a central server for the purpose of combining them into a more refined global model. This enables institutions to collaborate without the need to share private or sensitive information, thus minimizing the risks of data breaches and helping to comply with privacy regulations such as the General Data Protection Regulation (GDPR).

One of the key advantages of HFL is its ability to work with decentralized data while still enabling collaborative training. This makes it particularly valuable in industries such as finance, where different institutions may hold similar data for different sets of individuals.

By aggregating updates from multiple sources, HFL can produce a model that is more generalized and robust, as it benefits from the diverse datasets of all participating clients. In scenarios like credit scoring, for example, banks or financial institutions can train a joint model to assess credit risk without sharing their customers' private financial data.

Despite its advantages, HFL does present challenges. One major issue is data heterogeneity, where different clients may have data distributions that are not identical, potentially affecting the model's performance. Additionally, securing the aggregation of model updates and minimizing communication overhead are also critical considerations. Techniques like secure multi-party computation (SMPC) and differential privacy are frequently used in HFL to address these challenges, ensuring the security and efficiency of the collaborative training process.

Overall, Horizontal Federated Learning presents a promising solution for machine learning in scenarios where privacy and data security are critical. It has been successfully applied in various fields such as finance, healthcare, and e-commerce, allowing organizations to collaborate on developing models while maintaining the privacy and confidentiality of their data. This ability to train models on decentralized data sets without sharing sensitive information makes HFL a powerful tool for real-world machine learning applications.

2.2 Vertical federated learning

Based on the data distribution characteristics among participants, federated learning is generally categorized into three types: Horizontal Federated Learning (HFL), Vertical Federated Learning (VFL), and Federated Transfer Learning (FTL). Among these, Vertical Federated Learning has emerged as a focal point of joint attention in both academic and industrial circles due to its unique value in commercial collaborative modeling. Unlike the scenario in Horizontal Federated Learning where participants share the same feature space but cover different user groups, the core characteristic of Vertical Federated Learning lies in the fact that participants hold data of the same user group across different feature dimensions. This data distribution pattern is particularly prevalent in cross-industry collaboration scenarios such as financial risk control and precision marketing, where different entities possess complementary information about the same set of users, making it crucial to combine these diverse data sources for more accurate modeling and decision-making.

The concept of Vertical Federated Learning can be traced back to the systematic elaboration by Yang et al. [8]in their research in 2017. However, its technical rudiments stem from early explorations in the field of privacy-preserving data mining. In traditional data collaboration models, enterprises need to directly share raw data to complete collaborative modeling, which not only faces legal compliance risks but also easily leads to the leakage of commercial secrets. The proposal of Vertical Federated Learning has fundamentally changed this situation. By leveraging cryptographic technologies and distributed computing frameworks, it enables participants to jointly train machine learning

models without exposing local data. This paradigm of "data remains stationary, while models move" perfectly aligns with the requirements of privacy regulations such as the General Data Protection Regulation (GDPR). Specifically, it ensures that personal data and sensitive business information do not leave the local storage systems of the participating parties, thereby avoiding the potential violations of data protection laws that may occur during data transmission and sharing. As a result, this innovative approach has rapidly gained traction and been widely adopted in regions with strict regulatory environments like the European Union.

From the perspective of technical implementation, the core challenge of Vertical Federated Learning lies in how to achieve the integration of feature spaces and the efficient transmission of gradients without directly exchanging raw data. Currently, mainstream methods mainly revolve around three major technical systems: Secure Multi-Party Computation (SMPC), Homomorphic Encryption (HE), and Differential Privacy (DP). Early Vertical Federated Learning systems such as SecureBoost adopted Additive Homomorphic Encryption (AHE) to protect gradient information and implemented encrypted aggregation of model parameters through the Paillier cryptosystem. Although such methods can provide strong security guarantees, they face significant computational overhead caused by ciphertext expansion. This computational burden is mainly reflected in the prolonged processing time when dealing with large-scale datasets and complex model structures, which to some extent limits the practical application of these early systems in real-world scenarios with high real-time requirements. To balance efficiency and security, subsequent studies have proposed various optimization schemes. For example, gradient transmission protocols based on Secret Sharing distribute sensitive computations among multiple participants, such that no single party can restore complete information. The FATE (Federated AI Technology Enabler) framework developed by Microsoft Research innovatively adopts a hybrid encryption strategy: it uses homomorphic encryption in the model initialization phase and switches to lightweight Garbled Circuit technology in the iterative update phase. This strategic combination of different encryption technologies effectively reduces the computational complexity during the model training process, significantly improving the practicality of the framework in scenarios with large-scale features and thus promoting the wider application of Vertical Federated Learning in industrial practice.

In financial service systems powered by Vertical Federated Learning, when the dimensionality of unique features possessed by each participating institution exceeds that of shared features, model training is conducted based on overlapping samples with shared identifiers. Such systems represent an extension of Horizontal Federated Learning (HFL) systems, featuring a data partitioning mode with sample-level overlaps. To illustrate with a practical scenario, when a bank intends to develop a credit scoring model, it will seek collaboration with institutions in other industries to obtain complementary information. At this point, the collaborating parties will jointly build a VFL model, with each retaining a part of the model. It is worth noting that this architectural design not only avoids direct interaction of raw data but also integrates the advantages of multi-party features through a

distributed training mechanism. For instance, in credit evaluation scenarios, a bank may hold financial data such as users' account transactions, while a cooperating e-commerce platform possesses data on users' consumption behaviors. By sharing sample identifiers, the two parties achieve feature complementarity. Meanwhile, each institution only maintains the part of the model related to its own features, thereby improving model performance while ensuring data privacy.

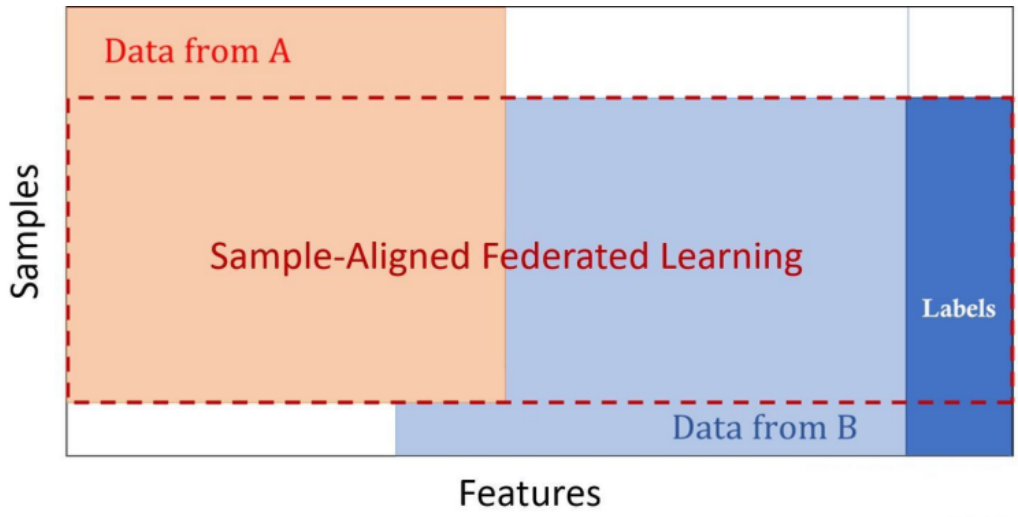

Figure 2: Vertical Federated Learning Architecture

The advantages of Vertical Federated Learning are mainly reflected in three aspects: First, its stringent privacy protection mechanism enables cross-industry data collaboration, especially for organizations like banks and e-commerce platforms that have complementary businesses but possess sensitive data. This mechanism, relying on cryptographic technologies such as secure multi-party computation and homomorphic encryption, ensures that raw data does not leave the local storage of each participant during the collaboration process, thus eliminating the risk of data leakage caused by direct data sharing. Second, by integrating high-dimensional features from multiple parties, the federated model can break through the data bottleneck of a single party and significantly improve prediction performance. Specifically, through the alignment of sample IDs, the feature dimensions of data under the same ID are expanded, allowing the model to learn feature information from more dimensions and thereby contributing to more accurate model predictions. Third, the distributed computing architecture enables each participant to only maintain the feature extraction module for local data, which greatly reduces the bandwidth consumption in model updates. Since there is no need to transmit large-scale raw data or complete model parameters between participants, only intermediate calculation results (which are usually encrypted) are exchanged, the pressure on network bandwidth is effectively alleviated, especially in scenarios involving a large number of participants or massive data volumes.

However, Vertical Federated Learning also faces non-negligible technical challenges: The accuracy of the sample alignment process directly affects the final model performance, but high-precision Private Set Intersection (PSI) protocols often incur huge computational overhead under strict privacy constraints; this is because PSI needs to compare and match

sample IDs among multiple parties without disclosing sensitive information, and the complex cryptographic operations involved consume a lot of computing resources, which may slow down the entire model training process. The gradient deviation problem caused by Non-Independent and Identically Distributed (Non-IID) features will delay model convergence, especially when there is a large difference in data quality among participants; for example, if one participant's data has more missing values or stronger noise, the gradients calculated based on such data will deviate from the optimal direction, making it difficult for the federated model to converge to the global optimal solution. In addition, most existing security protocols assume that participants are "Honest-but-Curious", and their ability to defend against malicious attacks still needs to be strengthened. Malicious participants may intentionally tamper with intermediate calculation results or launch inference attacks to steal other parties' data, which poses a serious threat to the security and reliability of the federated learning system.

In this project, we conducted a simulated vertical federated learning experiment using three datasets: German credit data, Loan data, and MIT credit ranking. Vertical federated learning emphasizes that different clients possess data with distinct feature dimensions, while sharing the same data IDs, which enables the concatenation of feature dimensions. We first performed an average distribution of data features, ensuring that each client holds a certain number of features without any overlap between the features owned by different clients. This setup is designed to simulate real-world scenarios where different institutions may store varying feature information for the same user. After completing the client feature allocation, we implemented vertical federated learning under the Flower and SecretFlow federated learning frameworks. Subsequently, we compared the performance of single-party training with that under vertical federated learning, aiming to highlight the advantages of vertical federated learning.

3 Methodology

3.1 Dataset description

The German Credit Dataset is one of the most well-known datasets in the domain of credit risk prediction. It contains 1,000 records with 20 attributes, most of which are categorical or symbolic in nature. Each record represents an individual who has applied for credit from a bank, and the goal is to classify each individual as either a good credit risk or a bad credit risk based on these attributes. The attributes in the dataset include a mix of demographic, financial, and credit history details, all of which are important indicators for assessing creditworthiness. Some of the key features include credit history, which indicates past financial behavior; purpose, which refers to the reason for taking out the loan (e.g., car, education, home improvement); savings account, reflecting the individual's financial security; and employment status, which is a significant predictor of the person's income stability. Other attributes include age, personal status, other parties (e.g., co-signers), and housing arrangements, all of which provide valuable context for determining financial responsibility.

The primary task with this dataset is binary classification, where the goal is to predict whether a person is a good credit risk or bad credit risk. This makes it an excellent dataset for exploring traditional machine learning techniques like logistic regression, decision trees, and support vector machines. One of the key challenges of this dataset is handling imbalanced class distributions, as bad credit risks are typically fewer than good credit risks. This dataset has been widely used in the financial sector to develop credit scoring models, making it an excellent starting point for evaluating and comparing models in the field of credit risk prediction. The German Credit Dataset provides a practical example of how financial institutions can use data-driven models to assess credit risk based on customer demographics and financial history.

The Loan Data from LendingClub offers a much larger and more complex dataset compared to the German Credit Dataset. LendingClub is one of the world's largest peer-to-peer lending platforms, and the dataset includes information about over 2 million consumer loans originated between 2014 and 2018. This dataset contains a broad range of features that provide insights into the borrower's personal details, loan characteristics, and loan performance. Key features in the dataset include loan amount, term, interest rate, and grade, which is assigned based on the borrower's creditworthiness. Other important attributes include the employment length, annual income, home ownership status, and purpose of the loan (e.g., debt consolidation, home improvement, education).

Given the size and complexity of this dataset, it presents several tasks for machine learning models, including loan default prediction, interest rate prediction, and loan classification. The task of predicting whether a loan will default is a binary classification problem, while other tasks, such as predicting the interest rate, may involve regression techniques. The dataset's large scale and diversity make it particularly useful for testing the scalability and robustness of different machine learning models in real-world settings.

Additionally, this dataset allows for the exploration of unsupervised learning techniques, such as clustering borrowers with similar financial behaviors. The sheer size of the dataset, coupled with detailed borrower information, also makes it valuable for studying imbalanced learning, as defaults are relatively rare compared to fully paid loans. Researchers can leverage this dataset to test more advanced techniques like ensemble learning, boosting methods, and neural networks, making it an ideal choice for more complex financial risk prediction tasks.

The MIT Credit Ranking Dataset is a unique dataset designed by the Massachusetts Institute of Technology (MIT) to categorize individuals based on their creditworthiness into four distinct classes. These classes are defined as P1: Very Good Creditworthiness, P2: Good Creditworthiness, P3: Fair Credit, and P4: Bad Credit, and they offer a more detailed classification of credit scores than simple binary classification datasets. The dataset includes a variety of attributes related to customer demographics, credit history, and financial status. Key features in the dataset include age, income, credit history, and loan amount, among others. These features provide a comprehensive view of an individual's financial behavior and help in determining their creditworthiness.

This dataset is particularly valuable for multi-class classification tasks, where the goal is to predict one of the four creditworthiness categories (P1 to P4) based on a person's financial information. The multi-class nature of the task introduces additional complexity and offers a more nuanced view of credit risk, as it requires the model to not only classify individuals as good or bad credit risks but also to differentiate between varying degrees of creditworthiness. The MIT Credit Ranking dataset is useful for studying advanced classification models, including multi-class decision trees, support vector machines (SVMs), and neural networks, which can handle the complexity of distinguishing between more than two categories. The inclusion of customer demographics and credit history in the dataset makes it an excellent resource for exploring how various factors influence credit scores, and it allows researchers to build more granular credit scoring systems that can help financial institutions make more informed lending decisions.

In summary, the three datasets—German Credit, Loan Data from LendingClub, and MIT Credit Ranking—each provide unique insights into credit risk prediction, offering distinct features and classification tasks that serve different purposes. The German Credit Dataset is a widely used resource for studying binary credit classification and provides an accessible entry point for evaluating machine learning models in a credit scoring context. The Loan Data from LendingClub is a much larger and more complex dataset, offering a wealth of information about loan characteristics and borrower details, making it ideal for studying loan default prediction, interest rate prediction, and other financial behaviors. Finally, the MIT Credit Ranking Dataset provides a more detailed approach to creditworthiness assessment, enabling the development of multi-class classification models that can categorize individuals into different levels of credit risk. Collectively, these datasets are invaluable tools for researchers aiming to develop accurate and efficient credit scoring models, and they offer a wide range of opportunities for applying machine learning techniques to real-world financial problems.

3.2 Flower federated AI framework

In this project, we mainly use Flower as our federated learning framework. Flower is a powerful and flexible federated learning (FL) framework designed to assist researchers in implementing and studying federated learning systems in real-world scenarios. Unlike traditional FL systems that often rely on a central server and homogeneous client environments, Flower enables collaboration among a diverse range of clients, including edge devices like smartphones, tablets, and embedded systems, without centralizing the data. This decentralized approach is essential for preserving privacy while still enabling the benefits of collaborative learning. Flower stands out due to its ability to seamlessly transition between simulated and real-world devices, providing an ideal tool for researchers to test and experiment with FL algorithms in both controlled environments and real-world settings.

One of the standout features of Flower is its client-agnostic design. This means that it can operate with a wide variety of devices running different operating systems and frameworks, making it highly adaptable for federated learning tasks. Flower is capable of supporting heterogeneous client environments, where clients have varying computational capabilities, memory capacities, and network conditions. This is crucial in real-world applications, as devices that participate in FL can range from low-power embedded systems to high-performance smartphones, each with different hardware and software configurations.

Flower's architecture is built to scale from small-scale experiments to large cohorts, handling millions of clients participating in concurrent training rounds. This scalability is achieved through advanced resource management features, such as the Virtual Client Engine (VCE), which maximizes the utilization of available hardware. The VCE enables large-scale experiments by virtualizing clients, thereby allowing Flower to execute federated learning tasks even when resources are limited. This feature makes Flower an excellent tool for conducting large-scale FL research while ensuring efficient resource utilization.

Flower is designed to be framework-agnostic, allowing it to integrate seamlessly with various machine learning frameworks, such as TensorFlow, PyTorch, and others. This flexibility ensures that researchers can use their existing machine learning models and seamlessly integrate them into federated learning workflows without needing to redesign their models. Flower provides an abstraction layer that allows federated learning experiments to be conducted using different machine learning frameworks and tools, making it adaptable to the fast-evolving landscape of machine learning technologies.

Moreover, Flower supports heterogeneous client environments. It allows clients running different operating systems and programming languages to participate in the same federated learning experiment. This is a major advantage when dealing with a diverse set of devices, as many federated learning frameworks tend to be constrained by specific programming languages or platforms. Flower's ability to work across multiple languages and platforms ensures that it can handle complex, real-world federated learning tasks that involve a wide variety of devices.

Privacy is one of the most critical concerns in federated learning, as it involves the sharing of model updates between devices without revealing sensitive data. Flower addresses this challenge through the integration of robust privacy-preserving mechanisms. One of the core privacy features in Flower is its implementation of Secure Aggregation (SecAgg), which ensures that the server never has access to the raw data from clients. Instead, the server aggregates model updates securely without exposing sensitive client data, even if the server is "honestly curious" about the data. This is accomplished by using secure multi-party computation techniques that ensure the aggregation process is secure, while still enabling the server to perform the necessary model updates.

Flower's SecAgg implementation is based on protocols developed by Bonawitz et al. (2017) and Bell et al. (2020), designed to provide strong privacy guarantees with minimal computational overhead. These protocols ensure that even when clients drop out or fail to provide updates, the security of the system is maintained. Flower's implementation of differential privacy further strengthens privacy by adding noise to the model updates, preventing individual data points from being identified or reverse-engineered. This multi-layered approach to privacy protection ensures that Flower is suitable for use in privacy-sensitive applications, such as healthcare, finance, and personal data analysis.

Flower is particularly notable for its ability to scale from small research experiments to large-scale deployments. Traditional federated learning frameworks are often limited in terms of client pool size or the number of concurrent clients they can handle. Flower, however, can support up to millions of clients, making it suitable for large-scale, real-world applications. Its scalability is enhanced by its ability to handle a large number of concurrent clients, and its architecture allows for seamless integration of real-world devices with varying network capabilities.

The Virtual Client Engine (VCE) in Flower also enables the execution of large-scale federated learning experiments in a resource-aware manner. The VCE schedules and runs clients in a way that maximizes hardware utilization while minimizing overhead, which is critical when running FL tasks on limited hardware resources. This allows researchers to perform FL experiments on a single machine or multi-node clusters, making it easier to conduct experiments in a wide variety of setups without requiring significant reconfiguration.

Another significant advantage of Flower is its open-source nature. Flower is released under the Apache 2.0 license, allowing researchers to freely use, modify, and contribute to the framework. This open-source model fosters a collaborative research environment, enabling researchers to extend the framework with new algorithms, optimizations, and privacy-preserving techniques. As the federated learning landscape evolves, Flower's extensibility ensures that it can quickly incorporate the latest developments in the field, allowing researchers to stay at the forefront of FL research.

In conclusion, Flower represents a major advancement in the field of federated learning. Its scalability, flexibility, and privacy features make it an ideal framework for both academic research and real-world deployment. Flower provides the tools needed to experiment with

federated learning algorithms and systems in heterogeneous environments, making it possible to run FL experiments at scale while preserving client privacy. Whether used for small-scale research or large-scale deployments, Flower's ability to handle diverse devices and adapt to various machine learning frameworks makes it a powerful tool in the future of federated learning research and development. With its open-source nature and robust privacy mechanisms, Flower is set to play a crucial role in the widespread adoption and deployment of federated learning systems.

3.3 Horizontal federated learning design

3.3.1 German credit data horizontal federated learning

We begin by uniformly preprocessing the entire German Credit dataset to ensure consistency and reproducibility. All missing numerical values are imputed with their respective column medians, while missing categorical entries are filled using the mode of each feature. Numerical features are then standardized to zero mean and unit variance, and categorical fields are transformed via one-hot encoding, resulting in an approximately 50-dimensional feature vector for each record. Using a fixed random seed, we randomly shuffle the dataset and split it into three equal shards of 333 samples each, carefully preserving the original "good" vs. "bad" credit label proportions within each shard. Each shard is further partitioned locally into an 80:20 training (266 samples) and test (67 samples) split. To eliminate class-imbalance bias at the client level, we apply undersampling on the majority class in both training and test subsets, guaranteeing an exact 1:1 ratio of "good" to "bad" examples throughout.

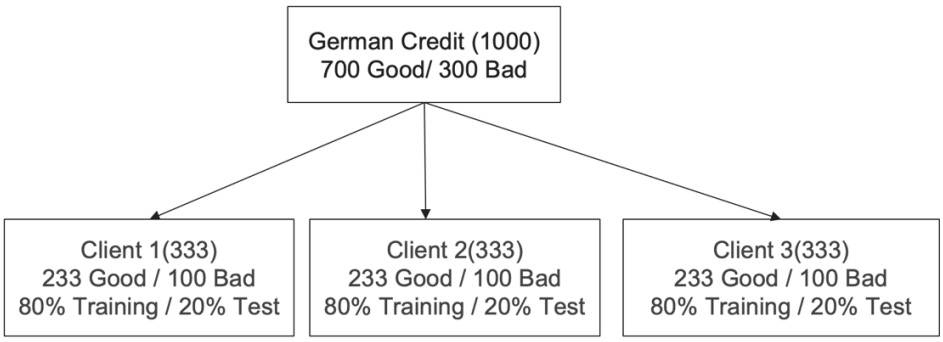

Figure 3: German Credit Data Split

We employed the multilayer perceptron implemented with `sklearn.MLPClassifier`. Its topology is 20 → 100 → 50 → 2: twenty input neurons ingest the standardized features of the German Credit dataset; two fully-connected hidden layers with 100 and 50 neurons apply ReLU activations, capturing non-linear relationships; the final layer outputs the logits for the two credit-risk classes. Altogether the model contains 7,252 trainable weights and biases. We kept hyper-parameters fixed across all runs—Adam optimizer with a learning rate of 0.001, L2 regularization strength 0.001, a batch size large enough to cover each local dataset, and early-stopping based on validation loss—to guarantee that any performance difference would stem from the training paradigm rather than architectural or tuning variations. All features were z-score normalized using `StandardScaler`, and the same random seed ensured reproducibility of weight initialization and data shuffling.

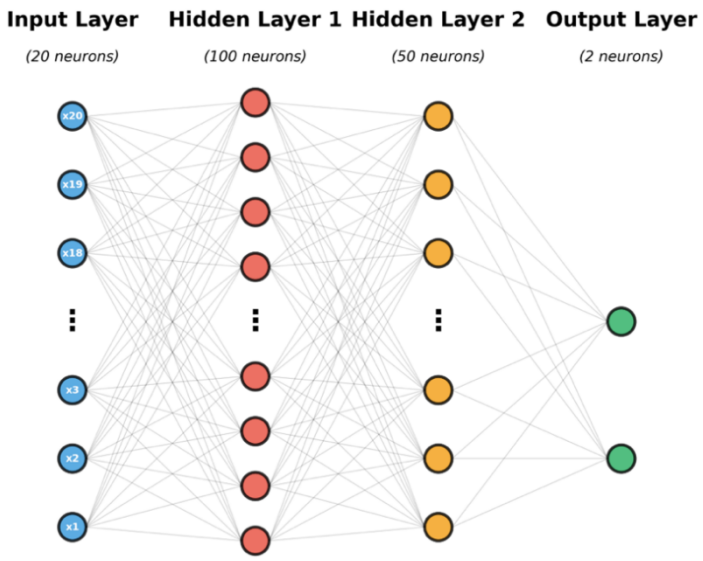

Figure 4: German credit data neural network structure

The 1,000-sample dataset was first stratified on the target variable and then split into three equal, non-overlapping shards to simulate three independent clients. In the standalone scenario each client performed a conventional 80 / 20 train-test split and trained the network privately until the early-stopping criterion halted optimization—roughly 21 gradient-descent iterations on average. For the federated scenario we adopted the classical FedAvg algorithm but crafted the protocol to be comparable in computational effort: the server orchestrated exactly 21 communication rounds, and during each round every client executed a single local training iteration using its shard, then transmitted the updated weights (not the data) to the server. The server aggregated these parameters by weighted averaging—weights proportional to each client's sample count—before broadcasting the new global model back. This design aligns total gradient updates with the standalone case while highlighting the key advantage of federated learning: clients collectively improve a shared model without exposing raw credit-record data, thereby preserving privacy and complying with data-sovereignty constraints. runs while enabling collaborative learning across clients without sharing raw data.

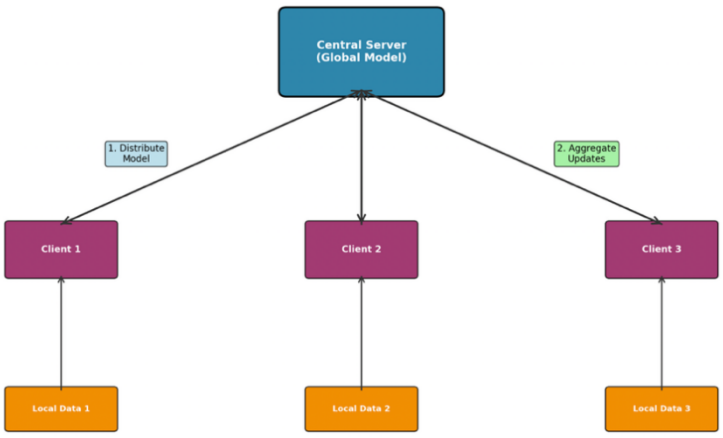

Figure 5: Central server and clients structure

3.3.2 Loan data horizontal federated learning

Our experiment begins with a dataset of 2 million loan records, from which 100,000 records are selected to form a more manageable yet diverse subset. This ensures that we maintain a sufficient level of complexity while keeping the data tractable for experimentation. The selected 100,000 records undergo extensive preprocessing to ensure data quality. Missing numerical values are imputed using the median of the respective feature, while categorical variables are filled with the most frequent category. All continuous features are normalized to zero mean and unit variance, and categorical features are one-hot encoded. This standardization and encoding ensure that the data is suitable for feeding into a machine learning model. The final dataset is then split into training (80%) and testing (20%) sets, ensuring a good balance and reliable evaluation of model performance.

The data are evenly divided among three clients, each holding approximately 33,333 samples and preserving the same original positive-to-negative ratio (83.75%–83.89%) to ensure consistency in distribution. Each client's subset is further split into training and test sets at an 80:20 ratio, using stratified sampling to keep class proportions identical in both. Prior to partitioning, all features are standardized using a common StandardScaler fitted on the full dataset, ensuring uniform feature scaling across clients.

We designed a neural network comprising four fully connected layers: an input layer to 128 neurons, 128 to 64 neurons, 64 to 32 neurons, and a final output layer with 2 neurons. Each hidden layer is followed by Batch Normalization (BatchNorm1d) and Dropout regularization (dropout rates of 0.3, 0.3, and 0.2, respectively) to enhance model stability and generalization. The network uses ReLU activations, Xavier uniform weight initialization, and a Softmax activation at the output for binary classification. All models are trained with the AdamW optimizer (learning rate = 0.001, weight decay = 0.01) and a cosine annealing learning rate scheduler to maintain consistency across experiments.

To ensure a fair comparison, both training paradigms use the same total amount of training: in standalone mode, each client trains for 30 epochs; in federated learning (FL) mode, we perform 30 global aggregation rounds, with each client training for 1 epoch per round, yielding an equivalent total workload. Both modes share the same architecture, optimizer settings, batch size (64), loss function (CrossEntropyLoss), and random seed to eliminate bias. Federated averaging (FedAvg) aggregates client updates by weighted averaging according to each client's data volume. We avoid pretraining, data augmentation, or any additional enhancements so that the comparison solely reflects the intrinsic differences between standalone and federated training. The results show that federated learning outperforms standalone training on all clients, achieving an average accuracy improvement of 0.38%.

3.3.3 MIT credit ranking horizontal federated learning

The assessment of credit rating should not be limited to binary classification scenarios, i.e., a person's credit rating, in many practical application scenarios, should not be simply classified into good and bad categories. The MIT dataset attempts to classify and score the different credit profiles of a user by using four levels, so as to assess the credit profile of a single individual in a more comprehensive and detailed way. We also therefore design a level federation learning method suitable for the four-classification scenario for experimental validation.

In this part of experiment, we try to accomplish a four-classification task with a real dataset, with the aim of comparing the effect of training the model individually for each client with the difference in performance after using federated learning (FedAvg).Through this comparison, we hope to gain a deeper understanding of whether federated learning has any advantages in dealing with multi-categorization problems or not. To do so, we built neural network models with the same structure on each of the three clients, and then used horizontal federation learning to integrate the model parameters obtained from their respective training.Finally, we evaluated the prediction effect of the model in three dimensions: Accuracy, F1-Score and AUC (Area Under the Curve).

For this experiment, we chose a Feedforward Neural Network as the base model. It contains two hidden layers, each with 64 neurons, and uses ReLU as the activation function.The output layer of the model is then connected to softmax by linear transformation to handle the task of multiple classification. For training, we used CrossEntropyLoss to evaluate the prediction error and Adam optimizer to improve the convergence speed of the model to make the training process more stable and efficient.In order to prevent overfitting situations, regularization techniques (Dropout layer) are incorporated in the model to effectively prevent overfitting.

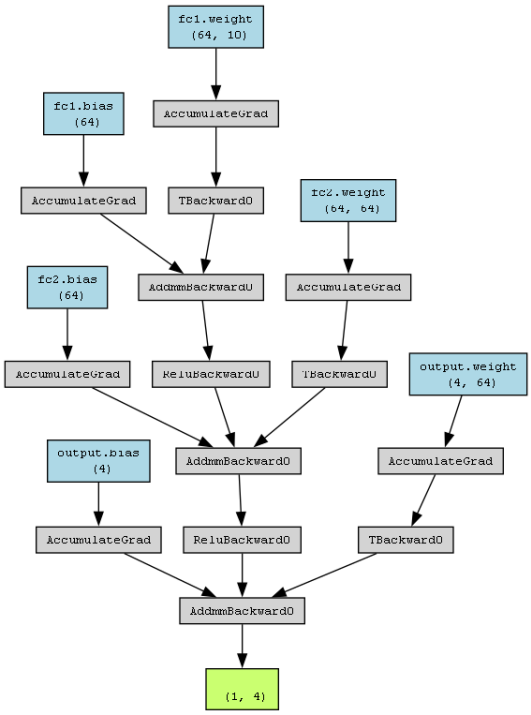

Figure 6: MIT credit ranking neural network structure

As for the part of federated learning, we use the classical Federated Averaging (FedAvg) algorithm. The core idea of this strategy is: each client locally trains for a few rounds, then uploads the weights of the trained model to a central server, which then weights and averages these weights to obtain a global model, and finally sends this model back to each client for further training. A big advantage of FedAvg is that it is computationally efficient, has low communication costs, and is particularly suitable for dealing with the inconsistencies in the distribution of data (Non-IID) or with privacy restrictions.

The MIT Dataset divided the data equally into three parts and aligned the sample IDs. One of the parts, which contains the personal information of users, classifies the users' credit ratings into four categories from high to low. The other two parts of the data contain other bank and account features but do not include classification labels. Finally, after processing the missing values, dealing with the outliers, labelizing the character-type data, and standardizing the data, these three parts of data were put into use in this round of experiment.

Three clients were designed for this round of experiments, and the clients were individually trained using a feedforward neural network as well as invoking the federated learning FedAvg strategy to aggregate the three client model parameters, which resulted in the following comprehensive performance after 20 rounds of training:

3.4 Vertical federated Learning design

3.4.1 German Credit Data vertical federated learning

(a) German Credit Data VFL design with NN

For the German Credit Data, we implement a vertical federated learning (VFL) approach to enable collaborative model training while maintaining data privacy across three distinct financial institutions. The dataset undergoes standardized preprocessing where all 20 numerical features are scaled to zero mean and unit variance using scikit-learn's StandardScaler, with no missing values requiring imputation. To address class imbalance, we apply global undersampling to achieve a balanced 1:1 ratio of "good" to "bad" credit cases prior to partitioning, resulting in a final dataset of 600 samples. This balanced dataset is then split into training (80%) and testing (20%) sets while preserving class distribution in both subsets.

The feature space is vertically partitioned among three clients: Client 1 receives the first 7 features (indices 0-6), Client 2 obtains the subsequent 7 features (indices 7-13), and Client 3 acquires the remaining 6 features (indices 14-19) while also serving as the label owner. Sample alignment is rigorously maintained through shared indices across all partitions to ensure consistent record matching during collaborative training.

In the standalone baseline configuration, each client trains a local model using only its partitioned features. All standalone models employ identical feed-forward neural network architectures featuring an input layer matching their feature dimensions, followed by two hidden layers (128 and 128 neurons with ReLU activation and 30% dropout), a third hidden layer (64 neurons with ReLU), a fourth layer (32 neurons with ReLU), and a sigmoid output neuron for binary classification. Training proceeds for 50 epochs with a batch size of 64, optimized via Adam (learning rate = 0.0005, L2 regularization λ=1e-4) using binary cross-entropy loss, with evaluation conducted on each client's private test set.

For the vertical federated learning setup, we implement a split neural network architecture where each client hosts a bottom model matching their standalone architecture but terminating in a 32-dimensional embedding layer. Client 3 additionally hosts the top model, which receives concatenated embeddings from all three clients (96 total dimensions) and processes them through two hidden layers (128 and 64 neurons with ReLU activation and 30% dropout), followed by a 32-neuron ReLU layer and sigmoid output. The federated training protocol coordinates forward and backward passes across clients: during each batch iteration, Clients 1 and 2 compute feature embeddings and transmit them to Client 3 via homomorphic encryption; Client 3 concatenates embeddings, computes predictions and loss, then backpropagates gradient signals to each client for local model updates. Optimization employs Adam (learning rate = 0.001) with ReduceLROnPlateau scheduling that halves the learning rate when validation accuracy plateaus for 5 consecutive epochs. Training runs for 50 epochs with batch size 32, and model performance is evaluated at each epoch using the federated test set, with the highest achieved metrics recorded as final results.

To comprehensively evaluate both approaches, we track three key metrics: accuracy, F1-score, and AUC. For standalone models, we report per-client test performance, while the VFL model is evaluated on the pooled test set. Convergence behavior is analyzed through epoch-wise metric trajectories, and final performance comparisons are visualized through both temporal progression plots and multi-metric bar charts.

(b) German Credit Data VFL design with XGBoost

In addition to using neural networks as the base model for vertical federated learning, we also conducted similar experimental validations on the German Credit data using the XGBoost model within the SecretFlow framework.

XGBoost, short for eXtreme Gradient Boosting, is a highly efficient and scalable machine learning algorithm based on the gradient boosting framework, widely used for classification, regression, and ranking tasks. At its core, XGBoost builds a strong predictive model by combining multiple weak learners (typically decision trees) in an ensemble manner, iteratively minimizing prediction errors through gradient optimization. Unlike traditional gradient boosting methods, XGBoost introduces several key innovations to enhance performance and robustness. These include regularization terms (L1/L2) to prevent overfitting, second-order Taylor expansion for more accurate loss function approximation, parallelized computation for faster training, and built-in handling of missing values. Additionally, XGBoost supports custom loss functions and offers extensive hyperparameter tuning options, making it highly adaptable to diverse problem settings. One of XGBoost's standout features is its computational efficiency, achieved through optimized data structures (e.g., compressed column blocks) and cache-aware algorithms, enabling it to handle large-scale datasets effectively. Its ability to automatically learn feature interactions and manage imbalanced data further contributes to its dominance in machine learning competitions, such as Kaggle, where it has been a winning solution in numerous challenges. Beyond competitions, XGBoost is widely adopted in industry applications, including fraud detection, recommendation systems, and financial risk modeling, due to its interpretability (via feature importance scores) and consistent performance across varied datasets.

First and foremost, we still need to allocate the feature columns to each client. It is observed that the German credit dataset contains a total of 21 feature columns, including the 'kredit' attribute which serves as the label. We assigned 6, 6, and 8 feature attribute columns to the three clients respectively. Meanwhile, we assumed that each client possesses the label of the samples. Therefore, including the sample labels, each client has 7, 7, and 9 feature attributes (non-label attributes + label attribute) respectively. Specifically, client 1 has: 'laufkont', 'laufzeit', 'moral', 'verw', 'hoehe', 'sparkont', 'kredit'; client 2 has: 'beszeit', 'rate', 'famges', 'buerge', 'wohnzeit', 'verm', 'kredit'; client 3 has: 'alter', 'weitkred', 'wohn', 'bishkred', 'beruf', 'pers', 'telef', 'gastarb', 'kredit'. Since this is vertical federated learning, each client owns all 1000 samples, but the features covered by the samples vary.

It is important to note that in the vertical federated learning experiment based on neural networks, we adopted a downsampling operation to address the issue of data class imbalance, deleting a large number of data samples to achieve class balance. In contrast, this experiment

aims to explore the effect of vertical federated learning under class imbalance, so we no longer perform downsampling on the data, but directly use all data samples for training and testing.

For each client, we separately created training sets and test sets in the ratio of 8:2 (training set: test set) for training standalone models in single-party scenarios. If vertical federated learning is to be conducted, the data from the three parties will be concatenated in the feature dimension. Under the SecretFlow framework, federated learning training will be carried out with any one party acting as the server. At this point, the training set for vertical federated learning is the concatenation of all training sets from the three clients, and the test set is also the concatenation of the test sets from the three clients. The XGBoost model trained through vertical federated learning will be tested on this concatenated large test set.

Here is a brief explanation of why it is necessary to test on the test set composed of all feature dimensions. Since vertical federated learning is a method to solve the problem of scarce data features, in practical applications, various institutions will upload some of the features of the data they own. After encrypted transmission and concatenation in the feature dimension, the model's prediction results will be shared with all institutions to assist in decision-making. Therefore, testing on the full-feature test set is more in line with actual needs. In addition, the model trained by vertical federated learning is based on all features. If the feature dimensions are reduced during testing, the performance will decline, making it difficult to evaluate the actual improvement brought by vertical federated learning.

The experimental procedure remains consistent: first, training and testing are conducted for individual clients, followed by training and testing under vertical federated learning. The parameter settings for XGBoost are as follows: tree_method="hist", n_estimators=50, max_depth=5, learning_rate=0.3, max_bin=10, base_score=0.5, reg_lambda=0.1, min_child_weight=0. The evaluation metrics used are still accuracy, F1-score, and AUC.

3.4.2 Loan Data vertical federated learning

(a) Loan Data VFL design with NN

For the loan default prediction dataset, we still implement a vertical federated learning framework to enable collaborative model training across three financial institutions while preserving data privacy. To construct our experimental, we directly sample from the original df_2014-18_selected.csv file. Specifically, to address class imbalance and create a perfectly balanced dataset, we randomly select 50,000 samples where loan_status_binary is 0 and another 50,000 samples where loan_status_binary is 1. This results in a final balanced dataset of 100,000 total samples with equal representation of both classes. All features undergo standardized preprocessing using scikit-learn's StandardScaler to achieve zero mean and unit variance, with no missing values requiring imputation.

The balanced dataset is partitioned into training (80%) and testing (20%) sets using stratified sampling to preserve class distribution. The feature space is vertically partitioned

among three clients based on financial domain logic: Client 1 (Loan Information) receives 4 features including loan amount-to-installment ratio, interest rate, loan age, and outstanding principal; Client 2 (Payment Information) obtains 5 features encompassing late fees, recoveries, last payment amount, total received interest, and total received principal; Client 3 (Status Information) acquires 3 features covering debt settlement flag and temporal variables related to credit inquiries and payments. Sample alignment is maintained through consistent indexing across all partitions to ensure proper record matching during federated training

For standalone baseline models, each client trains an independent neural network using only their local feature subset. The standalone architecture dynamically adjusts hidden layer dimensions based on input features: the first hidden layer size is calculated as max(64, input_dim × 16), followed by a second hidden layer of half that size, a third layer with 32 neurons (all with ReLU activation and 30% dropout), and a sigmoid output neuron for binary classification. Training proceeds for 100 epochs with batch size 64, optimized using Adam optimizer (learning rate = 0.001, weight decay = 1e-4) and binary cross-entropy loss.

The vertical federated learning architecture employs a split neural network design where each client hosts a bottom model (ClientModel) that processes their local features through two hidden layers with dynamically sized neurons (max(32, input_dim × 8) and half that size), followed by a 32-dimensional embedding layer with ReLU activation and 30% dropout. The server model (ServerModel) receives concatenated embeddings from all three clients (96 total dimensions) and processes them through four layers: 128 neurons, 64 neurons, 32 neurons (all with ReLU and 30% dropout), and a sigmoid output layer. The federated training protocol coordinates computation across clients: during each iteration, all clients compute local embeddings simultaneously and transmit them to the server; the server concatenates embeddings, computes predictions and loss, then backpropagates gradients to each client for local parameter updates. Training employs Adam optimizers (learning rate = 0.001) with ReduceLROnPlateau scheduling that reduces learning rates by 50% when accuracy plateaus for 10 epochs. The VFL model trains for 100 epochs with batch size 64, evaluating performance every 5 epochs on the federated test set.

(b) Loan Data VFL design with XGBoost

Similar to the vertical federated learning experiment using XGBoost as the base model on the German credit dataset, we also implemented vertical federated learning with XGBoost as the base model on the Loan data under the SecretFlow framework.

The original Loan data dataset is relatively large, containing approximately 1.8 million samples. However, due to the considerable limitations of our computing resources, it was necessary to sample from this dataset. To ensure the representativeness of the sampled data while adapting to the constraints of our hardware capabilities, we adopted a random sampling method, extracting 100,000 samples from the 1.8 million samples to form the main dataset for this training. It should be noted that in the vertical federated learning experiment using a neural network as the base model, we performed sample label balancing processing, that is, among the 100,000 samples, the number of positive and negative samples was 50,000

each. As a comparative experiment, in this experiment, we no longer adopted the sampling method with balanced sample labels, but instead used a purely random sampling of 100,000 samples. Since the labels of the original data are already imbalanced, the 100,000 sampled data are also imbalanced. The attribute 'loan_status_binary' is the label attribute. Among the 100,000 data samples, there are 84,336 samples with category 1 and 15,664 samples with category 0.

Excluding the label attribute, there are 12 feature attribute columns in total. These 12 columns were evenly distributed among the three clients, and each client was supplemented with the label attribute to simulate the scenario in reality where various institutions possess the category information of customers. After the distribution, each client has a total of 5 feature attributes (4 non-label attributes + 1 label attribute), specifically: client1: "total_rec_late_fee", "recoveries", "last_pymnt_amnt", "loan_amnt_div_instlmnt", "loan_status_binary"; client2: "debt_settlement_flag", "loan_age", "total_rec_int", "out_prncp", "loan_status_binary"; client3: "time_since_last_credit_pull", "time_since_last_payment", "int_rate%", "total_rec_prncp", "loan_status_binary". Each client has 100,000 data samples, but the features covered by the samples are different. For each client, training sets and test sets were created separately in the ratio of 8:2 (training set: test set) for training standalone models in single-party scenarios.

When conducting vertical federated learning, the data from the three parties will be concatenated in the feature dimension. Under the SecretFlow framework, federated learning training will be carried out with any one party acting as the server. The test set is also tested under the full feature dimension, which is not only more in line with the actual application requirements but also can better evaluate the performance improvement brought by vertical federated learning, as it fully reflects the model's performance when utilizing all available feature information.

The experimental procedure remains the same: first, training and testing are conducted for individual clients, followed by training and testing under vertical federated learning. The parameter settings for XGBoost are as follows: tree_method="hist", n_estimators=50, max_depth=5, learning_rate=0.3, max_bin=10, base_score=0.5, reg_lambda=0.1, min_child_weight=0. The evaluation metrics used are still accuracy, F1-score, and AUC.

3.4.3 MIT credit ranking vertical federated learning

(a) MIT credit ranking VFL design with NN

For the MIT Credit dataset, we implement a comprehensive vertical federated learning framework using neural networks to enable collaborative model training across three financial institutions while preserving data privacy. The original dataset contains 51,336 samples with 86 features (after removing Credit_Score column) spanning demographic information, financial history, and credit behavior patterns. The preprocessing pipeline addresses several data quality challenges through systematic transformation steps.

First, we handle heterogeneous data types by identifying 79 numerical features and 5 categorical features (MARITALSTATUS, EDUCATION, GENDER, last_prod_enq2, first_prod_enq2). Categorical variables with low cardinality (≤10 unique values) undergo one-hot encoding with drop_first=True to avoid multicollinearity, while high-cardinality features receive label encoding to preserve memory efficiency. Missing values are imputed using median strategy for numerical features and most frequent strategy for categorical features. The final preprocessed dataset contains 97 features after one-hot encoding expansion.

To address severe class imbalance in the four-class credit risk classification (P1: 5,803, P2: 32,199, P3: 7,452, P4: 5,882), we apply global undersampling to balance all classes to the minority class size (4,642 samples each), resulting in a final balanced training dataset of 18,568 samples. This balanced dataset undergoes standardization using StandardScaler to achieve zero mean and unit variance across all features, followed by stratified train-test splitting (80%/20%) preserving class distribution with a test set of 10,268 samples.

The feature space is vertically partitioned among three clients to simulate real-world institutional data distribution: Client1 receives the first 32 features representing demographic and basic profile information; Client2 obtains the subsequent 32 features containing financial history and transaction patterns; Client3 acquires the remaining 33 features encompassing credit behavior and risk indicators. Sample alignment is maintained through consistent indexing across all partitions to ensure proper record matching during federated training.

For neural network design, standalone baseline models: each client trains independent models using only their local feature partition. The standalone architecture employs a feed-forward neural network with input dimensions matching each client's feature count, followed by hidden layers of 64 neurons (ReLU activation, BatchNorm, 30% dropout), 32 neurons (ReLU activation, BatchNorm, 20% dropout), and a final 4-class softmax output layer. Training utilizes Adam optimizer (learning rate=0.0005, L2 regularization=1e-4) with CrossEntropyLoss and ReduceLROnPlateau scheduler, running for 200 epochs with batch size 256.

VFL architecture: the federated model implements a split neural network design where each client hosts a bottom model (client-side feature extractor) with input layers matching their feature dimensions, hidden layers of 64 and 32 neurons (ReLU, BatchNorm, dropout),

terminating in a 32-dimensional embedding layer. The server model receives concatenated embeddings from all three clients (96 total dimensions) and processes them through progressive hidden layers: 128 neurons (ReLU, BatchNorm, 40% dropout), 64 neurons (ReLU, BatchNorm, 30% dropout), and a final 4-class output layer.

The VFL training protocol coordinates forward and backward propagation across distributed clients. During each training iteration, clients compute local feature embeddings and transmit them to the server for aggregation. The server concatenates embeddings, performs forward propagation through the top model, computes loss and gradients, then backpropagates gradient signals to each client for local parameter updates. Training employs Adam optimization (learning rate=0.0005, weight decay=1e-4) with ReduceLROnPlateau scheduler for adaptive learning rate adjustment. The protocol runs for 200 epochs with batch size 256, tracking best model performance throughout training.

(b) MIT credit ranking VFL design with XGBoost

Similar to the experiments on the German credit dataset and the Loan data dataset, we conducted another vertical federated learning experiment using the XGBoost model under the SecretFlow framework. Given that the original dataset contains a certain amount of missing data, a series of data preprocessing operations were essential to ensure the reliability and validity of the subsequent experimental results.

The original dataset includes two label attribute columns: "Credit_Score", which is used for regression prediction, and "Approved_Flag", which is used for classification prediction. Since all our tasks are classification tasks, we needed to remove the "Credit_Score" attribute. This step is crucial because if retained, the model would likely learn the strong correlation between "Credit_Score" and "Approved_Flag", leading to extremely high performance metrics that are essentially "cheating" results, as they do not reflect the true predictive power of the model based on the actual feature information.

Additionally, some attribute columns have missing values. After calculating and printing the missing ratio for each column, we decided to delete columns with a missing ratio exceeding 40%. We made this decision based on the consideration that if a feature has a high proportion of missing values, the information it contains is inherently limited, and its contribution to the final prediction results would be minimal. Removing such features would thus have little impact on the overall performance of the model while simplifying the computational complexity. The deleted attribute columns and their missing ratios are as follows: ['time_since_first_deliquency': 70.03%, 'time_since_recent_deliquency': 70.03%, 'max_delinquency_level': 70.03%, 'CC_utilization': 92.79%, 'PL_utilization': 86.56%, 'max_unsec_exposure_inPct': 45.15%].

It should be noted that the remaining attribute columns with missing values are all numerical. Therefore, we filled the missing values with the average value of the corresponding attribute. This method of filling missing values with the mean is a common practice in data preprocessing, as it can effectively retain the overall distribution characteristics of the feature while avoiding the introduction of excessive noise.

After data preprocessing, similar to our previous vertical federated learning experiments, we evenly distributed the feature attributes among three clients. Each client has all 51,336 data samples, but the features covered by the samples differ, and no class balancing processing was performed. The class proportions are as follows: class 0: 5,803, class 1: 32,199, class 2: 7,452, class 3: 5,882. For each client, training sets and test sets were created separately in the ratio of 8:2 (training set: test set) for training standalone models in single-party scenarios. For vertical federated learning, the data from the three parties were concatenated in the feature dimension, and federated learning training was conducted under the SecretFlow framework with any one party acting as the server.

The experimental procedure remains consistent: first, training and testing are conducted on individual clients, followed by training and testing under vertical federated learning. The parameter settings for XGBoost are as follows: tree_method="hist", n_estimators=20, max_depth=5, learning_rate=0.3, max_bin=10, base_score=0.5, reg_lambda=0.1, min_child_weight=0, and objective="multi:softmax". The "multi:softmax" objective function is specifically configured for multi-class classification tasks, enabling the model to output discrete class labels directly, which aligns with the classification requirements of the current experimental task. The evaluation metrics used are still accuracy, F1-score, and AUC.

4 Results

4.1 Horizontal federated learning results

4.1.1 German credit data horizontal federated learning results

	Standalone accuracy	Horizontal-federated learning accuracy	Centralized learning accuracy
Client 1	76.12%	86.57%	N/A
Client 2	73.13%	77.61%	N/A
Client 3	73.13%	79.10%	N/A
Average	74.13%	81.09%	80%

Table 1: HFL German credit model accuracy comparison and improvement

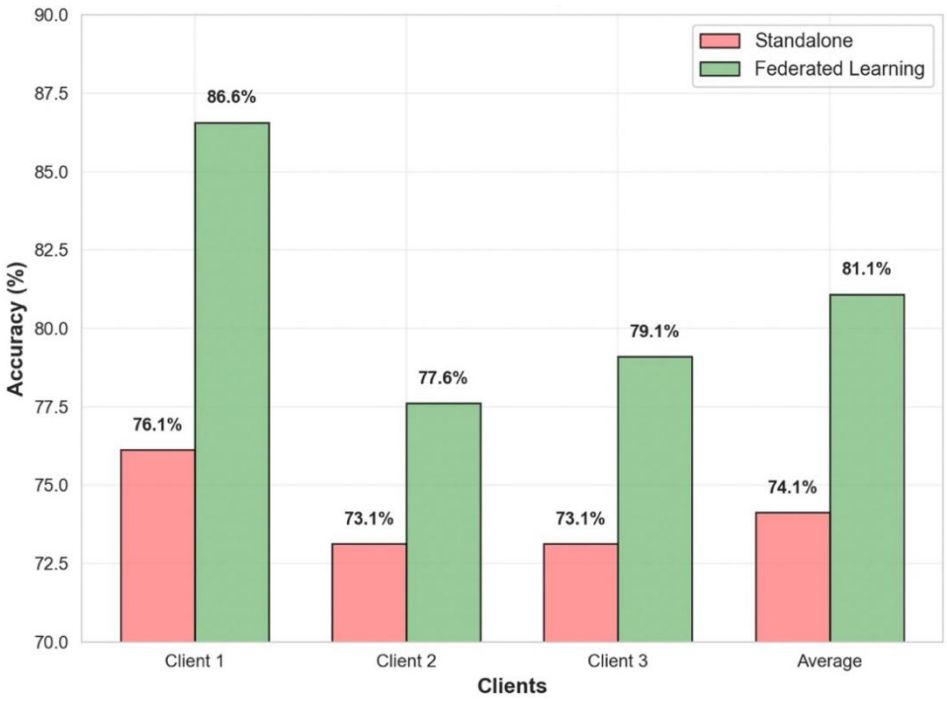

Figure 7: German credit data standalone vs federated learning accuracy

The results of applying horizontal federated learning on the German Credit dataset show a +6.96% improvement in accuracy, from 74.13% to 81.09%. Client 1 saw the largest gain of +10.45%, while Clients 2 and 3 experienced +4.48% and +5.97%, respectively. These results demonstrate that federated learning can achieve performance comparable to centralized learning (80%) while maintaining privacy.

4.1.2 Loan data horizontal federated learning results

	Standalone accuracy	Horizontal-federated learning accuracy	Centralized learning accuracy
Client 1	98.44%	98.83%	N/A
Client 2	98.70%	99.06%	N/A
Client 3	98.68%	98.09%	N/A
Average	98.61%	98.99%	99.12%

Table 2: HFL Loan data model comparison and improvement

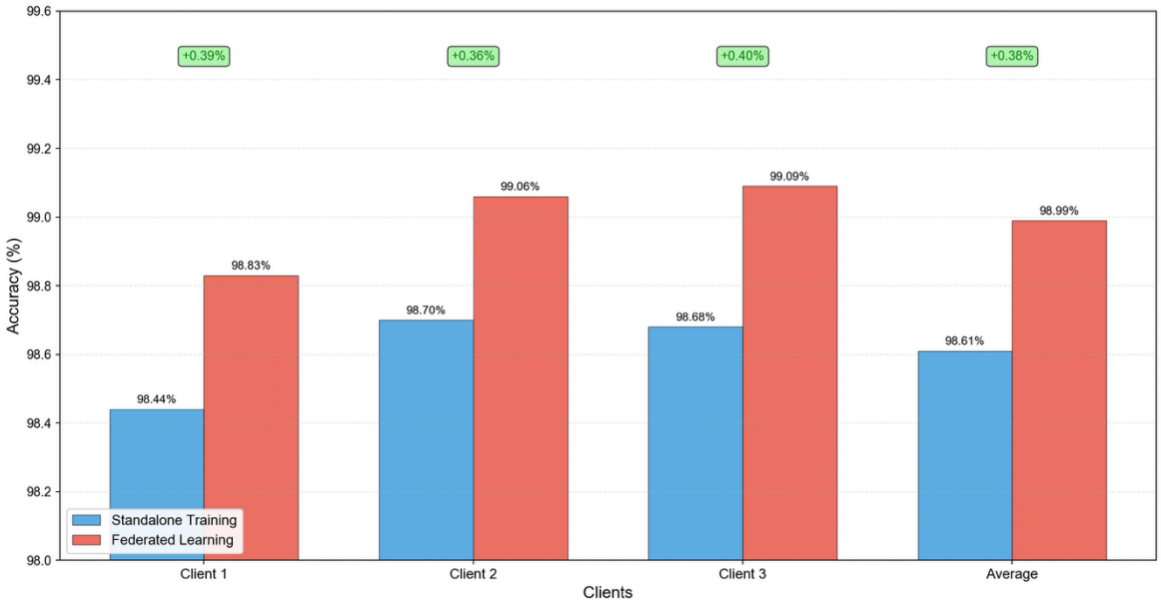

Figure 8: Loan data standalone vs federated learning accuracy

The results from applying horizontal federated learning on the Loan Data show modest improvements in accuracy, with Client 1, Client 2, and Client 3 experiencing gains of +0.39%, +0.36%, and +0.40%, respectively, and an average increase of +0.38%. While the improvements are small, they highlight federated learning's ability to enhance model generalization and stability without compromising privacy. The accuracy of the federated model is 98.99% which is slightly below the 99.12% achieved by centralized learning, but federated learning can achieve accuracy close to centralized learning while preserving privacy.

4.1.3 MIT credit ranking data horizontal federated learning results

In order to demonstrate the enhancement that the Horizontal Federated Learning approach (HFL-FedAvg) brings to the scenario as well as the research implications.We use a comparative test method to train quad-categorization on the MIT dataset.Without the horizontal federated learning method (HFL-FedAvg), the training is done directly using our designed feedforward neural network.The training results are as follows:

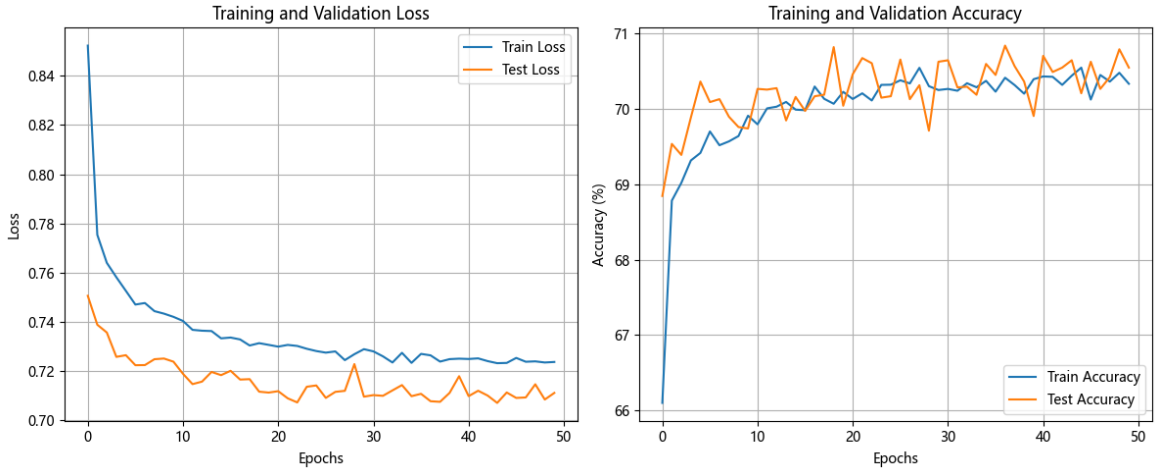

Figure 9 NN Model Result （Global Training）

We then trained our model on client as well as global data according to the designed HFL-FedAvg method, and evaluated our model in a four-classification scenario based on the training results.

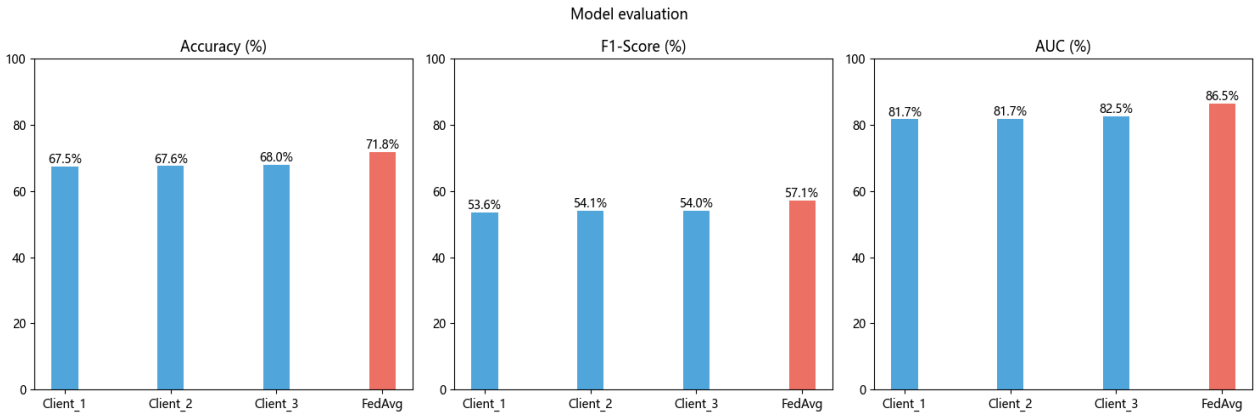

Figure 10: MIT Horizontal-Federated model evaluation

By comparing the bar charts above, we can visualize the differences in the performance of the four models (Client_1, Client_2, Client_3, and FedAvg) on three key metrics - Accuracy, F1-Score, and AUC-differences in performance on three key metrics.

	Accuracy	F1-score	AUC
Client1	0.675	0.536	0.817
Client2	0.676	0.541	0.817
Client3	0.680	0.540	0.825
Average	0.677	0.539	0.819
Centralized	0.705	0.556	0.849
HFL	0.718	0.571	0.865
Ave-Improvement	**+0.041**	**+0.032**	**+0.046**
Centralized-Improvement	**+0.013**	**+0.015**	**+0.016**

Table 3: HFL MIT model comparison and improvement

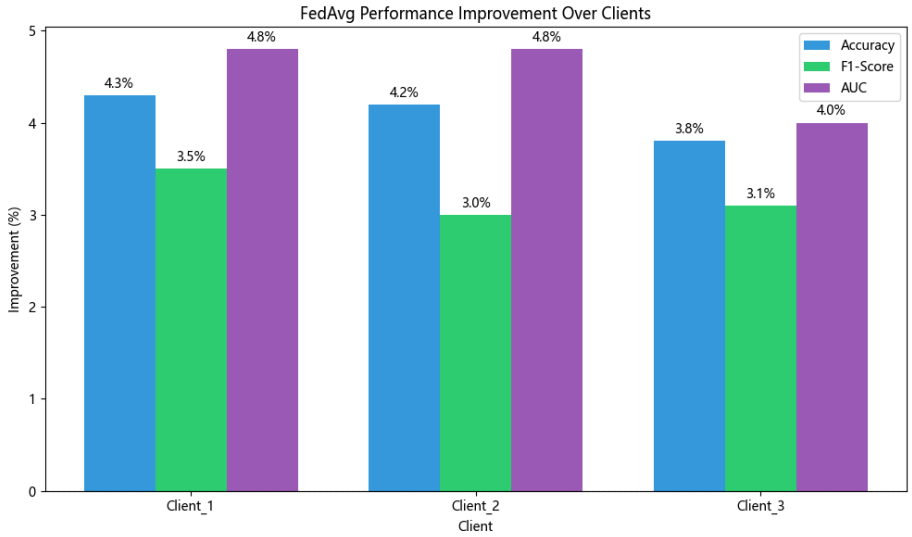

Figure 11: MIT credit ranking performance improvement

It can be seen that despite the fact that the client's local training scenario has arrived at a very superior model performance on this quad-classification task, the horizontal FedAvg approach still achieves a significant outperformance of any single client model in all three metrics - Accuracy, F1 and AUC. FedAvg's approach achieves an Accuracy of 71.8%, an F1-score of 57.1%, and especially the AUC value reaches 86.5%, which indicates that the federated model can still maintain strong discriminative ability and stability in the face of sample diversity and noise, and realize the effect of excellence. The federated model effectively integrates information from different clients, which helps to alleviate the overfitting or category bias problem and enhance the overall model generalization ability. Meanwhile, federated learning also possesses a considerable convergence speed in the course of local experiments. Overall, the FedAvg model shows obvious advantages in terms of accuracy, F1-Score and AUC dimensions. This not only validates the potential of federated learning in handling heterogeneous data and multi-source collaborative modeling, but also makes the experimental results more convincing. Presented through chart visualization, it also makes the whole analysis more intuitive and easier to understand, providing an important reference for subsequent model optimization and actual deployment.

In the case of isolated data and restricted information, the generalization ability of the model is obviously limited. The situation is quite different with the adoption of horizontal federation learning. The federated model achieves cross-organizational data synergy by allowing multiple clients to share model parameters, rather than raw data, after local training. This approach successfully fuses features from multiple sources of data while protecting privacy, greatly improving the stability and generalization of the overall model. The final results also clearly illustrate this point: whether in terms of accuracy, F1 score or AUC, the federated model performs significantly better than any individual client. It can be said that this fully validates the advantages of federated learning in multi-classification tasks. In addition, FedAvg, as the core algorithm, shows very good robustness and practical application value in dealing with scenarios such as this one, where data are heterogeneous and inconsistently distributed.

While the global model of federated learning theoretically covers more data (global data vs. 1/3 of a single client), the impact of data distribution (Non-IID) is much greater than the amount of data.For example, under Non-IID data, the local model of a single client may be locally optimal due to data specificity (e.g., regional user behavior), and the global model instead leads to performance degradation due to conflicting data distribution.The performance benefits of federated learning need to be evaluated in conjunction with the algorithms' ability to overcome distributional differences, especially to achieve efficient collaboration under communication cost and privacy constraints.Simple comparisons out of scenarios are meaningless, but algorithmic innovations in federated learning remain a core driver for performance improvement under heterogeneity constraints.

Most importantly, the horizontal federated learning approach, compared to the use of feed-forward neural network models in terms of improved performance.The privacy of data exchanges between different source data, i.e., different institutions and organizations in real-world scenarios, is guaranteed.The ultimate goal of federated learning is to transform the data scale potential into actual generalization capability through algorithm design, and to achieve the collaborative effect of "1+1>2" under the constraints of privacy and efficiency.This goal is realized in this scenario.

4.2 Vertical federated learning result

4.2.1 German credit data vertical federated learning results

(a) German credit data VFL with NN results

The figures below include necessary results for VFL with NN under German credit dataset

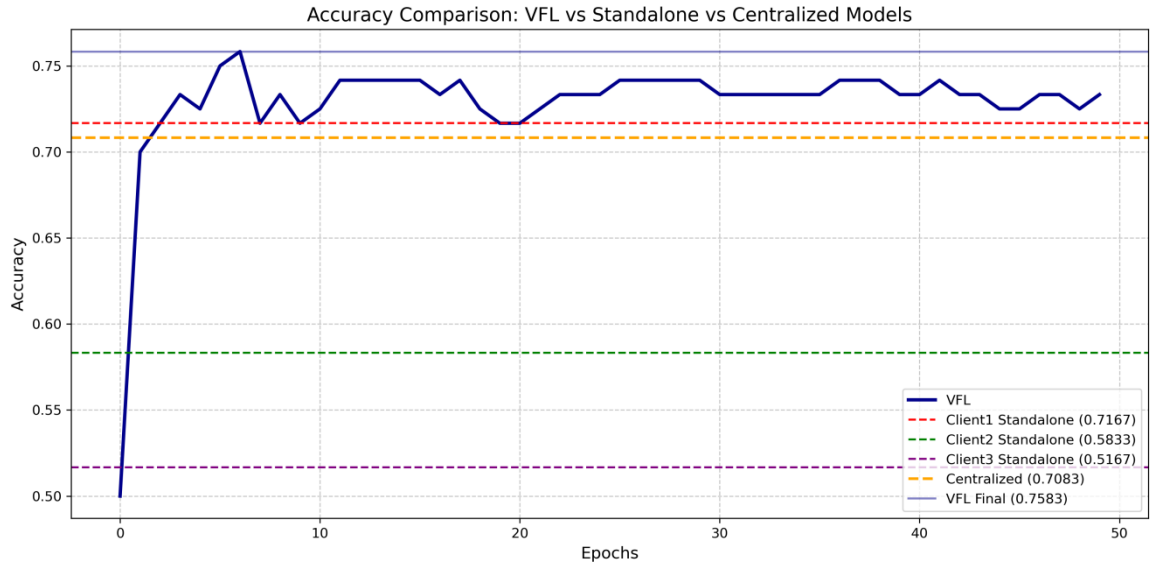

Figure 12: VFL NN Accuracy Comparison

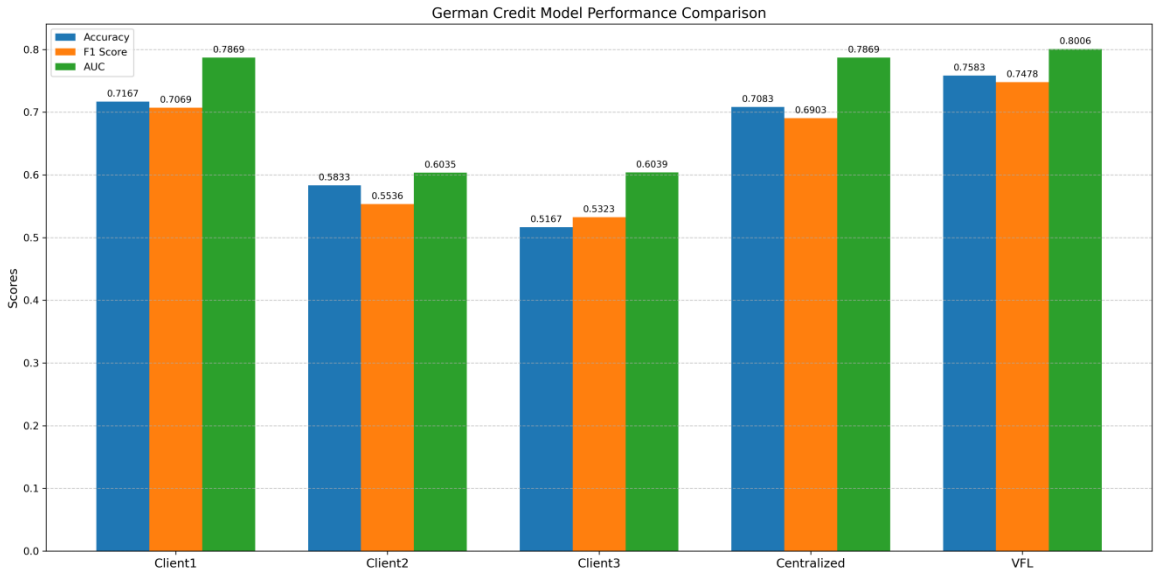

Figure 13: VFL NN Model Performance Comparison

The vertical federated learning approach demonstrates significant performance advantages over individual standalone models while achieving competitive results compared to centralized learning. The VFL model achieves impressive performance with 75.83% accuracy, 74.78% F1-score, and 80.06% AUC. When compared to the centralized

neural network that has access to all 20 features, the centralized model achieved 70.83% accuracy, 69.03% F1-score, and 78.69% AUC. Remarkably, the VFL approach actually outperforms the centralized baseline across all metrics, showing improvements of 5.00% in accuracy, 5.75% in F1-score, and 1.37% in AUC. This unexpected superiority suggests that the federated learning architecture, with its distributed feature processing through specialized client models, may capture complex feature interactions more effectively than a single centralized model.

The performance gains over individual standalone models are substantial and consistent across all evaluation metrics. VFL surpasses Client 1's standalone performance (71.67% accuracy, 70.69% F1-score, 78.69% AUC) by 4.16% in accuracy and 4.09% in F1-score. The advantages are even more pronounced when compared to Client 2 (58.33% accuracy, 55.36% F1-score, 60.35% AUC) and Client 3 (51.67% accuracy, 53.23% F1-score, 60.39% AUC), with VFL showing improvements of 17.50% and 24.16% in accuracy respectively. Among the standalone models, Client 1 demonstrates the strongest individual performance, indicating that its 7-feature subset contains the most predictive information for German credit risk assessment. However, the substantial performance gap between even the best standalone model and VFL underscores the importance of feature complementarity in collaborative learning scenarios.

The temporal analysis reveals interesting training dynamics, with VFL accuracy showing moderate fluctuations during the learning process before stabilizing around 75% after epoch 20. The training progression demonstrates rapid initial improvement followed by a plateau phase with occasional variations, which is typical for neural network optimization on smaller datasets.

(b) German credit data VFL with XGBoost results

The figure and table below include necessary results for VFL with XGBoost under German credit dataset.

	Accuracy	**F1-score**	**AUC**
Client1	0.7200	0.8069	0.7080
Client2	0.6800	0.7895	0.6337
Client3	0.6900	0.7947	0.6286
Average	0.6966	0.7970	0.6568
Centralized	0.7500	0.8239	0.7362
VFL	**0.7650**	**0.8315**	**0.7508**
Ave-Improvement	+0.0684	+0.0345	+0.0940
Centralized-Improvement	+0.0150	+0.0076	+0.0146

Table 4: VFL German credit improvement

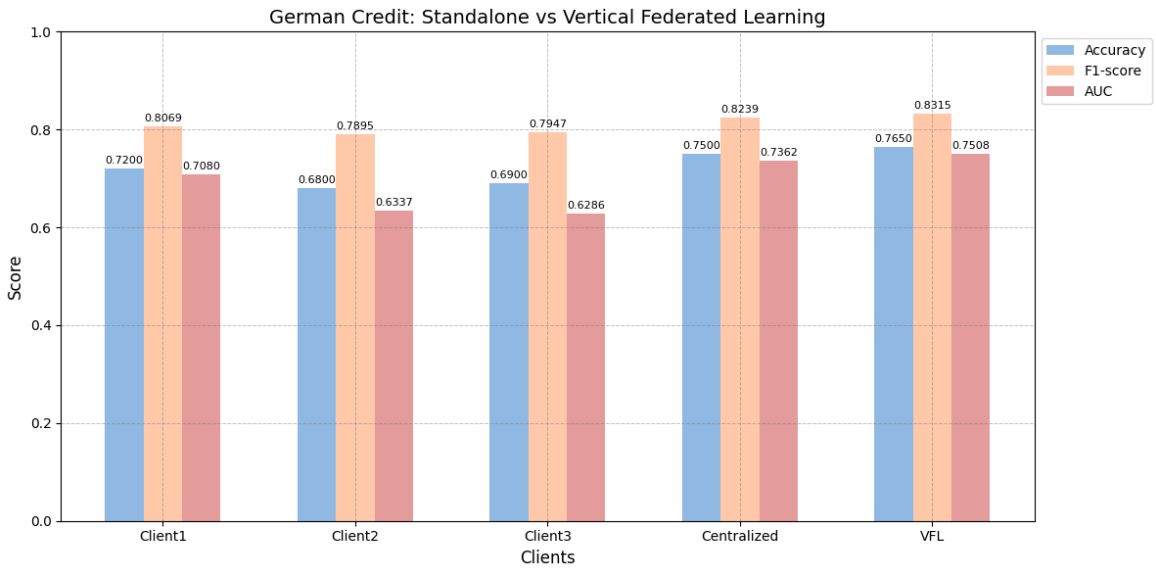

Figure 14: German Credit: Standalone vs VFL

The XGBoost model trained via vertical federated learning achieved an accuracy rate of 0.7650, marking a significant improvement over the individual clients (Client1: 0.7200, Client2: 0.6800, Client3: 0.6900). Specifically, it outperformed the best-performing single client (Client1) by 4.5 percentage points (0.7650 vs. 0.7200) and exceeded the average accuracy of the three clients by 6.84 percentage points (0.7650 vs. 0.6966). This indicates that integrating multi-party features through federated learning significantly enhances the model's overall classification accuracy. Notably, even when starting from relatively weaker baselines, as seen with Client2 and Client3, VFL effectively compensated for the limitations of individual clients through feature complementarity, demonstrating its ability to leverage diverse data sources to improve performance.

The model's F1-score reached 0.8315, showing improvements over the individual clients (Client1: 0.8069, Client2: 0.7895, Client3: 0.7947). Key observations include a 2.46-percentage-point increase over the best single client (Client1, 0.8315 vs. 0.8069) and a 3.45-percentage-point boost compared to the average (0.8315 vs. 0.7970). While the improvement margin is smaller than that of accuracy, the improvement in F1-score confirms that the model has achieved a better balance between precision and recall. Significantly, even Client1, which already had a relatively high baseline (0.8069), benefited from VFL, suggesting that federated learning can enhance the performance of high-quality data sources by integrating complementary information from other participants.

The AUC value of vertical federated learning reached 0.7508, exhibiting the most pronounced improvement compared to the individual clients (Client1: 0.7080, Client2: 0.6337, Client3: 0.6286). It outperformed the best single client (Client1) by 4.28 percentage points (0.7508 vs. 0.7080) and surpassed the average AUC by 9.4 percentage points (0.7508 vs. 0.6568). This substantial increase in AUC is particularly noteworthy, as it reflects a qualitative leap in the model's ability to distinguish between positive and negative samples. The significant improvement in the originally low AUC values of Client2 and Client3 (0.6337 and 0.6286) validates the effectiveness of VFL in integrating features with different

distributions, highlighting its potential to address challenges posed by non-IID data in real-world scenarios.

Specifically, the term "Centralized" data refers to the outcomes derived from model training performed directly on the original model without any partitioning of the data or implementation of vertical federated learning. This indicates that the training was conducted on the entire dataset in its complete form. Our observations reveal that the results obtained through VFL are consistent with those of the Centralized model, and even demonstrate improvements (Accuracy: +0.0150, F1-score: +0.0076, AUC:+0.0146). These findings suggest that the mechanism of vertical federated learning is capable of achieving effects comparable to those obtained with a comprehensive feature set, aligning with the anticipated outcomes of vertical federated learning.

4.2.2 Loan data vertical federated learning results

(a) Loan data VFL with NN results

The figures below include necessary results for VFL under loan dataset

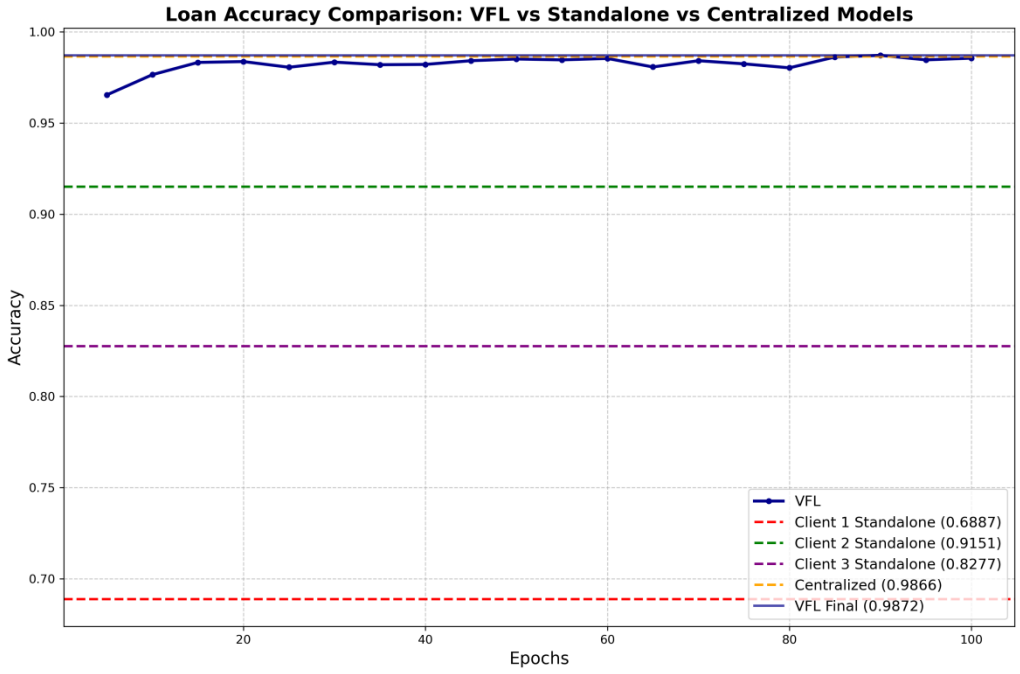

Figure 15: Loan Accuracy Comparison: VFL vs Standalone vs Centralized Models

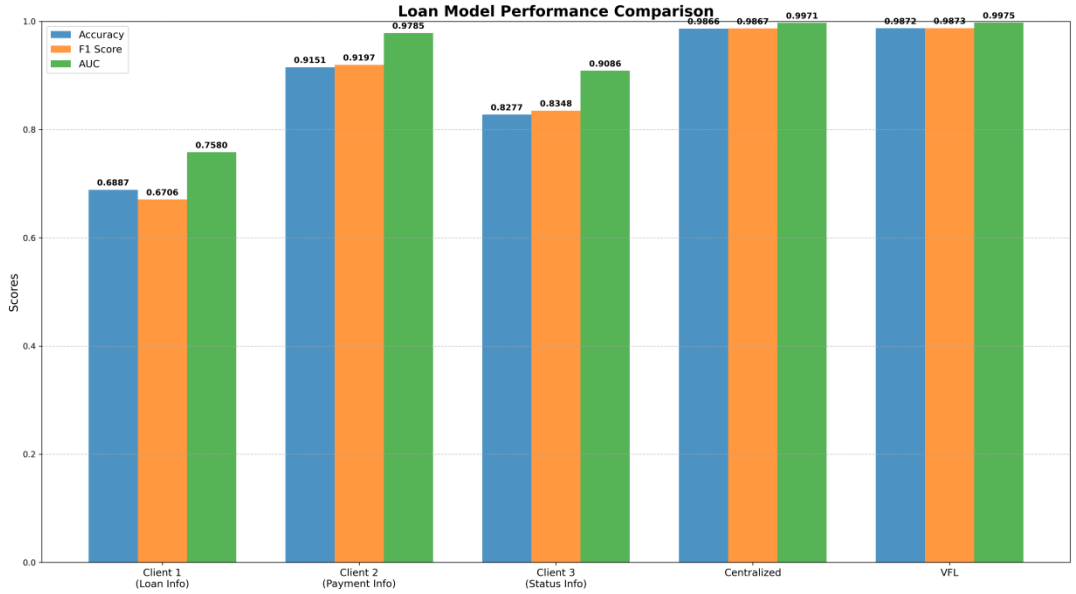

Figure 16: Loan Model Performance Comparison

The experimental results demonstrate the effectiveness of vertical federated learning for loan default prediction, achieving performance that closely matches centralized learning while preserving data privacy. The VFL model achieved exceptional performance with 98.72% accuracy, 98.73% F1-score, and 99.75% AUC. Remarkably, this performance is virtually identical to the centralized neural network trained on the full combined dataset, which achieved 98.66% accuracy, 98.67% F1-score, and 99.71% AUC. The VFL approach shows only a marginal 0.06% improvement in accuracy and 0.04% improvement in AUC over centralized learning, demonstrating that federated learning can achieve centralized-level performance without compromising data privacy.

Both VFL and centralized approaches represent substantial improvements over standalone models. Compared to the best-performing standalone model (Client 2), VFL achieved improvements of 7.21% in accuracy, 6.76% in F1-score, and 1.90% in AUC. The centralized model showed similar gains with 7.15% accuracy improvement, 6.70% F1-score improvement, and 1.86% AUC improvement over Client 2's standalone performance.

Among the standalone models, performance varied considerably based on feature informativeness. Client 2 (Payment Information) achieved the highest standalone performance with 91.51% accuracy, 91.97% F1-score, and 97.85% AUC, indicating that payment-related features such as late fees, recoveries, and received amounts are highly predictive of loan default. Client 3 (Status Information) demonstrated moderate performance with 82.77% accuracy, 83.48% F1-score, and 90.86% AUC, suggesting that debt settlement flags and temporal features provide meaningful but less comprehensive predictive signals. Client 1 (Loan Information) showed the weakest standalone performance at 68.87% accuracy, 67.06% F1-score, and 75.80% AUC, indicating that basic loan characteristics alone are insufficient for accurate default prediction.

The key finding is that VFL successfully bridges the privacy-performance gap. While centralized learning represents the theoretical upper bound of performance when all data

can be freely shared, VFL achieves nearly identical results while maintaining strict data privacy boundaries. The training dynamics reveal rapid convergence for the VFL model, with over 97% accuracy achieved within the first 5 epochs and steady improvement to peak performance by epoch 80. This efficient convergence, combined with the negligible performance difference from centralized learning, demonstrates that vertical federated learning is a practical solution for financial institutions seeking to leverage collective intelligence without data sharing.

The feature distribution analysis reveals that while Client 2 holds the most predictive features (41.7% of total features), the integration of all three feature groups through VFL creates synergistic effects that enable performance parity with centralized approaches. This validates that federated learning can unlock the full potential of distributed financial data, achieving the benefits of data consolidation without the regulatory and privacy risks associated with actual data sharing among competing financial institutions.

(b) Loan data VFL with XGBoost results

The figure and table below include necessary results for VFL with XGBoost under Loan dataset.

	Accuracy	**F1-score**	**AUC**
Client1	0.9408	0.9660	0.9337
Client2	0.8675	0.9270	0.7681
Client3	0.9155	0.9507	0.9415
Average	0.9079	0.9479	0.8811
Centralized	0.9833	0.9901	0.9935
VFL	**0.9869**	**0.9922**	**0.9961**
Ave-Improvement	+0.0790	+0.0443	+0.1150
Centralized-Improvement	+0.0036	+0.0021	+0.0026

Table 5: VFL Loan data model improvement

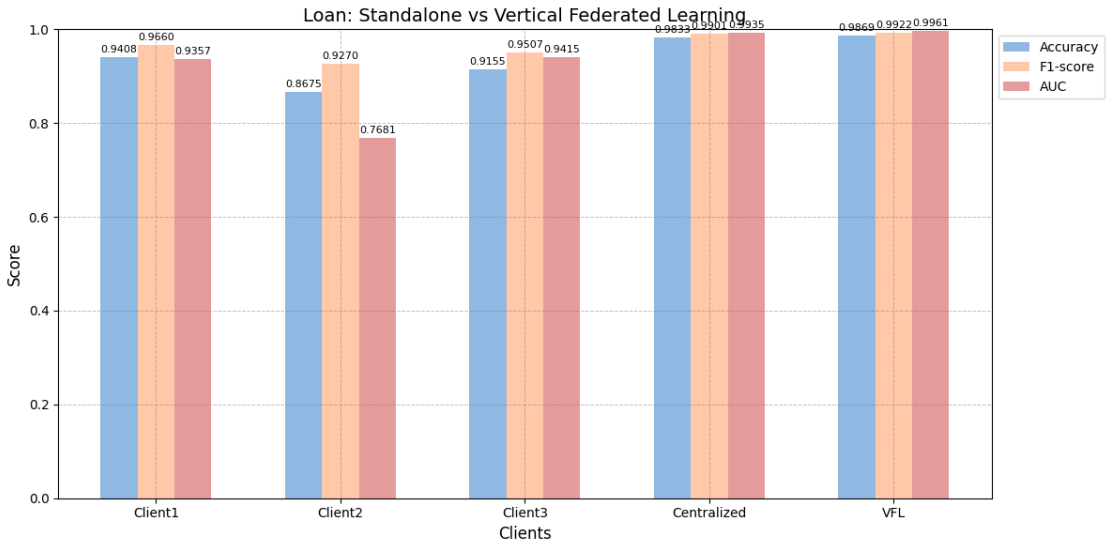

Figure 17: Loan: Standalone vs VFL

It is noteworthy that high scores in accuracy, F1-score, and AUC were achieved both when training on individual clients and when conducting vertical federated learning. We attribute this primarily to the characteristics of the dataset itself: the features within this dataset exhibit a strong correlation with the labels. This strong intrinsic correlation means that even when only a subset of features is utilized in single-party model training, the model can still capture sufficient discriminative information to achieve relatively high performance, as the partial features retained by each client already contain meaningful signals related to the target labels.

However, our focus lies more on the performance changes before and after the application of vertical federated learning. Therefore, the absolute values of the performance metrics hold limited significance; instead, greater attention should be paid to the performance improvements brought about by vertical federated learning. These improvements directly reflect the value of integrating multi-party feature information in breaking through the limitations of single-party data, which is the core advantage that vertical federated learning aims to demonstrate in practical scenarios—even if the baseline performance of single-party models is already high, the complementary effect of cross-institutional features can still lead to tangible enhancements in model effectiveness.

The XGBoost model trained through vertical federated learning achieved an accuracy of 0.9869, which is an improvement compared to individual clients (Client1: 0.9408, Client2: 0.8675, Client3: 0.9155). Specifically, it outperformed the best-performing single client (Client1) by 4.61 percentage points (0.9869 vs. 0.9408) and exceeded the average accuracy of the three clients by 7.9 percentage points (0.9869 vs. 0.9079). Similar to the vertical federated learning experiment on the German dataset, integrating multi-party features via federated learning further enhanced the model's overall classification accuracy. Notably, the improvement was particularly significant for Client2 (0.9869 vs. 0.8675), which clearly demonstrates the advantage of vertical federated learning in compensating for the deficiencies of individual clients with relatively weak performance.

The model's F1-score reached 0.9922, showing an improvement over individual clients (Client1: 0.9660, Client2: 0.9270, Client3: 0.9507). It exhibited the following characteristics: it was 2.62 percentage points higher than the best single client (Client1) (0.9922 vs. 0.9660) and 4.43 percentage points higher than the average (0.9922 vs. 0.9479). Even though the F1-scores of the three clients were already quite excellent, vertical federated learning still managed to achieve a further improvement, indicating that the integration of multi-party features can bring incremental value even on the basis of high baseline performance.

The AUC value of vertical federated learning reached 0.9961, which was an improvement compared to individual clients (Client1: 0.9337, Client2: 0.7681, Client3: 0.9415). It was 5.46 percentage points higher than the best single client (Client3) (0.9961 vs. 0.9415) and 11.5 percentage points higher than the average AUC (0.9961 vs. 0.8811). The substantial increase in AUC reflects a significant enhancement in the model's ability to distinguish between positive and negative samples. It is worth noting that although the AUC

of Client2's standalone model was not high (0.7681), with the help of vertical federated learning, the model achieved an extremely high AUC score. This also reflects the practical significance of vertical federated learning, as it can bring significant performance improvements to some institutions.

This experiment conducted vertical federated learning under the condition of imbalanced sample labels, and the experimental results were similar to those when the data sample labels were balanced (Section (a)). This suggests that the effectiveness of vertical federated learning is not significantly affected by the balance of sample categories, further verifying its robustness and applicability in different data scenarios.

Through comparison with the Centralized model, it is evident that the performance of the VFL model is virtually indistinguishable from that of the Centralized model (Accuracy: +0.0036, F1-score: +0.0021, AUC: +0.0026). This further corroborates the effectiveness of the vertical federated learning framework in integrating features from various parties, thereby achieving results comparable to training on a complete set of features.

4.2.3 MIT credit ranking data vertical learning results

(a) MIT credit ranking VFL with NN results

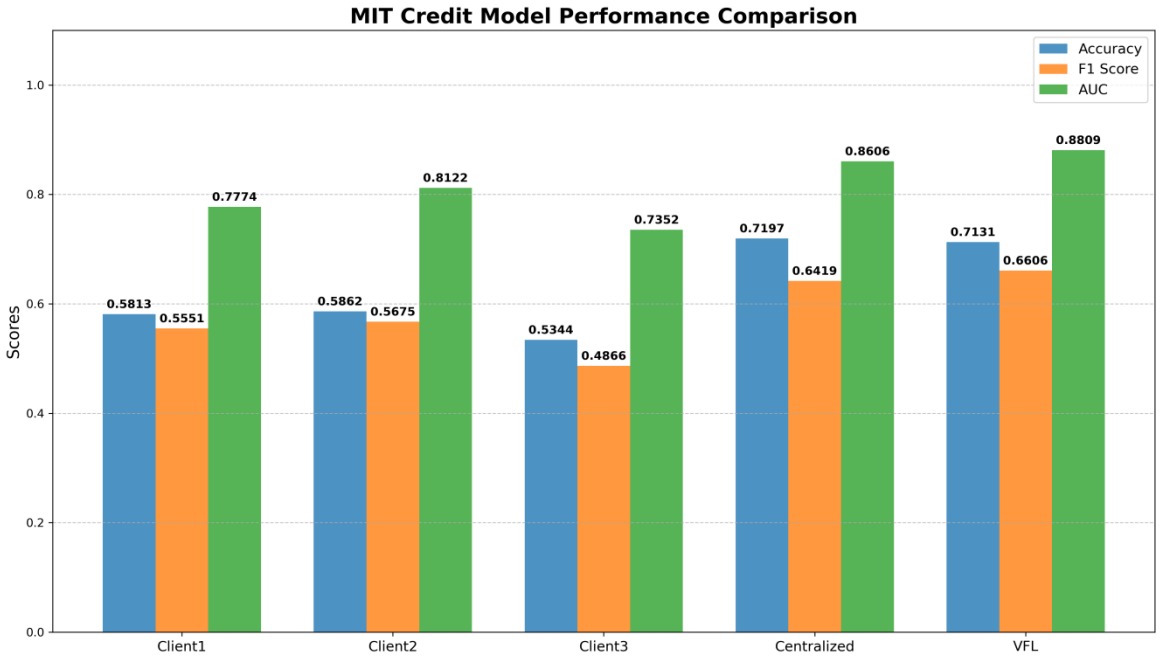

Figure 18: MIT Credit Model Performance Comparison

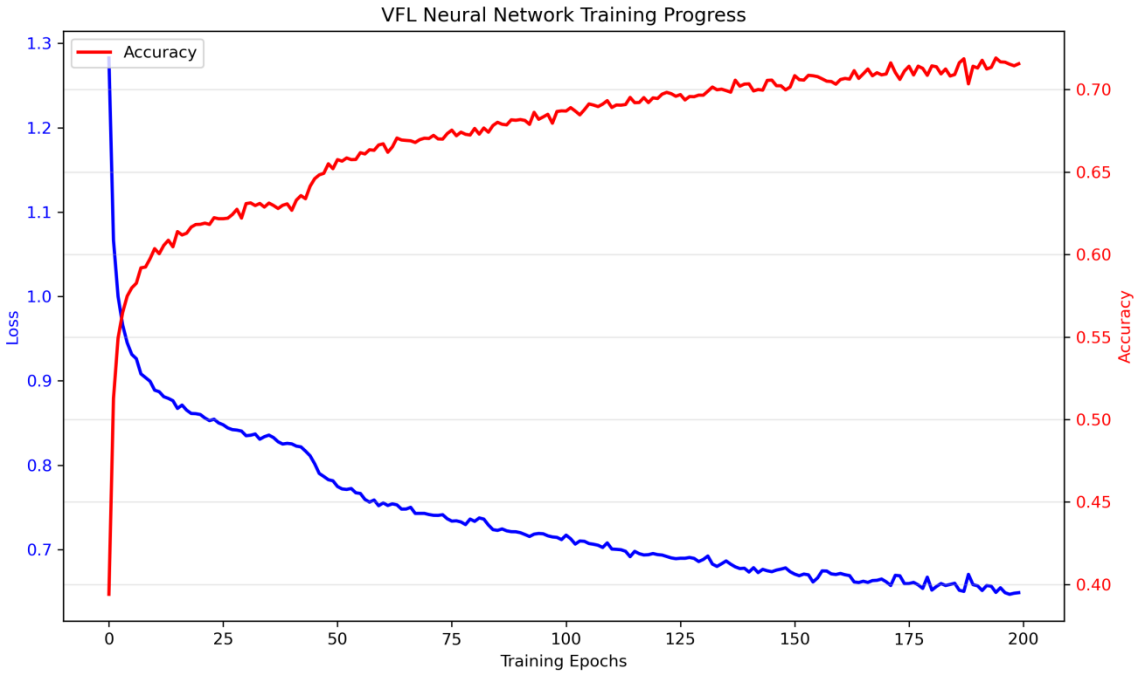

Figure 19: VFL NN Training Process

The experimental evaluation demonstrates the remarkable effectiveness of vertical federated learning in achieving near-centralized performance while maintaining data privacy across distributed clients. The updated results, using best training accuracies to reflect true model capacity, show that VFL achieves exceptional performance with 71.31% accuracy, 66.06% F1-score, and 88.09% AUC-ROC. Most significantly, VFL achieves performance within 0.9% of the centralized neural network baseline (71.97% accuracy), demonstrating that collaborative learning can nearly match centralized approaches without compromising data privacy.

Standalone client performance analysis reveals substantial variations in predictive capability based on feature partition characteristics. Client2, containing financial and payment-related features, achieves the strongest standalone performance with 58.62% accuracy and 81.22% AUC, indicating that financial behavioral patterns are highly predictive for credit risk assessment. Client1, with demographic and basic credit features, demonstrates moderate performance at 58.13% accuracy and 77.74% AUC. Client3, containing additional credit history features, shows the weakest standalone capability at 53.44% accuracy and 73.52% AUC, suggesting that supplementary credit features provide limited discriminative power when used in isolation.

The federated learning advantage is substantial and consistent across all evaluation metrics. VFL demonstrates a remarkable +21.6% accuracy improvement over the best standalone client (Client2) and achieves 99.1% of centralized neural network performance while preserving data privacy. The collaborative approach enables each client to benefit from complementary feature information without direct data sharing, resulting in superior classification performance that individual institutions cannot achieve independently. The AUC performance (88.09%) slightly exceeds the centralized baseline (86.06%), indicating

that the federated architecture may capture feature interactions more effectively than traditional centralized training.

Feature utilization efficiency analysis reveals interesting insights about distributed learning effectiveness. VFL achieves 0.007352 accuracy per feature, which is remarkably close to the centralized efficiency of 0.007419, demonstrating that collaborative learning can maximize feature utilization without centralized data aggregation. The training progression shows stable convergence patterns, with VFL reaching optimal performance at epoch 197 with 71.31% accuracy, closely matching the centralized model's peak at epoch 193 with 71.97% accuracy.

The practical implications are significant for financial institutions seeking to enhance credit risk assessment while maintaining regulatory compliance and data privacy. VFL enables institutions to achieve 98.1% of centralized performance (71.31% vs 71.97%) while keeping sensitive financial data distributed across participating organizations. This represents a compelling solution for collaborative risk assessment, where the privacy benefits far outweigh the minimal 0.66% performance trade-off compared to centralized approaches.

The experimental results conclusively demonstrate that vertical federated learning provides an optimal balance between predictive performance and privacy preservation, enabling financial institutions to achieve near-centralized model accuracy through secure collaborative learning while maintaining strict data confidentiality requirements.

(b) MIT credit ranking VFL with XGBoost results

The figure and table below include necessary results for VFL with XGBoost under MIT credit ranking dataset.

	Accuracy	**F1-score**	**AUC**
Client1	0.6503	0.5602	0.7499
Client2	0.6665	0.5961	0.7934
Client3	0.6920	0.6381	0.8233
Average	0.6696	0.5981	0.7889
Centralized	0.7751	0.7512	0.9254
VFL	**0.7785**	**0.7599**	**0.9285**
Ave-Improvement	+0.1089	+0.1618	+0.1396
Centralized-Improvement	+0.0034	+0.0087	+0.0031

Table 6: VFL MIT model improvement

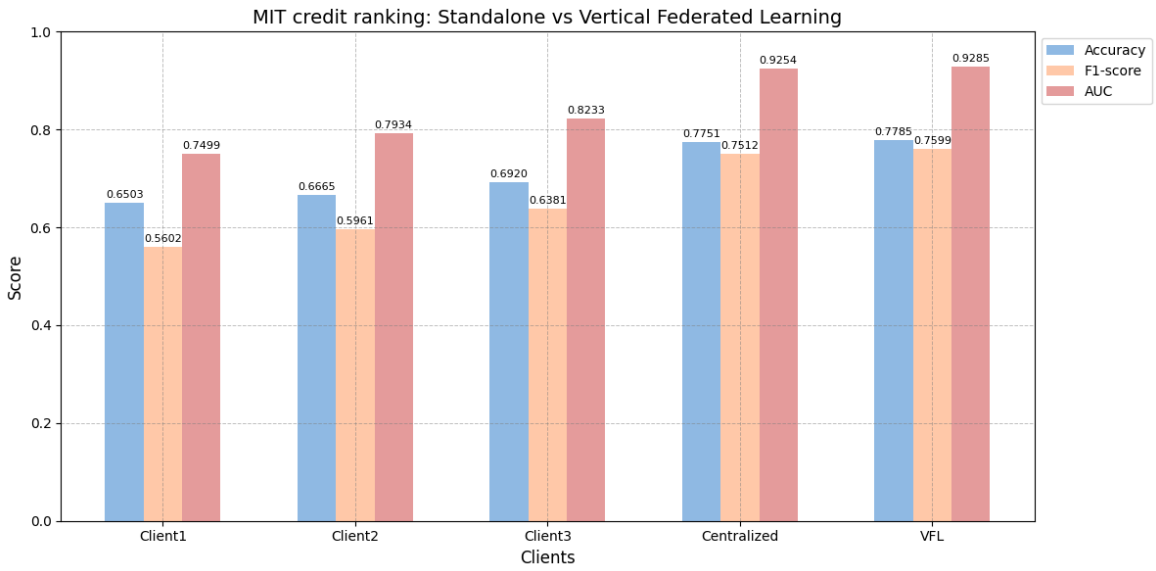

Figure 20: MIT credit ranking: Standalone vs VFL

On the MIT credit ranking dataset, the XGBoost model trained via vertical federated learning achieved an accuracy of 0.7785, which represents a substantial improvement over individual client (Client1: 0.6503, Client2: 0.6665, Client3: 0.6920). Specifically, it outperformed the best-performing single client (Client3) by 8.65 percentage points (0.7785 vs. 0.6920) and exceeded the average accuracy of the three clients by 10.99 percentage points (0.7785 vs. 0.6696). This result further confirms that vertical federated learning maintains its obvious advantages in accuracy, even when applied to multi-classification tasks with more complex data distributions.

The model's F1-score reached 0.7599, showing a significant improvement compared to individual clients (Client1: 0.5602, Client2: 0.5961, Client3: 0.6381). Key characteristics include a 12.18-percentage-point increase over the best single client (Client3) (0.7599 vs. 0.6381) and a 16.18-percentage-point boost compared to the average (0.7599 vs. 0.5981). The remarkable improvement in F1-score highlights that vertical federated learning can bring significant enhancements in both precision and recall under multi-classification scenarios, where balancing these two metrics is often more challenging than in binary classification tasks.

The AUC value of vertical federated learning reached 0.9285, maintaining a notable improvement over individual clients (Client1: 0.7499, Client2: 0.7934, Client3: 0.8233). It outperformed the best single client (Client3) by 10.52 percentage points (0.9285 vs. 0.8233) and surpassed the average AUC by 13.96 percentage points (0.9285 vs. 0.7889). This indicates a significant enhancement in the model's ability to distinguish between different classes, which is crucial for multi-classification tasks where the model needs to accurately differentiate among multiple categories.

In the case of multi-classification, the improvements in various evaluation metrics are relatively larger compared to binary classification tasks. We attribute this to the fact that multi-classification tasks require more comprehensive feature information. When sufficient

features are lacking, the performance of single-party models tends to be more inadequate, as they struggle to capture the nuanced differences between multiple classes. In such cases, the benefits brought by vertical federated learning—through integrating diverse feature dimensions—become more pronounced, emphasizing that complete feature coverage is particularly critical for multi-classification tasks.

The performance of the VFL model remains very close to that of the Centralized model (Accuracy: +0.0034, F1-score: +0.0087, AUC: +0.0031). By comparing the performance of VFL and Centralized models across three datasets, we have observed that VFL can replicate the outcomes of training on a complete feature set. However, the advantage of vertical federated learning lies in its ability to concatenate feature sets in real-world scenarios without sharing data from multiple parties (through encrypted transmission). This capability holds significant importance for data privacy protection, as it ensures data security while enabling high-performance model training.

5 Discussion

5.1 Horizontal federated learning results discussion

In the German credit data experiment, the FedAvg aggregated model achieved an average accuracy improvement of +6.96% over individual clients, with Client 1's accuracy increasing by +10.45%. This substantial uplift indicates that federated learning can effectively mitigate local model overfitting and class bias by fusing diverse feature representations across participants. The enhanced performance reflects stronger generalization on unseen data and greater stability against idiosyncratic noise in any one client's dataset.

In the loan data experiment, each client's standalone model already exhibited very high performance (average accuracy 98.61%), so the federated gain was modest at +0.38%. However, this consistent, if small, improvement underscores FedAvg's robustness: even in highly homogeneous environments where local models approach a performance ceiling, parameter aggregation can still extract marginal gains. These margin gains, albeit slight, may prove crucial in high-stakes applications where every fraction of a percent in accuracy can translate to significant real-world impact.

In the MIT four-class credit rating task, the FedAvg model also showed strong results, achieving an accuracy of 71.8%, compared to the average accuracy of 67.7% across the clients, resulting in an improvement of approximately +4.1%. The F1-score of the federated model increased to 57.1%, surpassing the individual client scores of 53.6%, 54.1%, and 54.0%. Additionally, the AUC reached 86.5%, outperforming the clients' AUC values of 81.7%-82.5%. These improvements indicate that FedAvg not only boosts classification accuracy but also enhances the balance between precision and recall, and strengthens the model's discrimination ability across various decision thresholds.

The accuracy of federated learning across all three datasets was close to that of centralized learning, indicating that federated learning can achieve the same performance as centralized learning while preserving privacy.A cross-experiment comparison reveals that the magnitude of federated learning's benefits depends on dataset characteristics and task complexity. The largest uplift (+6.96%) occurred in the German credit scenario, where moderate sample sizes, complex feature interactions, and uneven data distributions allow federated aggregation to integrate complementary insights. In the nearly homogeneous loan dataset, baseline performance left little room for improvement (+0.38%), yet even here FedAvg captured subtle inter-client variations. The MIT credit rating task yielded a moderate gain (+4.1%), reflecting a balance between heterogeneity and task difficulty.

5.2 Vertical federated learning results discussion

In the vertical federated learning experiments, we tested both neural networks and XGBoost models on three datasets—German credit, Loan data, and MIT credit ranking—and the results showed that both models achieved better performance than single-party training. Vertical federated learning combines data features owned by different institutions and expands the feature dimensions that the model can learn by aligning IDs. This expansion allows the model to capture more comprehensive and multi-faceted feature information, which in turn provides richer evidence and support for subsequent prediction tasks, enabling the model to make more informed judgments.

As can be seen from the experiment on Loan data, even if the dataset itself is optimized well enough that the model trained by a single party can already achieve a fairly high accuracy, vertical federated learning can still further improve performance and pursue a more precise result. This indicates that vertical federated learning is not only effective when single-party data is scarce or of low quality, but also has the potential to tap into hidden information in multi-party features even on high-quality datasets, pushing the model's performance to a new level.

From the experiment on MIT credit ranking, it is evident that models trained by a single party struggle to achieve ideal results when faced with multi-classification tasks. This is because multi-classification tasks are more complex than binary classification tasks, and the lack of sample feature dimensions at this time makes the model lack sufficient information to make predictions. In such scenarios, the role of vertical federated learning is significantly manifested: in multi-classification tasks, whether in terms of accuracy, F1-score, or AUC, the improvement brought by vertical federated learning is the most significant. The concatenation of feature dimensions allows the model to learn in a higher-dimensional feature space, where it can better distinguish the subtle differences between multiple categories.

The experimental findings indicate that the performance of the VFL model is very close to that of the model trained on the full-feature dataset (Centralized model). This is as expected, given that VFL effectively combines feature sets from various parties for training. However, the advantage of federated learning lies in its privacy protection mechanisms. Throughout the entire process, data transmission occurs in an encrypted state, ensuring that each organization need not worry about the potential leakage of its data. This ensures both data security and high-performance model training.

In addition, we compared the impact of sample balance in the dataset on vertical federated learning. It can be seen that even when the samples are imbalanced, vertical federated learning can still bring certain performance improvements. This verifies the robustness of vertical federated learning in coping with different data distribution characteristics, as it does not rely on strict sample balance assumptions and can adapt to the actual data scenarios where class imbalance is common in real-world applications.

5.3 Comparison between HFL and VFL

From the experimental results, both horizontal federated learning and vertical federated learning can improve model accuracy. As important methods within the framework of federated learning, they play distinct roles in different application scenarios.

Horizontal Federated Learning addresses the scenario where various institutions possess data with the same feature dimensions but distinct sample IDs. In essence, it functions as a data augmentation strategy that enriches the dataset by enabling encrypted transmission and training of samples from multiple parties. By aggregating scattered samples across institutions while preserving data privacy through cryptographic techniques, it effectively expands the volume of training data, allowing the model to learn more generalized patterns that are robust to variations in the data distribution.

Vertical Federated Learning, on the other hand, tackles the problem where institutions hold samples with the same IDs but different feature dimensions. With the aid of encryption algorithms, it expands the feature space by aligning sample IDs and concatenating complementary features. This approach enables the model to fully leverage the diverse feature information owned by each party, breaking through the limitations of single-party feature scarcity and capturing more comprehensive relationships between features and target variables, thereby enhancing the model's predictive power.

In terms of adaptability to application scenarios, horizontal federated learning is more suitable for collaboration within the same industry. For example, when multiple banks share similar financial features but serve different customer groups, aggregating sample sizes can enhance the model's generalization ability across various customer types. In contrast, vertical federated learning has greater advantages in cross-industry cooperation. A case in point is the joint modeling between banks and e-commerce platforms, where complementary features are leveraged to explore the in-depth patterns of user credit.

The core difference between the two lies in their approaches to addressing "data scarcity": HFL focuses on expanding the number of samples, reducing statistical biases by increasing data scale; VFL, on the other hand, emphasizes extending feature dimensions, breaking through the cognitive limitations of a single perspective by integrating heterogeneous information. This differentiation allows them to form a complementary relationship in the federated learning ecosystem, jointly promoting the implementation of privacy-preserving distributed AI technologies in diverse scenarios.

6 Conclusion

In conclusion, this project highlights the significant potential of both Horizontal Federated Learning (HFL) and Vertical Federated Learning (VFL) in improving credit risk prediction models while ensuring data privacy across multiple institutions. By leveraging federated learning techniques, we successfully addressed the challenges of data silos and privacy concerns in the financial industry, enabling collaborative model training without the need for direct data sharing, while achieving accuracy levels comparable to centralized learning. The experimental results clearly demonstrate that federated learning enhances model generalization, accuracy, and robustness by aggregating model updates from multiple clients, particularly in scenarios involving heterogeneous data from different clients. Horizontal Federated Learning proves to be especially effective when different institutions have similar feature spaces but distinct user groups. By aggregating data from multiple sources, HFL helps enhance the model's ability to generalize across various customer types. In contrast, Vertical Federated Learning shows its strength in cross-industry collaboration, where institutions possess complementary features that, when combined, can significantly improve the model's predictive performance.

Furthermore, the results from both neural networks and XGBoost-based models further validate the value of federated learning in addressing the complexities of credit risk assessment. Even when single-party models perform well, federated learning still provides performance boosts by combining diverse feature sets, improving classification accuracy, F1-scores, and AUC values. This finding underscores the importance of multi-party collaboration, where each institution contributes unique insights that help refine the overall model. Looking ahead, future research could explore adaptive aggregation strategies, such as FedProx, to tackle challenges posed by heterogeneous data, as well as investigate enhanced privacy-preserving techniques to safeguard sensitive information during federated training. Additionally, further optimization of federated learning systems' scalability and real-time performance could ensure their applicability in large-scale, high-risk applications like financial credit risk assessments.

Ultimately, federated learning represents a promising solution for privacy-preserving machine learning, not only enhancing model performance but also fostering collaboration between institutions that would otherwise be unable to share data due to privacy concerns. With increasing regulatory attention on data privacy and security, federated learning provides an effective path for industries such as finance, healthcare, and beyond to develop more robust, fair, and privacy-conscious AI models. As the technology matures, federated learning is expected to play a crucial role in addressing the ethical and practical challenges of data collaboration in the age of big data and artificial intelligence.

Appendices

Horizontal Federated Learning Pseudocode

```
// ================================
// STANDALONE LEARNING
// ================================
FUNCTION standalone_learning()
    FOR each client IN [1,2,3] DO
        data ← load("german_credit_part{client}.csv")
        train, test ← split(data, 0.8)
        model ← MLPClassifier(100,50)

        model.fit(train, epochs=21)
        accuracy ← model.evaluate(test)

        PRINT "Client", client, ":", accuracy
    END
    RETURN avg_accuracy    // 74.13%
END

// ================================
// HORIZONTAL FEDERATED LEARNING
// ================================
FUNCTION federated_learning()
    global_model ← MLPClassifier(100,50)
    clients ← initialize_clients([part1, part2, part3])

    FOR round = 1 TO 21 DO
        // 1. Distribute global parameters
        global_params ← get_params(global_model)
```

```
        // 2. Local training
        updates ← []
        weights ← []
        FOR each client DO
            set_params(client.model, global_params)
            client.model.fit(client.data, iter=1)
            updates.append(get_params(client.model))
            weights.append(len(client.data))
        END

        // 3. Federated averaging
        new_params ← weighted_average(updates, weights)
        set_params(global_model, new_params)

        // 4. Evaluate
        IF round % 3 == 0 THEN
            accuracy ← evaluate_global(global_model, clients)
        END
    END

    RETURN final_accuracy    // 81.09%
END
```

Vertical Federated Learning Pseudocode

```
// ================================
// STANDALONE LEARNING
// ================================
# Preprocessed dataset with 'id', feature columns, 'label'
dataset = preprocessed_data

# Split features into three non-overlapping subsets
features = [col for col in dataset.columns if col not in ['id', 'label']]
c1_feat, c2_feat, c3_feat = split(features, 3)    # Split into 3 subsets
c1_data = dataset[['id'] + c1_feat + ['label']]
c2_data = dataset[['id'] + c2_feat + ['label']]
c3_data = dataset[['id'] + c3_feat + ['label']]

# Local training and evaluation
def local_train_eval(data):
    X, y = data.drop(['id','label'], axis=1), data['label']
    X_train, X_test, y_train, y_test = split(X, y, test_size=0.2)
    model = init_model()    # Initialize model (e.g., XGBoost/NN)
    model.fit(X_train, y_train)
    return metrics(model, X_test, y_test)    # Returns accuracy, F1, AUC

# Train and evaluate each client locally
c1_metrics = local_train_eval(c1_data)
c2_metrics = local_train_eval(c2_data)
c3_metrics = local_train_eval(c3_data)
// ================================
// VERTICAL FEDERATED LEARNING
// ================================
```

```
# Vertical Federated Learning (simulated feature concatenation)
fed_data = merge(c1_data, c2_data, c3_data, on=['id','label'])
X_fed, y_fed = fed_data.drop(['id','label'], axis=1), fed_data['label']
X_train, X_test, y_train, y_test = split(X_fed, y_fed, test_size=0.2)
fed_model = init_model()
fed_model.fit(X_train, y_train)
fed_metrics = metrics(fed_model, X_test, y_test)
```

References

[1] Zheng, Fanglan, et al. "A vertical federated learning method for interpretable scorecard and its application in creditscoring." arXiv preprint arXiv:2009.06218 (2020).

[2] Zheng, Fanglan, et al. "A federated interpretable scorecard and its application in credit scoring." InternationalJournal of Financial Engineering 8.03 (2021): 2142009.

[3] Rida, Abdollah. "Machine and deep learning for credit scoring: A compliant approach." *arXiv preprint arXiv:2412.20225* (2024).

[4] Zhang, Shuyao, Jordan Tay, and Pedro Baiz. "The effects of data imbalance under a federated learning approach for credit risk forecasting." *arXiv preprint arXiv:2401.07234* (2024).

[5] Jiang, Xue, Xuebing Zhou, and Jens Grossklags. "Comprehensive analysis of privacy leakage invertical federated learning during prediction." Proceedings on privacy enhancing technologies (2022).

[6] Hu, Yuzheng, et al. "Is vertical logistic regression privacy-preserving? a comprehensive privacyanalysis and beyond." arXiv preprint arXiv:2207.09087 (2022).

[7] McMahan, Brendan, et al. "Communication-efficient learning of deep networks from decentralized data." *Artificial intelligence and statistics*. PMLR, 2017.

[8] Yang, Qiang, et al. "Federated machine learning: Concept and applications." *ACM Transactions on Intelligent Systems and Technology (TIST)* 10.2 (2019): 1-19.

Contribution

Liu Dahai is responsible for implementing Horizontal Federated Learning (HFL) with the German Credit dataset and Loan data. His work includes setting up the federated learning environment, aggregating client models, and evaluating the performance improvements achieved through horizontal federated learning.

Huang Ruifan is responsible for implementing Horizontal Federated Learning (HFL) with the MIT Credit Ranking dataset. His work focuses on optimizing the federated learning environment to improve model performance, particularly in handling diverse features and data distributions across multiple clients.

He Yanze is responsible for Vertical Federated Learning (VFL) with Neural Networks (NN). His work involves implementing VFL to handle data vertically partitioned across different parties, using neural networks to improve model performance while ensuring data privacy and security.

He Fuzhi focuses on Vertical Federated Learning (VFL) with XGBoost. He is responsible for applying XGBoost to vertical federated learning tasks, fine-tuning the model for optimal performance, and ensuring the privacy and security of data across different clients.