An Explainable Federated Learning and Blockchain based Secure Credit Modeling Method

Fan Yang[a] (410944749@qq.com)

Mohammad Zoynul Abedin[b] (m.z.abedin@swansea.ac.uk)

Petr Hajek[c] (petr.hajek@upce.cz)

[a] School of Computer Science and Technology, Xi'an Jiaotong University, No.28, Xianning West Road, Xi'an, Shaanxi, 710049, P.R. China

[b] Department of Accounting and Finance, School of Management, Swansea University, Bay Campus, Fabian Way, Swansea SA1 8EN, Wales, the United Kingdom

[c] Faculty of Economics and Administration, University of Pardubice, Studentska 95, Pardubice, 53210, Czech Republic

Corresponding Author:

Mohammad Zoynul Abedin

Department of Accounting and Finance, School of Management, Swansea University, Bay Campus, Fabian Way, Swansea SA1 8EN, Wales, the United Kingdom

Tel: (+44) 7459248600

Email: m.z.abedin@swansea.ac.uk

Funding: This research is supported by the Natural Science Basic Research Program of Shaanxi [Program No.2023-JC-YB-490]. This research is also supported by the Czech Sciences Foundation [grant number 22-22586S]; and the COST Action CA19130.

An Explainable Federated Learning and Blockchain based Secure Credit Modeling Method

Abstract

Federated learning has drawn a lot of interest as a powerful technological solution to the "credit data silo" problem. The interpretability of federated learning is a crucial issue due to the lack of user interaction and the complexity of credit data monitoring. We advocate the importance of a credit data processing-as-a-service model, which completes conventional credit models in local environments, in order to overcome these restrictions. In particular, we describe an explainable federated learning and blockchain-based credit scoring system (EFCS) in this work. First, we propose an explainable federated learning method with controllable machine learning efficiency and controllable credit model decision making, thus having controllable credit model complexity and transparent and traceable credit decision-making mechanism. Then, we suggest an explainable federated learning training mechanism for credit data that prevents leakage of the model gradients trained by individual nodes during the training of the overall model. Neither the credit data provider nor the data user has access to the raw data in the credit model training ecosystem. Therefore, privacy protection, model performance, and algorithm efficiency, the core triangular cornerstones of federated learning, when added with model interpretability, together constitute a more secure and trustworthy federated learning-based methodology, thus providing a more reliable service for credit model training and construction. The EFCS scheme is presented via simulations of different types of federated learning and their resistance to system attack, applying the proposed model to six different credit scoring datasets. Extensive experimental analyses support the efficiency, security, and explainability of the EFCS.

Keywords: Analytics; Explainable federated learning; Privacy-preserving; Information leakage; Byzantine fault-tolerant

1. Introduction

Blockchain is a relatively new trend and fundamental technology in the field of digital economy, in part due to the advent of Bitcoin and its capacity to establish a trust environment that allows economic activity in the face of information asymmetry and identity uncertainty. Users are becoming more concerned about whether their private information is being used, or even misused, for commercial or political objectives without their consent as a result of the expanding privacy and data security concerns associated to the usage of Big Data and related algorithms. Many Internet corporations have recently paid hefty fines for revealing user privacy to for-profit businesses. Additionally, governments are aggressively promoting legislation to safeguard user privacy. Therefore, secure training of credit models is of paramount importance (Cheng et al., 2021; Imteaj and Amini, 2022). Recent research on credit scoring models has drawn attention to the importance of explainability (Dumitrescu et al., 2022; Gunnarsson et al., 2021). On the one hand, explainability is essential for safety-critical artificial intelligence-based algorithms that aim to extend some widely available capabilities toward fully automated credit scoring. On the other hand, explainability is required during the design stage to perform model debugging and knowledge discovery, which improves system security by lowering model vulnerabilities against external threats.

In general, the credit lending services made available to customers by the big data sectors are a major factor in the financial sector's explosive expansion. Credit data and models must be secured during the sharing process because they contain a lot of private information (Dastile et al., 2022). Users' credit scores determine whether they can get loans and how much they can borrow. Through the use of statistical models to convert pertinent data into numerical measures that inform credit decisions, credit scores can help lenders analyze the potential risk of new customers as well as the future behavior of existing customers. Traditional credit scores are based on a person's financial history and reflect the credit risk associated with a borrower (Medina-Olivares et al., 2022; Yfanti et al., 2023; Štěpánková, 2021). Therefore, while we consider credit models to be efficient, we also need to focus on the security and interpretability of the models in order to meet the current urgent needs for credit system study.

As a result, consumers are becoming more and more concerned about whether their personal information is being utilized without their consent for commercial or political goals. Several Internet companies have recently received hefty fines for disclosing customer privacy to for-profit entities. Data silos also make it difficult to access the large amounts of data needed for artificial intelligence (Kriebel and Stitz, 2022). To address the shortage of data in small and medium-sized financial institutions and to lessen the information asymmetry between lenders and borrowers, one idea is to use blockchain technology to create a credit data sharing alliance (Zhang et al., 2020).

Research on credit model sharing and techniques for credit data privacy assurance faces a number of difficulties: 1) Local model training techniques are the main focus of today's research on credit model training processes. However, due to the growing number of credit data providers, it is challenging for such locally trained models to fully tap into the credit data of various data providers on the one hand, and it is impossible to share model updates on the other hand, significantly decreasing the effectiveness of credit scoring; 2) The issue of data silos among credit data providers makes it difficult for credit data to be shared and interoperable. 3) Research on credit model sharing mechanisms is further hampered by the difficulty of safely storing and using credit data due to the lack of an appropriate privacy protection method. 4) Current federated learning methodologies (Cheng et al., 2021; Imteaj and Amini, 2022) are not interpretable enough to offer reliable credit assessment services.

The following has to be addressed in the current study on credit model construction and credit data privacy protection. First, it is difficult to build effective credit models since credit data suppliers are now reluctant to provide their own data and are unable to sufficiently protect customers' privacy. Second, there are insufficient financial incentives for data providers to share their data, which discourages businesses or organizations from making their data available for use by others. Third, current research on dispersed training credit models is difficult to comprehend and does not adequately support the development of retrospective credit models. To bridge this gap, we here propose an explainable blockchain-based federated learning architecture for credit scoring. To the best of our knowledge, this study proposes for the first time a credit scoring system using explainable federated learning.

Therefore, we propose a credit modeling approach that makes use of federated learning and credit data privacy assurance mechanisms in order to address the difficulties of credit model sharing and credit data privacy assurance research. This approach solves the problems of privacy risk, inefficiency of credit models, and poor utilization of credit data from different nodes that exist in conventional credit scoring systems. The following are significant contributions of our work:

- We propose a new methodology for secure credit data sharing. Due to the current release of data privacy regulations, access to data needs to be compliant and legal, no more uncontrolled access to user data as in the past. We therefore design a blockchain-based credit data sharing method EFCS, where data are stored locally, while data features and indexes are uploaded to the chain, and federated learning is used to train the credit

model on the data, which ultimately results in a model rather than real credit data and therefore protects user privacy.

- We offer a fair incentive mechanism to motivate data providers to actively participate in the construction of the credit data sharing platform, thus solving the current problem of data silos in credit data sharing and greatly facilitating the interoperability of credit data.
- We also provide an explainable federated learning mechanism for credit model construction and propose the decentralized Byzantine fault-tolerant stochastic gradient descent algorithm D-SGD, which can improve the reliability of credit model training process.
- In addition, we suggest the credit qualification proof (CQP) and credit score evaluation proof (CSEP) methods to verify the authenticity of credit score results. Furthermore, we propose the use of automated neural architecture search methods for credit model construction. Extensive experimental results have evidenced the EFCS system to be highly efficient, safe and reliable, and outperforms competitive state-of-the-art methods.

The next sections of the study are structured as follows. In Section 2, pertinent studies on blockchain and federated learning are reviewed. The blockchain idea and important federated learning methodologies are outlined in Section 3. The proposed credit model sharing method based on blockchain and federated learning is presented in Section 4. The main EFCS methods and algorithms are presented in Section 5. Additionally, based on zero-knowledge proofs, the privacy assurance mechanism is proposed for credit scoring data. A thorough security analysis of the EFCS system is provided in Section 6. In Section 7, we compare and analyse the security and effectiveness of our proposed credit model sharing and credit data privacy assurance mechanisms through extensive experiments. In Section 8, the core contributions of the paper are summarized and the implications of the research are presented, together with an outlook on future research directions.

2. Related Work

2.1. Blockchain technology

Blockchain technology allows any data and digital assets to be put into a blockchain, which uses a series of immutable records with timestamps to hold information, managed by a cluster of computers. It is the decentralized and untamperable nature of blockchain that makes it an essential technology in secure data storage systems (Gai et al., 2020; Wang and Su, 2020; Bodkhe et al., 2020; Berdik et al., 2021; Dai et al., 2019; Yang et al., 2021b, 2022c). For three-echelon supply chain with upstream sellers offering trade credits to downstream buyers that are cash constrained,Wang et al. (2023a) concentrated on the implementation of blockchain technology. To understand participants' motivations and conditions for adopting BCT as well as the various roles of participants in the adoption of blockchain technology, they examine the game equilibria of the no, upper-stream, lower-stream, and complete BCT-driven accounts receivable chains. Amini et al. (2022) created a decentralized clearing method that automatically and endogenously delivers a claims resolution process. This approach can be used to leverage blockchain to clear a network of commitments. Additionally, they offer an algorithm that builds the blockchain, ensuring that payments can be validated and that miners are paid. In order to achieve data security, traceability, and immutability, Li et al. (2023) introduced Fabric-SCF, a Blockchain-based secure storage system that designs and implements these features via distributed consensus. In addition to using smart contracts to specify system operations and access policies, the attribute-based access control model is implemented for access control in order to assure the system's effective operation. Latif et al. (2021) presented a blockchain-based architecture for industrial processes that is both safe and

trustworthy. To control access to vital sensor and actuator data, a secure and lightweight blockchain architecture was suggested. For cross-domain IIoT, Shen et al. (2020) introduced BASA, a block-chain-assisted secure device authentication technique. The consortium blockchain was used to build trust between diverse domains. Lin et al. (2020) suggested a new concept of Decentralized Conditional Anonymous Payment (DCAP) as well as the security standards that go along with it.

Overall, blockchain technology is a fundamental technology with promising applications in banking. Blockchains have the potential to disrupt the core technology of banking credit information systems, thereby modernising and reshaping them (Guo and Liang, 2016). Despite the fact that blockchains are self-governing and permissionless, actual deployment of a decentralized system providing explanations for their decisions is a problem that need to be solved.

2.2. Credit scoring methods

Over the last ten years, the focus of credit scoring methods has shifted from single classification models to ensemble-based models (Gunnarsson et al., 2021). In the area of credit scoring, Xia et al. (2018) proposed a novel heterogeneous ensemble credit model that combines the bagging and stacking algorithms. The proposed system differs from existing ensemble credit models in three ways: pool generation, base learner selection, and trainable fuser. Pławiak et al. (2019) proposed a novel technique based on deep genetic cascade ensembles of multiple support vector machine (SVM) classifiers (DGCEC). The method included a novel 16-layer genetic cascade ensemble of classifiers with two types of SVM classifiers, normalization approaches, feature extraction methods, three types of kernel functions, parameter optimizations, and stratified 10-fold cross-validation. Penalized logistic tree regression (PLTR), which Dumitrescu et al. (2022) devised, is a high-performance credit scoring method that makes use of data from decision trees to enhance the performance of logistic regression for credit scoring. Two tree-based enhanced GBDTs which combines AugBoost-RFS and AugBoost-RFU were proposed by Liu et al. (2022). for credit scoring. The augmentation of features for credit scoring has been rendered easier by tree-based embedding systems. Tripathi et al. (2019) proposed a credit scoring model that is a combination of the two. The first phase is preprocessing, which assigns ranks and weights to classifiers. The ensemble feature selection approach is then used to the preprocessed dataset in the following phase. Finally, the dataset with the selected features is used in a multilayer ensemble classifier architecture in the final phase. Zhang et al. (2019) proposed a novel multi-stage hybrid model that combines feature selection and classifier selection to achieve the best feature subset and best classifier subset, and then uses classifier ensemble to improve prediction performance based on the two best subsets.

Credit scoring relies mainly on payment history, transaction records, professional history, and the like, which are obtained from the various credit databases. Despite the interest in credit scoring, surprisingly few studies have addressed its security and the preservation of borrowers' privacy. To overcome these issues, blockchain has the potential to deliver a decentralised credit scoring solution (Hassija et al., 2020), while federated learning can be used as a privacy-preserving machine learning framework in which multiple parties are permitted to train a single credit scoring model without sharing their customer data.

2.3. XAI and Automated machine learning approaches

The XAI study has so far shown various objectives to derive from the development of an explainable model. In terms of the objectives necessary to specify what an explainable model should compel, very few of the studies we studied totally concur. All these various objectives, however, might make it easier to determine the reason why a particular machine learning explainability exercise is carried out. Unfortunately, only a small number of contributions

have attempted to conceptually define such goals (Arrieta et al., 2020). The definitions for these XAI goals are now summarized and listed in Table 1.

Empirical studies on real credit data have shown that, while maintaining the predictive power of machine learning, a comparable degree of interpretability to scorecards can be achieved (Bücker et al., 2022). Moreover, the AutoML tools, such as H2O or MLJAR, provide the appropriate means to achieve these objectives of credit scoring automatically (Bücker et al., 2022). Another avenue of recent research is the combination of blockchain used to ensure the tracability and security of the scoring system and AutoML-based random forest model performing feature engineering and credit model construction (Yang et al., 2022b). However, it is desirable that such scoring systems meet all these requirements for accuracy, explainability, and security. To this end, it is appropriate to take advantage of automated machine learning, XAI, and blockchain technology when developing credit scoring systems.

Table 1: Goals pursued toward reaching explainability and their main target audience

XAI Goal	Main target audience
Trustworthiness	Domain experts, users of the model affected by decisions
Causality	regulatory entities/agencies
	Domansferability experts, data scientists
Confidence	Domain experts, developers, managers
Fairness	Users affected by model decisions, regulatory entities/agencies
Accessibility	Product owners, managers, users affected by model decisions
Interactivity	Domain experts, users affected by model decisions
Privacy awareness	Users affected by model decisions, regulatory entities/agencies

2.4. Federated learning

Federated learning opens up new avenues for artificial intelligence research. Federated learning is a revolutionary training strategy for developing tailored models that does not compromise user privacy (Lim et al., 2020; Khan et al., 2020; Kim et al., 2019; Yang et al., 2022a). The goal of federated learning is to carry out efficient machine learning between multiple participants or computing nodes while safeguarding the security of information in the exchange of big data, protecting the privacy of endpoint data and personal data, and ensuring legal compliance (Zhang et al., 2021; AbdulRahman et al., 2021; Mothukuri et al., 2021).

Regarding financial applications, Imteaj and Amini (2022) proposed a federated learning-based model to predict borrowers' financial distress by building a global machine learning model that evolves from distributed agents' local models. The model attained nearly identical prediction accuracy as the centralized model. Notably, Cheng et al. (2021) performed a thorough analysis of SecureBoost, a federated learning boosting model, theoretically proving that the model performs as accurately as non-federated boosting models.

Accordingly, existing research has revealed that blockchain and federated learning technologies can effectively address the main challenges of secure data storage and secure transmission of model parameters that arise during the credit model sharing training process. Our research will use the federated learning training mechanism to build a secure and efficient credit model sharing system, thus ensuring that credit data and models are not compromised during the training process. Table.2 summarizes previous efforts to develop explainable, privacy-preserving and secure credit scoring systems, showing the gap in the literature addressed in the current study.

Table 2: Summary of technologies used in existing credit scoring systems

Study	XAI	AutoML	Blockchain	Federated learning
Moscato et al. (2021)	✓			
Yang et al. (2021a)		✓		
Jammalamadaka and Itapu (2022)	✓	✓		
Yang et al. (2022b)		✓	✓	
Bücker et al. (2022)	✓	✓		
Cheng et al. (2021)				✓
Imteaj and Amini (2022)				✓
This study	✓	✓	✓	✓

Therefore, in order to achieve efficient and secure credit model sharing training, a combination of blockchain and federated learning can effectively improve the credit data siloing problem. For multiple nodes with data, blockchain ensures the secure storage of credit data, and federated learning can realize the process of model training jointly by multiple contacts and ensure the data privacy and security during the training process. In the training process of federated learning, only intermediate results of the training process and model parameters are exchanged between nodes to construct a global type model under fused virtual data based on fusion, and no direct transfer of data is made, thus protecting the need of data security and privacy. Neural architecture search technology enables fast search to get efficient and accurate local credit models. The XAI technology combined with federated learning can further enhance the interpretability of credit models, thus improving the efficiency and transparency of joint training of credit models.

3. Preliminary

3.1. Methodology

Blockchain is a technology that originated from Bitcoin, a theory published by Satoshi Nakamoto at a cryptography forum in 2008, and whose founding block was created on January 3, 2009. Bitcoin is essentially a decentralized distributed ledger that is difficult to tamper with and maintains consistency without the need for a central authority due to the incorporation of cryptographic algorithms and distributed consensus algorithms. Blockchain is a kind of decentralized distributed ledger in which data is generated and stored in blocks and linked in chronological order to form a chain structure. A block is a packet of data that carries transaction information on a blockchain network, which is a data structure that is tagged with a timestamp and a hash of the previous block. Blocks are verified and the transactions in the block are validated by the consensus mechanism of the network. The ledger in blockchain records transaction information such as flowing water in a certain format. Especially in various cryptographic digital currencies, the content of transactions is various transfer information.

Consensus algorithms are a core component of blockchain and have been a hot topic of research in the last few years. Consensus algorithms focus on how to ensure the consistency and accuracy of the state and data of each node in a distributed environment. Traditional consensus algorithms consider situations where nodes will not do evil, such as malicious tampering and falsification of data, and only consider errors against network and hardware problems, such as the Paxos and Raft algorithms. Blockchains, because they operate in a non-trusted environment, need to consider the case where nodes can do evil and require consensus algorithms to be Byzantine fault-tolerant in the workplace. As the blockchain evolved, many new consensus protocols with Byzantine fault tolerance were created, inspired by PoW (Proof-of-Work).

Table 3 shows the importance of different technologies for credit scoring models in the proposed methodology. As shown in Table 3, XAI provides transparency and interpretability, allowing users to understand how credit decisions

are made. Federated learning enables collaborative model training on decentralised data, preserving privacy while improving accuracy. AutoML automates the model development process, ensuring that privacy protections are built into credit scoring models. These technologies improve the accuracy, fairness and privacy of credit scoring, benefiting both lenders and borrowers. Fig. 1 illustrates the general design of the EFCS method proposed in this paper.

Table 3: Importance of proposed technologies for credit scoring models

Technology	**Importance for credit scoring models**
XAI	XAI plays an important role in improving knowledge of and confidence in the way machine learning models make decisions. It helps interpret the credit model's predictions and uncover underlying causes and causal relationships, increasing the model's acceptance, interpretability and reliability.
AutoML	AutoML is critical to automating and streamlining the machine learning model building process. It enables non-experts to create accurate and effective credit models, driving the adoption and use of machine learning across a range of industries.
Blockchain	Blockchain is essential for enabling decentralized and secure transactions, immutable record-keeping, and open accountability.
Federated learning	The key value of federated learning resides in its capacity to provide cooperative model training on decentralized credit data while maintaining data privacy and security and producing high-performance models, opening up new opportunities for distributed machine learning in credit scoring models.

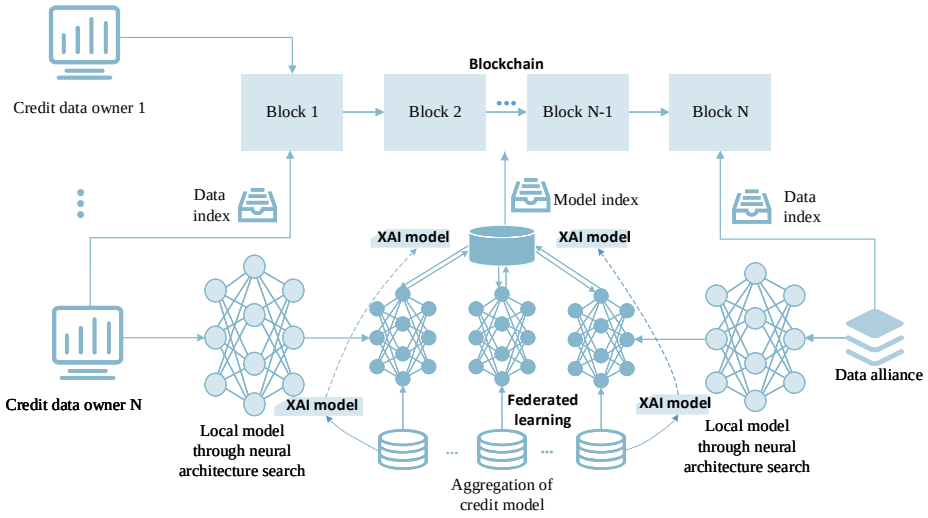


Figure 1: Overview of our approach for Explainable Federated Learning of EFCS

As depicted in Fig. 1, the setup of the EFCS scheme can be divided into several steps, including requesting credit data from N owners, judging the request, indexing the credit data in blockchain storage using the corresponding N blocks, training federated learning models based on multiple XAI models, and obtaining the output credit results as an aggregation of local machine learning models obtained using AutoML architecture search. The coordinator refers to the node selected by consensus mechanism in federated learning, which is responsible for the aggregation of parameter information of different nodes in federated learning and the coordination of different participants in the model training process. The specific process is as follows.

- 1) First, the parameters about the credit evaluation request will be sent from the credit evaluation request node, and in order to prevent a large number of duplicates from affecting the system, it is necessary to use the

Bloom filter method to determine whether the request parameters have been sent before.

- 2) If the request parameter has already been sent and processed before, the cache server can be used to search for the credit evaluation result and output the credit evaluation structure. If it is found that the request parameter has not been sent and processed before, the request can be retrieved on the blockchain.
- 3) If the request is not processed before, but it is shown as processed after retrieval, the request is retrieved again for possible false positives. In the cache server, it is necessary to record the request after re-processing the information retrieval, and set the parameter of the request to 1, indicating that the request has been successfully processed. After receiving the request parameters, the blockchain can output the retrieval result of the request.
- 4) The data providers (credit data owner 1 to N in Fig. 1) are searched by using the Data index. As a result, the N different credit data participants form the participating nodes of the federated learning to train the model together using the Model index. In this training process, neural architecture search techniques (AutoML) facilitate fast search to get local credit models (see the illustrations of the local neural architectures in Fig. 1). The global credit model is trained and aggregated using federated learning, and the coordinator chains the corresponding parameters of the model and provides feedback and updates to the model.
- 5) The credit data requestor can obtain the output of the corresponding request through the global model (aggregation of credit model in Fig. 1). The introduction of XAI into local models also improves the interpretability of the trained models during the federated learning model training process. In the process of credit data request, no direct credit data from different financial institutions will be obtained, but only through the sharing of model training parameters, thus effectively ensuring the privacy and security of credit data, as can be seen from the credit data sharing process in Fig. 2.

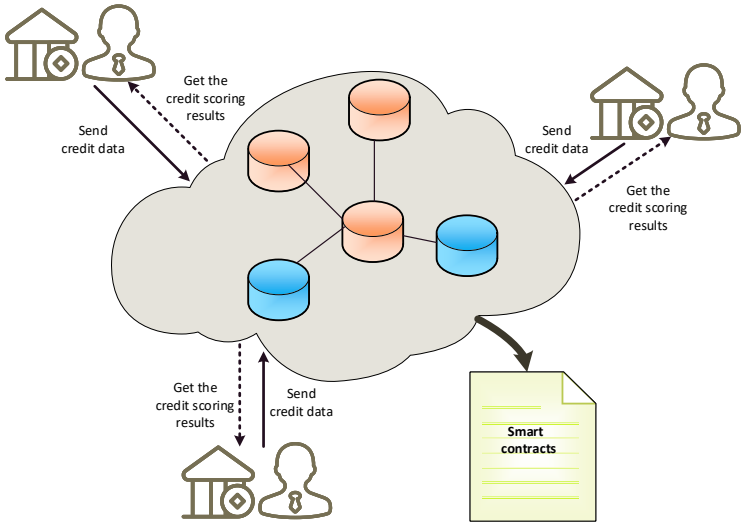


Figure 2: Credit data sharing process on the blockchain

3.2. Cryptography-related technologies

The hashing algorithm, also known as a hash function, maps any set of inputs into a summary of outputs of the same length, called a hash, in a finite amount of time. Its most important features are the following.

- The same hash algorithm acting on the same inputs will definitely result in the same hash value.
- The same hashing algorithm acting on different inputs almost always yields different hash values.
- The original image is irreversible; a hash value can be easily obtained from the input value by the hashing algorithm, but the hash value cannot be worked backwards from the original input.

Therefore, the data sharing scheme designed in our system is based on federated learning to train the model together rather than providing a data interface to the outside world, so that an attacker cannot obtain valid data from the blockchain. The only way to obtain the data is to receive the homomorphically encrypted model parameters from the data provider as a coordinator. If the model parameters are not encrypted and the structure of the original data is known, the real data information can be introduced, but the encrypted data cannot be stolen through regular channels unless the participant actively reveals its key to the coordinator.

3.3. Federated learning and credit model sharing

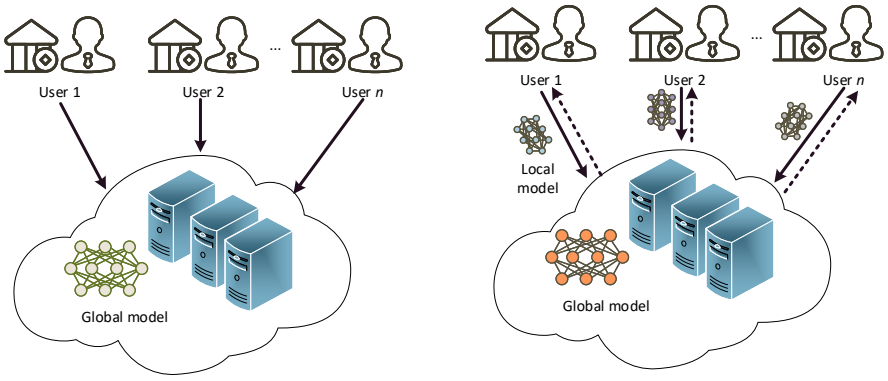


Figure 3: The basic framework for two different training schemes: centralized training and federated training. (a) Centralized Training. (b) Federated Training.

Federated learning is a machine learning model built on a distributed dataset. Federated schools involve two phases, namely model training and model inference. Fig. 3 demonstrates the basic framework for centralized training and federated learning. During model training, the parameters of the model in question can be passed between parties (or in the form of encrypted parameters), but the data must never go out of the local area, often referred to as "the data does not move the model". This process does not reveal any privacy-protected data between the parties. The trained federated learning model can be be placed in each participant's system or shared with all parties.

As illustrated in Fig. 3, credit data providers upload local data straight to the server, putting customers' privacy at risk. As a result, unlike centralized training, federated training necessitates collaboration between each user and server in order to train a single machine learning model. Each consequently improves local parameters (for example, gradients) with the cloud server, which collects all gradients and distributes the results to each user to accelerate model convergence. Finally, the server and each user will have the appropriate network settings. In contrast to centralized training, federated training reduces the likelihood of user privacy being compromised. Attackers can still gain indirect access to sensitive information by leveraging shared gradients. Furthermore, if a bad cloud provider is driven by criminal intent, In addition, if a bad cloud provider is driven by criminal intent, they may deliver the wrong results to consumers. As a result, in this study, we concentrate on protecting users' local gradients while verifying the accuracy of discoveries received by the server during the federated training process.

3.4. Security requirements for credit model sharing

According to the literature, the EFCS credit data sharing system (see Fig. 2) must meet the following basic security standards.

1) Byzantine attacks: The attacker controls multiple users, which are called Byzantine users. Instead of sending locally updated model parameters, Byzantine users can send arbitrary parameters to the central server. This attack can cause the global model to converge at the local optimum, or even cause the model to diverge. Assuming that the Byzantine client has access to the federated learning model, or has white-box access to non-Byzantine client updates, it is difficult to be detected by the system by adjusting the output through normal model updates.

2) Attacks on the credit model training process: During the training process, an attacker can attempt to learn, influence, or compromise the federated learning model. During federated training, the attacker can change the integrity of the training data set collection by means of a data poisoning attack or change the integrity of the learning process by means of a model poisoning attack. The attacker can attack the parameter update process of one participant or all participants.

3) Traceability: Traceability should be offered so that an authorised Manager can track down a rogue user's true identity.

4) Network Analysis Resilience: The EFCS should prevent attackers from tracing profiles that match addresses by evaluating the network history of publicly published transactions.

5) Attacks in the inference phase: If the participants of federated learning want to train a model using each party's data set, but do not want their data set to be leaked to the server, they need to agree on the model algorithm (e.g., neural network) and the mechanism of parameter updates (e.g., stochastic gradient descent (SGD)) for federated modeling. Then before training, the attacker can obtain the mechanism for updating the federation learning parameters and thus specify the corresponding inferred attack strategy. Under ideal conditions, it is generally assumed that there are two participants: one is the attacked party and the other is the attacker.

6) Federated learning security in credit data silos: Credit data sources are often distributed among enterprises and individuals, which are independent and isolated from each other, forming a "data silo" that hinders the development of technology. The inability to connect multiple data sources brings more industry benefits. Currently, it is possible to avoid centralized data storage and create new value from data that is not interoperable from multiple sources. Under the condition of protecting privacy and security, using multiple data sources to drive machine learning optimization and ensuring data security is a security issue that needs to be addressed.

3.5. Secure federated average learning algorithm

Cross-sectional federated learning, which is also known as Sample-Partitioned Federated Learning, can be applied to the case where each participant's dataset has the same characteristics and a different sample space. For example, consider two regional commercial banks in their respective regions. Their business models are similar, so their datasets have similar feature space but with different sample IDs.

If a financial institution uses our suggested Explainable Federated Learning approach and has its own credit dataset D, it can register the encrypted model parameter information it retrieves from its shared dataset on the blockchain, but the original shared data will not be leaked in the process. Government agencies like industry and commerce, tax agencies, public security, customs, and human society bureaus as well as corporate data providers like e-commerce and social networking platforms that index and record on the blockchain the financial data, legal records, criminal histories, social security payments, and other details of people or businesses can make up the data federation. Traditional regulation typically detects non-compliance after the event and is ex post facto. Regulators can proactively control the behavior of financial institutions in real time because all information flows are logged on the chain in real time under this system. Instead of using direct data transfer, credit data sharing uses shared models. Any request is delivered to the system, which will look for compatible data providers for output models from joint modeling and index the models on the chain in response to demand.

In Algorithm 1, it can be seen that the whole process of the homomorphic confidential-based federated averaging algorithm, which differs from the normal federated averaging algorithm in that the coordinator does not have access to the plaintext parameters from the participants. If the data structure is leaked then the leakage of the model gradient and model parameters may lead to the privacy of important data information and model information, therefore, the use of homomorphic encrypted model parameters of federated learning algorithms are referred to as secure federated averaging algorithms.

3.6. Proxy proof-of-stake consensus mechanism SDPOS

The data providers develop models that have the potential to earn money by taking part in federated learning and so contribute to the system as a whole. How to measure the contributions that members make to the system and

Algorithm 1 Homomorphic encryption-based model-averaged federated learning mechanism

Input: Global credit model parameters

Initialize the global credit model parameters w_0 and broadcast them to all participating nodes p_i for model training

for each global model update round $k = 1, 2, 3, ...$ **do**

  for each participating node $p = 1, 2, 3, ...$ **do**

    Update local credit model parameters $\left[\left[W_{k+1}^{(i)}\right]\right] \leftarrow \left[\left[\overline{W}_k\right]\right]$

    Send the updated model parameters $\left[\left[w_{k+1}^{(i)}\right]\right]$ and loss function $L_{k+1}^{(i)}$ to the coordinator

  end for

  Coordinators perform a weighted average of the encrypted model parameters $[[\bar{w}_{k+1}]] \leftarrow \sum_{j=1}^{n} \frac{1}{n} \left[\left[w_k^{(i)}\right]\right]$

  The coordinator tests the loss function to determine whether it converges or not. If it converges, a termination signal is sent to the participating nodes of the model.

  The coordinator sends the aggregated model parameters $\left[\left[w_{k+1}^{(i)}\right]\right]$ to all participating nodes.

end for

Participating nodes perform local model updates $(i,\ [[\bar{w}_k]])$

The participant decrypts $[[\bar{w}_k]]$ to obtain $\overline{W}_K$.

for Each iteration from 1 to local iteration number k **do**

  Randomly divide the credit dataset D_i into B batches

  Set the model parameters from the previous iteration as the initial parameters for the current iteration $w_{1,t}^{(i)} = w_{B,t-1}^{(i)}$.

  for Each iteration from 1 to local iteration number B **do**

    Calculate gradient values $g_i^{(b)}$.

    Update local model parameters $w_{b+1,t}^{i)} \leftarrow w_{b,t}^{(i)} - \eta g_i^{(b)}$.

  end for

end for

Set updated local model parameters as $\overline{W}_{k+1}^{(i)} = W_{B,K}^{(i)}$.

The participating nodes locally homomorphic encrypt $\overline{w}_{k+1}^{(i)}$ to obtain $\left[\left[\bar{w}_{k+1}^{(i)}\right]\right]$ and send it to the coordinating node together with the loss function.

ensure its sustainability is the main challenge. In this study, we combine the Shapley value with DPOS (Delegated Proof of Stake) to create a mechanism that satisfies both the consensus of the blockchain and the interests of all parties.

Algorithm 2 Proxy proof-of-stake consensus mechanism SDPOS

Input: Shared credit data for nodes involved in federated learning.

while In a consensus cycle **do**
  All participants vote according to their contribution.
  Sort the vote results to get sorted_vote_list.
  Select the N highest voted delegates from the sorted_vote_list.
  delegates ← get N delegates from sorted_vote_list.
  Random disorder delegates ← shuffle(delegates).
  for Pick out the packing node **do**
    Use the hash of the previous block over the interval of the last generated block to get the slot.
    slot ← pre_block_hash / block_interval.
    Use slot to take modulo N to get the representative index.
    index ← slot mod$N$
    if The current node is delegates[index] **then**
      if Current node status is coordinator **then**
        Add the coordinator's contribution of federated learning to your account.
      end if
      Record the contribution of each participant and verify it with each party's public key, and sign the transaction with your own public key.
      generate_block($sign_{\mathrm{sk}}$ (verified_transaction))
    else
      skip
    end if
  end for
end while

The consensus mechanism of Algorithm 2 is mainly through the election of representatives, who have the right to bookkeeping, supervision and the right to act as coordinators. After each round of voting, the N accounts with the highest number of votes become the representatives of the system, responsible for packing the blocks and acting as coordinators of the federated learning and receiving rewards.

When the federated learning is finished, the bookkeeping node, the coordinator, receives a fixed contribution, and the data supplier receives compensation based on the contribution. Each participant casts a vote for the N users they feel are most capable of helping to design and run the system. The system is maintained by these representatives, who also encourage the involvement of the federated learning data suppliers, achieve consensus in a decentralized manner, and make sure the system runs well and with minimal energy loss. The SDPOS consensus protocol technique, which is based on Shapley revenue sharing, involves two elections: one during a consensus cycle and another election of candidate representatives at the conclusion of the consensus cycle.

When playing the Shapley game of profit-sharing, a marginal strategy based on contributions is employed. It offers a fairer and more accurate assessment of the contributions made by the players than the union game profit-sharing technique since it also takes into account the influence of the rewards of members joining the group in different orders. As a result, it adds members to the pool in a different order than other players and figures out their average marginal

gains in the way that is illustrated below.

$$u(i) = \frac{1}{n!} \sum_{R} \left[v\left(P_i^R \cup i\right) - v\left(P_i^R\right) \right] \tag{1}$$

In the mechanism designed in this paper, the contribution of the participants to the reduction in the value of the loss is used as an evaluation function.

$$v(i) = \sum v_{k(i)} = a\left(\frac{L_{k-1}^{(i)} - L_k^{(i)}}{L_{k-1}^{(i)}}\right) \tag{2}$$

The algorithmic steps for calculating the contribution values of the parties in the execution of the federated tie-breaker algorithm are as follows.

- 1) The federated learning coordinator initializes a model and sends the initialized model to each participant.
- 2) N participants train the model locally for several rounds and send it to the coordinator.
- 3) The coordinator will average the models according to the combination of $C_N^2, C_N^3, \ldots, C_N^N$ and send these models to each participant.
- 4) The participants first find the utility according to the corresponding loss value of each model, and then substitute to find the current contribution value.

The consensus mechanism is based on the election of representatives, who have the right to bookkeeping, supervision and the right to act as coordinators. After each round of voting, the N accounts with the highest number of votes become the representatives of the system, responsible for packing the blocks and acting as coordinators of the federated learning and receiving rewards. When the federated learning is complete, the coordinator, the bookkeeping node, is awarded a fixed contribution and the data provider is rewarded according to the contribution. The purpose of this is that each person votes for the N users who are most trusted in the development and operation of the system. These representatives maintain the system and promote the participation of the federated learning data providers, achieving consensus in a decentralised manner and ensuring that the system operates efficiently and with less energy loss. The Shapley revenue sharing-based DPOS consensus protocol algorithm has one election during a consensus cycle and a new election of candidate representatives at the end of the consensus cycle.

4. Proposed EFCS scheme

4.1. Key security characteristics and design principles

The following are the most important security characteristics of EFCS. First, defence against node data theft is ensured. Second, the processes that execute the smart contract, the source data, and the outputs of data analysis must all be safeguarded in EFCS. The created intermediate credit data, in particular, should be safeguarded during the smart contract's execution. Finally, during the whole transmission and execution, no party may steal the providers' credit source data.

We will go over three crucial security components that make up EFCS in more depth. The three primary data kinds in EFCS are source data, intermediate data produced during the execution of smart contracts, and the results of the final data analysis. Furthermore, a bad contract should be avoided. The data supplier may leave a backdoor in the data analysis contract since contracts may invoke one another to send data to another contract. Ethereum Virtual Machine(EVM) encapsulated in the enclave forbids calls between contracts in EFCS to prevent this. Additionally, the malicious buyer might attempt to use the contract to obtain the raw data directly.

EFCS's overarching goal is to address three challenges that emerge throughout the federated training based credit model training process. One is to keep the user's local model gradients private throughout the method. Second, our EFCS assists each user in properly checking the integrity of the server's credit score information, avoiding dangerous server spoofing. Finally, EFCS offers offline help to users during the training period. The particular verification procedure is presented in Appendix 1.

Traditional forms of credit data sharing have been severely restricted, therefore, a new sharing method of federated learning is used in EFCS to solve the credit data sharing problem. Instead of exchanging user data directly between organizations, federated learning is used to model the data they have together, sharing the model instead of sharing the data to protect data privacy and legality requirements.

4.2. Credit scoring results qualification in EFCS system

4.2.1. Credit qualification proof

A blockchain-based credit qualification proof (CQP) (Algorithm 3) and secure credit assessment is proposed, which enables financial institutions to verify users' credit scores and confirm their creditworthiness by providing a small amount of non-sensitive data. This way, users do not have to worry about revealing their privacy when borrowing money, and financial institutions can avoid the risk of non-compliance. The financial institutions can also avoid the risk of non-compliance.

There are two roles in EFCS scheme: the prover and the verifier. The provers need to provide proofs to the verifiers without compromising their privacy, so that the verifiers believe the provers' proofs. Relation for QAP is defined as $R = \left(p, G_1, G_2, G_{\mathrm{T}}, e, g, h, 1, \{u_i(X), v_i(X), w_i(X)\}_{i=0}^{m}, t(X)\right)$, where p is a prime, (G_1, G_2, G_T) is the group of three cycles of order p, e is the bilinear mapping function:$G_1 \times G_2 \rightarrow G_{\mathrm{T}}$ and g and h are generators of G_1 and G_2, $e(g, h)$ is a generator of G_T, and the statements are $(a_1, \ldots, a_1) \in F_p^l$, withwitness is $(a_1, \ldots, a_m) \in F_p^{m-1}$, $t(X)$ is a polynomial of order n. The whole process includes:

- 1) setup process: setup gives the security parameter λ and generates random parameters through an initialisation operation. This includes the following parameters: a random selection from the domain $(\alpha, \beta, \gamma, \delta, x) \leftarrow F_p$.
- 2) generating the proof key and authentication key $(p_k, v_k) \leftarrow \mathrm{genkey}(\alpha, \beta, \gamma, \delta, x)$: First generate $(\sigma, \tau) : \tau = (\alpha, \beta, \gamma, \delta, x), \sigma = \left([\sigma_1]_1, [\sigma_2]_2\right)$, where σ_1 is homomorphically encrypted in G_1 and σ_1 is homomorphically encrypted in G_2. The expression for the authentication key is as follows:

$$\begin{gathered}\sigma_1 = \left(\begin{array}{c}\alpha, \beta, \delta, \left\{x^i\right\}_{i=0}^{n-1}, \left\{\frac{\beta u_i(x)+\alpha v_i(x)+w_i(x)}{\gamma}\right\}_{i=0}^{\ell} \\ \left\{\frac{\beta u_i(x)+\alpha v_i(x)+w_i(x)}{\delta}\right\}_{i=\ell+1}^{m}, \left\{\frac{x^i t(x)}{\delta}\right\}_{i=0}^{n-2}\end{array}\right) \\ \sigma_2 = \left(\beta, \gamma, \delta, \left\{x^i\right\}_{i=0}^{n-1}\right)\end{gathered} \tag{3}$$

- 3) the prover generates a proof: $\pi \leftarrow \mathrm{genproof}\ (p_k, r, s)$ Randomly choose two parameters r and s and compute $\pi = ([A]_1, [C]_1, [\ B]_2)$ with the following specific formulas for A, B and C.

$$\begin{aligned}A &= \alpha + \sum_{i=0}^{m} a_i u_i(x) + r\delta \\ B &= \beta + \sum_{i=0}^{m} a_i v_i(x) + s\delta \\ C &= \frac{\sum_{i=\ell+1}^{m} a_i \left(\beta u_i(x) + \alpha v_i(x) + w_i(x)\right) + h(x)t(x)}{\delta} \\ &\quad + As + rB - rs\delta\end{aligned} \tag{4}$$

- 4) verification process: $\mathrm{out} \leftarrow \mathrm{verify}(v_k, \pi)$. Using this algorithm, everyone can verify the validity of the zero-knowledge proof. If the verification is successful, the algorithm outputs out = 1, otherwise it outputs out = 0. The specific expression for verification is shown below.

$$\begin{aligned}&e\left([A]_1, [B]_2\right) = e\left([\alpha]_1, [\beta]_2\right) \\ &\cdot e\left(\sum_{i=0}^{\ell} a_i \left[\frac{\beta u_i(x) + \alpha v_i(x) + w_i(x)}{\gamma}\right]_1, [\gamma]_2\right) \\ &\cdot e\left([C]_1, [\delta]_2\right)\end{aligned} \tag{5}$$

Algorithm 3 Credit qualification using zero knowledge proof

Input: Credit assessment score results.

1: Take the number of binary digits of score and target as n. As the credit scores are in the range of 0 to 1000, take n=10.
2: Calculation of intermediate variables sym_1 = target $+2^n-$ score.
3: Convert sym_1 to an n+1 bit binary number bits=num_to_bits(sym_1)
4: out=bits[n]
5: $(\sum_{i=0}^{m} a_i u_{ji})\,(\sum_{i=0}^{m} a_i v_{ji}) = \sum_{i=0}^{m} a_i w_{ji}$
6: $\sum_{i=0}^{m} a_i u_i(X) \sum_{i=0}^{m} a_i v_i(X) = \sum_{i=0}^{m} a_i w_i(X) + h(X)t(X)$
7: **return** The credit qualification results

4.2.2. Credit score evaluation proof

The credit score evaluation proof (CSEP) (Algorithm 4) is used by institutions or regulation to verify the authenticity of the user's credit score source. The credit score calculation used in this section is the FICO credit card scoring model.

Algorithm 4 Credit score evaluation proof method

Input: Credit score results after shared Federated model

1: Calculating the user's probability of default odds $= \frac{p}{1-p}, p = \frac{1}{1+e^{-z}}$.
2: Calculating the weight of evidence $WOE_i = \ln\left(\frac{y_i/y_T}{n_i/n_T}\right)$
3: Calculate the score set by the scorecard score $= A - B\log(odds)$.
4: group$(x) = (x \geq s_0) + (x \geq s_1) + \ldots + (x \geq s_{m-1})$
5: score = base score + score $(x_1) + \ldots +$ score (x_n)
6: Convert each gate circuit to an R1CS constraint, then generate QAP constraints by Lagrange interpolation, and finally generate proofs by the zk-SNARK method.
7: **return** The credit score evaluation proof.

The model and data used is the model $W = (w_0, w_1, \ldots, \mathrm{w_n})$ obtained using the federated average algorithm in the previous section. The default probability of a user is defined as:

$$odds = \frac{p}{1-p}, p = \frac{1}{1+e^{-z}} \tag{6}$$

$$\log(odds) = z = w_0 + w_1 \cdot WOE_1 + \cdots + w_n \cdot WOE_n \tag{7}$$

WOE (Weight of Evidence) is a way of coding the original independent variables. First, the variable needs to be grouped (also known as discretization or boxing), and after grouping, the WOE value can be found for group i. The formula is as follows:

$$WOE_i = \ln\left(\frac{y_i/y_T}{n_i/n_T}\right) \tag{8}$$

where y_i denotes the number of defaulting users in group i, y_t is the number of all defaulting users in the sample, n_i is the number of normal users in group i, and n_t is the number of all normal users in the sample. The scorecard is set by defining the score as a linear expression of the logarithm of the ratio.

5. Explainable federated learning method

5.1. Decentralized Byzantine fault-tolerant stochastic gradient descent algorithm

We propose Decentralized Byzantine fault-tolerant stochastic gradient descent algorithm D-SGD for explainable federated learning. Several K participants work together to develop a global model without releasing their personal

training data in this study's cross-device federated learning scenario, also known as a horizontal federated learning scenario. It is necessary to implement specific privacy protection measures due to the hazard model. Certain shady rivals may carry out privacy intrusions on exchanged information to determine the private information of other participants.

The server first grants access to the global model data to all clients. The clients continue to update the local model information while downloading the global model W_a^S . The procedures are repeated till the usefulness of the aggregated model remains unchanged. Remark: In SFL, the local model information consists of the model gradients, model outputs, and model parameters, all of which may be communicated to the aggregator at the user's choosing and exposed to potentially dishonest opponents. A Bayesian privacy leakage metric is used to determine how much private data semi-honest adversaries may still be able to deduce despite the protections put in place for information that has been made publicly available. If the Bayesian privacy leakage is less than an acceptable threshold, the adopted protection mechanism is secure in preventing Bayesian inference assaults and allows one to evaluate the security of a secure federated learning system. To illustrate the use of XAI in the EFCS model (Fig. 1) in more detail, Fig. 4 presents the XAI process on the blockchain in EFCS.

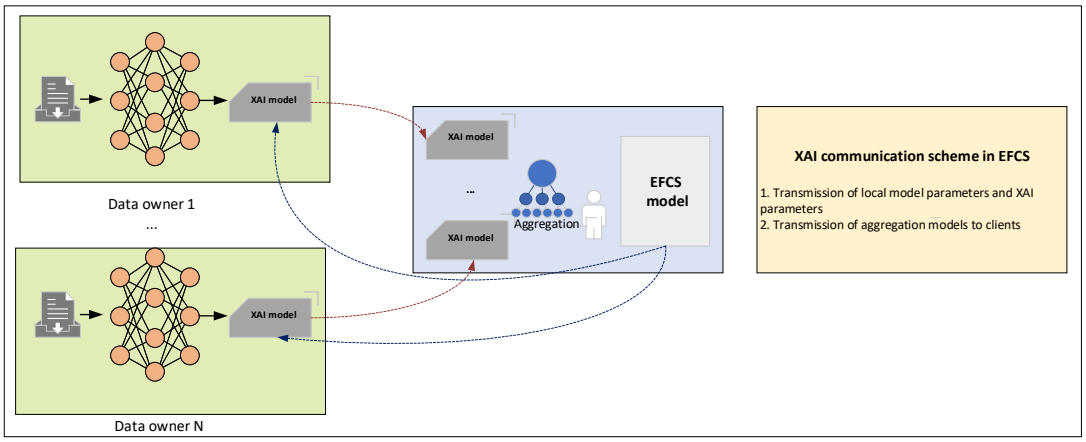


Figure 4: The XAI process on the blockchain in EFCS

We observed a compelling need in previous work to establish a strong foundation for the creation of unique federated learning algorithms. Particularly, existing privacy definitions, such as Differential Privacy, which is regarded as the gold standard definition of privacy, did not clearly reflect the latest Bayesian inference type of threats. It is unclear to practitioners how to choose the best algorithm design in terms of the privacy-utility trade-off due to the lack of precise formulations. There are no known frameworks that consider multiple security measures, such as randomization and homomorphic encryption, at once. Therefore, the remainder of this paper will demonstrate an unique statistical framework built on a Bayesian inference attack that seeks to overcome the aforementioned concerns that have not been well addressed in previous work.

In the XAI method, we use the fuzzy uniform partition with a constrained number of fuzzy sets to create partitions during the XAI communication scheme. The suggested approach ensures the highest level of semantic interpretability. Then, in place of the conventional weighted average approach, we suggest an inference procedure. Instead of producing the global model iteratively, the suggested federated learning technique develops the local fuzzy rules first, which are then given to each client. Each client creates local fuzzy rules initially, which are then submitted to a central server. In the XAI technique, we apply linguistic if-then rules. The general $m-$th rule is described in the following: R_m : IF U_1 is $B_{1,t_{m,1}}$ AND ... AND U_F is $B_{F,t_{m,F}}$, THEN $y_k = \gamma_{m,0} + \sum_{i=1}^{F} \gamma_{m,i} \cdot u_i$, where F represents the overall amount of attributes, B_i,t_m,i denotes the $t-$th fuzzy set of the fuzzy partitionover the $i-$th attribute examined in the $t-$th rule, and m,$i$indicate the value of the coefficient of the linear model, with $i = 0, ..., F$. The level of activation of each rule is calculated using the input pattern $u = [u_1, u_2, ..., u_F]$.

$$h_m(\mathbf{u}) = \prod_{f=1}^{F} \mu_{f,p_{m,f}}(u_f), \mid m = 1, \ldots, M \tag{9}$$

where $\mu_{f,p_{m,f}}(u_f)$ is the level to which u_f belongs to the fuzzy set $B_{f,p_{m,f}}$. The weighted mean of the outputs collected from the M engaged rules is then generated by the inference process.

$$\hat{y}(\mathbf{u}) = y_m(\mathbf{u}) \cdot \sum_{m=1}^{M} \left(\frac{h_m(\mathbf{u})}{\sum_{l=1}^{M} h_l(\mathbf{u})} \right) \tag{10}$$

The server then compiles the rules it has received. Comparing the rules obtained from the clients and resolving any conflicts that may arise when rules from various models, where the antecedents relate to the same or overlapping regions in the attribute space, provide different outputs, are involved in the aggregation routine. The weighted average of the original rule coefficients is used to compute the new results of the aggregated rules, where each rule's weight is based on how well-supported and confident it is.

The truncated median-based strategy is widely used in both centralized and decentralized algorithms. The truncated median-based strategy requires variable updating nodes to receive gradients/variables from other nodes and then dimension-by-dimension. The dimensional scalar of its received quantity is sorted, the largest and the smallest values are removed, and then the remaining values are the average value of the remaining values as the value of the corresponding dimension of the output quantity. Introducing the Byzantine court node, the changed objective function is as follows.

$$F(\mathbf{u}) = \frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \mathbb{E}_{\xi_i \sim D_i} f(\mathbf{u}; \boldsymbol{\xi}_i) \tag{11}$$

For any non-Byzantine node $i \in \mathcal{H}$ in the decentralized system, in the t-th iteration round, a small batch of data is taken from the node's own in the t-th iteration round, small batches of data are taken from the node's own dataset D_i for training. The size of the small batches sampled by each non-Byzantine node in each round may be different for each non-Byzantine node, but without loss of generality, the default small batch size sampled is $\mathcal{S}$ and the objective function is thus given as follows.

$$\tilde{f}_i(\mathbf{u}) = \sum_{l \in \mathcal{S}_i(t)} f(\mathbf{u}, \xi_l) \tag{12}$$

5.2. Credit Economical Neural Architecture Search (CE-NAS)

Recall that the local models in the EFCS model (Fig. 1) rely on neural architecture search (NAS). NAS (Elsken et al., 2019) is a technique for automatically designing neural networks, allowing algorithms to automatically design high-performance network structures based on sample sets, rivaling even the level of human experts in some tasks, and even discovering certain network structures not previously proposed by humans, which can effectively reduce the cost of using and implementing neural networks. Fig. 5 represents the three key steps of credit economical NAS (CE-NAS), which are credit model searching space, searching strategy, and credit model estimation.

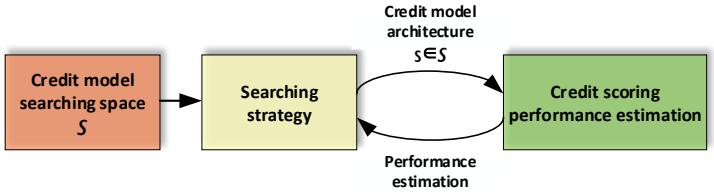


Figure 5: Credit economical neural architecture search

Because applying neural architecture search directly to credit scoring models wastes a lot of time and resources, we suggest a credit economical neural architecture search CE-NAS to reduce needless time spent on credit scoring model creation. In a given architecture search area, CE-NAS aims to locate the best training hyper-parameters. In this research, we implement the commonly used cell-based architecture search space. A network is made up of a

certain number of cells that can be norm or reduction cells. In each cell, the outputs of the preceding two cells are initialized. A cell is an M-node directed acyclic graph that is fully connected (DAG) $\{N_1, N_2, \ldots, N_M\}$. Each node N_i receives the dependent nodes as input and produces an output via a sum operation $N_j = \sum_{i<j} o^{(i,j)}(N_i)$. Each node represents a different tensor, and each directed edge (i, j) between N_i and N_j describes an operation $o^{(i,j)}$ that is drawn from the corresponding operation search space $\mathcal{O}(i, j)$.

We offer the minimum credit importance pruning strategy. We start by sampling the hyper-parameter examples that take the least amount of time. The hyper-parameter importance is then estimated using random forest utilizing the sampled cases. The least important hyper-parameter is then trimmed by setting the value with the lowest time cost. The trimming stage is completed when the search space contains only one hyperparameter. We introduce a category distribution related to the computational cost for every element in Θ_i:

$$c(\zeta_{i,j}) = \frac{\exp\{-F(\Theta_{i,j})\}}{\sum_j \exp\{-F(\Theta_{i,j})\}} \tag{13}$$

where the number of floating point operations is represented by the function $F(\Theta_{i,j})$. We generate a set D with different subsets $\boldsymbol{x}^{\text{ref}}$, $\boldsymbol{x}^{\text{pos}}$, and $\boldsymbol{x}^{\text{neg}}$ after repeating the prior operations $K = 20$ times, which is used as a training set for the random forest. A random forest's trees are made up of a set of D with replacement sampling from D. The decrease in node impurity, which is weighted by the number of samples that reach the node, is used to compute the parameter significance for each node m in the regression tree. The significance parameter for node m is defined as follows:

$$\begin{aligned} I_m = &|P_m| H(P_m) - \left|P_{\{pos,m\}}\right| H\left(P_{\{pos,m\}}\right) \\ &- \left|P_{\{neg,m\}}\right| H\left(P_{\{neg,m\}}\right) \end{aligned} \tag{14}$$

$$H(P_m) = \frac{\sum_{x_{ref,i} \in P_m} (x_{ref,i} - \overline{x_{ref,P_m}})^2}{|P_m|} \tag{15}$$

Following the importance estimate process, the hyper-parameter with the lowest I_m is trimmed by setting. The pruning process significantly improves search efficiency. By altering the less important hyper-parameter to a value that consumes fewer resources, we may devote more computer resources to vital parameters. Algorithm 5 presents the process of CE-NAS credit scoring model construction.

As we can see from Algorithm 5, after a total of k epochs, we will eventually select the optimal J credit models from a large number of candidate models through hyper-parameter pruning and an efficient search strategy. In the subsequent experiments of this study, we set J=1 in order to facilitate comparison with other credit models, which means that the best credit model obtained from each search is selected for comparative analysis.

6. Security analysis

According to the suggested privacy protection mechanism scheme and associated blockchain-system components, the proposed EFCS system can satisfy all security criteria (e.g., PBFT (Practical Byzantine Fault Tolerance)).

1) Credit data privacy leakage problems. This study has superior data security in the work associated to the evaluation based on blockchain credit than other credit chains that just utilize blockchain for data exchange. All user data is not directly saved on the chain, but just the data index is, and the model is exchanged instead of data. This study employs blockchain as a platform for financial institutions to collaborate on federated learning, and each participant use blockchain to record each party's contribution and get rewards, which can incentivize each party to actively submit their own data.

2) Credit transaction privacy. To conceal users' true identities, our scheme's transaction anonymity is included within the EFCS system. The anonymous private key (which correlates to the sender's anonymous address) provides an SPK proof, which is used to chain transactions in the EFCS system. These transactions are confirmed using the anonymous addresses of the data providers rather than their long-term addresses. Aside from these anonymous addresses, the transactions give no other identifiable information. As a result, the proposed EFCS may be able to secure user privacy.

Algorithm 5 CE-NAS credit scoring model construction Algorithm

Input: $C_E = \phi; C_{2E} = \phi; C_{3E} = \phi$

▷ C_k represents credit scoring network trained for k epochs

Train (credit model, a, b)

▷The function that trains credit model, starting from epoch a, for totally $b - a$ epochs

$h = \phi; R = \phi$

▷ h indicates the history network set and R indicates hyperparameters set

while $|R_E|$ <lowest I_{init} **do**

  Prune redundant hyperparameters

  Extract the triplet loss parameters of the credit assessment targets R_t

  Set the appropriate initial data augmentation parameters $\eta(i)$ and $\lambda(i)$

  Add R_t to R

  credit model = Random architecture()

  credit model.accuracy = Train (credit model, 0, E)

  Add credit model to C_E and h

end while

for $i = 1 \rightarrow N$ **do**

  for i=1 to N_0 **do**

    Randomly sample credit model from $C_E; C_{2E}; C_{3E}$

    child model = Random mutate(credit model)

    child.accuracy = Train (child model, 0, E)

    Add child model model to C_E and h

  end for

  for model = top1 to N_1 credit models in C_E **do**

    credit model.accuracy = Train (credit model, E, $2E$)

    Move credit model from C_E to C_{2E}

  end for

  for credit model = top 1 to N_2 credit model in C_E **do**

    credit model.accuracy = Train(credit model, $2E$, $3E$)

    Move credit model from C_{2E} to C_{3E}

  end for

  Remove dead credit model from $C_E; C_{2E}; C_{3E}$

end for

return top J credit models in h, here $J = 1$ in our study

3) Tracking and Tracing. The manager in our EFCS system can use the certificate to verify the authenticity of each transaction's participants (which maps a long-term address to its real identity). When a suspicious transaction is identified, the manager discloses the anonymous address to recover the long address (through Tracing using the smart contract's private key), and then obtains the matching certificates from the smart contract.

4) Interception and modification resistance. Our EFCS system can withstand any hostile intent due to the transaction's non-malleability in our scheme. That is, any changes to these data will render the transaction null and void.

5) The two malicious roles are collaborating. When a node agrees to participate, they want to use EFCS's computational resources but do not want to pay the trustworthy nodes and EVM to carry out the smart contract. They will not utilize a hostile contract since they require a certain outcome. When nodes and buyers interact, a malicious buyer can designate maliciously chosen nodes as smart contract execution nodes. Malicious nodes will consume all of the seller's data.

7. Performance analysis

We will evaluate the proposed EFCS solution through simulations and experiments in this section to demonstrate the results of the system design, using the Python (3.7) programming language, the Sklearn library to complete the construction of the classification model and data partitioning. The experimental environment for this experiment used PySyft as a communication framework for federated learning. The CPU model was Intel(R) Core(TM) i7-6700K 4.00GHz.

7.1. Data description and experimental setting

In agreement with existing federated learning studies (Imteaj and Amini, 2022; Cheng et al., 2021), the dataset used was the credit scoring dataset "Give Me Some Credit" from a bank in Kaggle. This dataset contains 150,000 credit card payment information and income related information, of which the sample number of bad customers is 10,026, accounting for 6.684% of the dataset. This dataset contains 11-dimensional features, where one-dimensional labels indicate good customers and bad customers (customers who are more than 90 days late are called bad customers), see Table 4. On the one hand, the dataset is relatively large, which increases the likelihood of capturing diverse patterns and trends present in credit data. It also includes a wide range of variables, such as age, income, debt, and payment history, providing a reasonably comprehensive representation of factors that might affect credit risk. On the other hand, the dataset is known to have imbalanced classes, which can pose challenges in model training and evaluation. It is worth noting that not only has this dataset been extensively validated over the years by academic research, but its credibility is also supported by the default rate (6.684%), which corresponds to the delinquency rate of commercial bank loans at the time the dataset was released [1].

7.2. Federated Learning Performance Results

Table 5 presents the comparison of the performance of different types of federated learning. Table 6 shows the percentage of contribution values obtained from the participant's federated learning, with the first three rows for cases where the local dataset is not under attack and the last row for cases where there are four data providers and the dataset is under attack (reversing the values of the label columns).

From Tables 5 and 6 it can be seen that the contribution values are calculated as negative when the nodes under attack are involved.

In summary, our proposed EFCS training mechanism ensures efficient and accurate credit model sharing training without causing significant increase in training time, and therefore has superior system stability.

[1] https://fred.stlouisfed.org/series/DRALACBS

Table 4: The features of Give me some credit dataset

Features	Min	Max	Mean	stdDev
whether experienced 90 days past due delinquency or worse	No	Yes	-	-
revolving utilization of unsecured lines	0	50708	6.048	249.755
age	0	109	52.295	14.772
number of time 30-59 days past due not worse	0	98	0.421	4.193
debt ratio	0	392664	353.005	2037.819
monthly income	0	3008750	5348.139	13152.057
number of open credit lines and loans	0	58	8.453	5.146
number of times 90 days late	0	98	0.266	4.169
number real estate loans or lines	0	54	1.018	1.13
number of time 60-89 days past due not worse	0	98	0.24	4.155
number of dependents	0	20	0.737	1.107

Table 5: Comparison of the performance of different types of federated learning

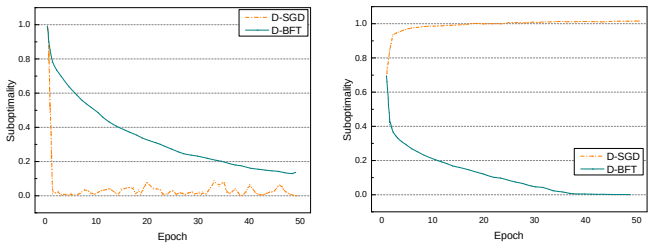
Type of federated learning	accuracy	macro-recall	macro-precision	macro-F1	AUC
Two-party	0.93643	0.57983	0.75725	0.61359	0.85176
Three-party	0.93640	0.57886	0.75735	0.61237	0.85172
Four-party	0.93654	0.57796	0.75988	0.61142	0.85207
Aggregation (non-federated learning model)	0.93650	0.58180	0.75739	0.61607	0.85189

Table 6: Comparison of the contribution percentage for different types of federated learning

Type of federated learning	A	B	C	D
Two-party	49.50%	50.50%	-	-
Three-party	33.33%	32.34%	34.33%	-
Four-party	25.18%	25.27%	24.75%	24.80%
Aggregation (non-federated learning model)	63.66%	62.75%	63.27%	-89.68%

7.3. System attack resistance

We further compared the performance of EFCS for two-party, three-party and four-party data providers under federated learning. They each owned a portion of bank credit card data, data users searched through the blockchain to the two parties that owned the data for federated learning modeling. In the modeling process, the coordinating party calculated its contribution based on the reduction of their respective contribution to the loss value, and separately calculated their respective contribution values, using their contribution values based on Shapley gain sharing contribution value, and finally recorded it in the current block of transactions. The results of the training conducted contains the training results of the aggregated data source, each participant in the local iteration 20 times before sending the gradient information encrypted to the coordinator for averaging, and in the coordinator iteration 500 times. The specific experimental results are are presented in Fig. 6. We further analyze the performance of our approach by comparing it with normal decentralized Byzantine fault-tolerant algorithm D-BFT.



(a) Random attack (b) Bit reversal attack

Figure 6: Loss and performance curves

The D-BFT algorithm performed slightly differently against different Byzantine attack methods, and the relative advantage of the algorithm against bit reversal attacks was not as great as against the other (Fig. 6). The relative advantage of the algorithm against bit reversal attacks is not as good as that against the other two Byzantine attacks, but its performance still had a significant advantage over the others. This type of experiments fully verifies the fault tolerance and good performance of the D-BFT algorithm. From the point of view of increasing the number of Byzantine nodes perspective, in bit reversal attacks, one can observe that the performance of the D-BFT algorithm decreased significantly and gradually as the number of Byzantine nodes increased. This situation was also as expected. The performance of the D-SGD algorithm was also different when facing the Byzantine attack methods. In contrast to the D-BFT algorithm, the relative performance of the D-SGD algorithm against bit reversal attacks was improved compared to the other two, and in this scenario the D-SGD algorithm and the Ubar algorithm win each other, second only to the D-BFT algorithm.

Table 7: Performance comparison of different zero-knowledge proof algorithms

Zero knowledge proof algorithm	Average execution time	Transaction size
verify_proof	2.315s	334B
verify_evaluate	3.134s	432B

From Fig. 6 we can see that the Groth16 algorithm took less time to generate the zkey (including the proof key p_k and verification key v_k), generate the proof and verify algorithms, basically keeping it under 1 second. The performance of the bn128 elliptic curve encryption algorithm was slightly higher than that of the bls12-381 algorithm. The time taken for the generation of the zkey and proof algorithms was positively related to the circuit complexity and the size of the circuit constraints, but the time taken for the verification algorithm was independent of the circuit complexity taking around 125 ms, which is in line with the description in Section V.

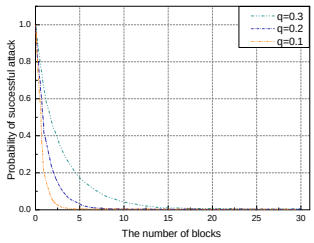


Figure 7: Malicious node attack success probability curve

As shown in Fig. 7, the three curves represent the probability graphs of successful attacks when the malicious nodes obtain bookkeeping rights with probabilities of 0.1, 0.2 and 0.3 respectively, from which it can be seen that the higher the number of blocks out the more difficult it was to change the blocks. As can be seen from Fig. 8, as

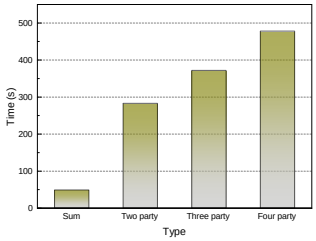
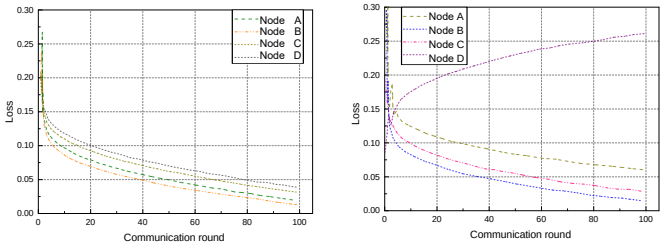


Figure 8: Time-consuming comparison of federated learning

long as each party honestly contributed its own local data for federated learning, the trained model was essentially the same as if the models were federated. The main difference was in the training time, as the more participants increased the communication overhead and the more computationally intensive the federation was on average, the more participants in the federation learning the more time it took, as shown in Fig. 8 below comparing the training time for different numbers of participants. It is clear from the above Fig. 9 that when a node with a dataset under



(a) Loss change without suffering an attack

(b) Loss change with suffering an attack

Figure 9: Change in value of losses

attack participated in federated learning, the loss value curve of its trained model on the local dataset was increasing, and the loss value curves of the remaining honest participants were decreasing, thus the evaluation of participant D according to the federated learning incentive method designed in this section was negative, indicating that the method is able to identify to some extent the participants whose datasets are under attack.

As the purpose of using federated learning in this paper is to address data silos and data privacy issues, it is not particularly concerned with the comparison of performance and effectiveness with related machine learning models, but only with whether the learning effect of federated learning on a distributed training set is comparable to that of traditional machine learning on a centralised dataset, a comparison that is presented in the paper. In work related to blockchain-based credit evaluation, the proposed EFCS model has better data security than other credit chains

that simply use blockchain for data sharing; all user data is not stored directly on the chain, but only the data index is stored with the chain, and it is the model rather than the data that is shared. This study uses blockchain as a platform for financial institutions to collaborate on federated learning, and each participant uses the blockchain to record each party's contribution and obtain benefits, which can encourage each party to actively contribute their own data.

In summary, our proposed EFCS training mechanism ensures efficient and accurate credit model sharing without causing significant increase in training time, and therefore has superior system stability.

7.4. Results and analysis of the performance for the EFCS approach

7.4.1. Comparison of the effects for different credit scoring methods

Our proposed method EFCS was thoroughly compared to benchmark methods such as individual base learners, homogeneous ensemble models, and heterogeneous ensemble models, as well as the basic automated model with traditional pipeline strategies. Both of the proposed model and the benchmark models were validated across four performance metrics using the aforementioned six credit datasets. The experiment made use of six credit datasets from either traditional financial institutions or peer-to-peer lending. German, Taiwan, Australia datasets are all freely accessible through the UCI machine learning repository. P2P lending datasets and credit card datasets were also used for additional validation. Two P2P datasets were gathered from one of China's earliest P2P lending platforms. Besides, a public dataset, the credit card dataset, was acquired from the website of the Kaggle for credit scoring. Table 8 demonstrates the description of the datasets in our study. We compared the proposed EFCS approach based

Table 8: Description of the datasets in our study

Dataset	Good/Bad	#Samples	#Features
P2P-1	1072/349	1421	17
P2P-2	1531/1000	2531	14
German	700/300	1000	24
Taiwan	23364/6636	30000	24
Australia	307/383	690	14
Credit card	600/492	1092	25

on automated machine learning with the individual classifier approach and the ensemble classifier approach, and the experimental results are presented under different research problems.

Table 9: Results of different credit scoring methods

Dataset	Evaluation measure	GPC	SVM	DT	LR	DNN	Bag-SVM	Bag-GPC	RF	Bstacking-LR-AVP	Bstacking-XGBoost-MV	Bstacking-all	vanilla NAS	EFCS
Credit card	accuracy	0.7124	0.7052	0.7125	0.7254	0.7301	0.7658	0.7874	0.7456	0.7865	0.7789	0.7856	0.7985	**0.8156**
	AUC	0.6865	0.6854	0.6235	0.6874	0.7014	0.7545	0.7456	0.7548	0.7958	0.7859	0.7489	0.7983	**0.8215**
	H-measure	0.1214	0.1236	0.2321	0.2401	0.2933	0.2426	0.2856	0.3025	**0.3102**	0.2726	0.3025	0.2956	0.3025
	Brier score	0.2654	0.2625	0.1985	0.2658	0.2665	0.2358	0.2658	0.2624	**0.2847**	0.2587	0.2685	0.2625	0.2485
P2P-1	accuracy	0.8654	0.8545	0.8985	0.8548	0.8821	0.8785	0.8856	0.8545	0.8958	0.8785	0.8956	0.9054	**0.9156**
	AUC	0.8215	0.8565	0.8548	0.8265	0.8234	0.8565	0.8548	0.8565	0.8545	0.8288	0.8658	0.8485	**0.8564**
	H-measure	0.5265	0.5625	0.5265	0.5485	0.5628	0.5685	0.5895	0.5856	0.5456	0.5785	0.5685	0.5256	**0.5865**
	Brier score	0.1252	0.2152	0.1658	0.2625	0.2093	0.1985	0.2562	0.1658	0.2156	0.1658	0.2652	**0.3056**	0.2985
P2P-2	accuracy	0.8634	0.8525	0.8975	0.8525	0.8657	0.8725	0.8856	0.8513	0.8922	0.8769	0.8941	0.8974	**0.9094**
	AUC	0.8218	0.8523	0.8518	0.8265	0.8545	**0.8548**	0.8532	0.8502	0.8231	0.8321	0.8658	0.8443	0.8523
	H-measure	0.5265	0.5625	0.5265	0.5483	0.5331	0.5682	0.5845	0.5813	0.5426	0.5755	0.5635	0.5222	**0.5865**
	Brier score	0.1245	0.2139	0.1633	0.2615	0.2265	0.1985	0.2542	0.1658	0.2126	0.1667	0.2621	**0.3032**	0.2943
German	accuracy	0.8125	0.8265	0.8482	0.8356	0.8267	0.8485	**0.8526**	0.8154	0.8265	0.8156	0.8256	0.8356	0.8154
	AUC	0.8654	0.8147	0.8728	0.8565	0.8611	0.8456	0.8685	0.8745	0.8565	0.8656	**0.8756**	0.8365	0.8665
	H-measure	0.4256	0.3652	0.4256	0.4125	0.4062	0.3658	0.4215	0.4125	0.3652	0.4125	0.3856	**0.4258**	0.3658
	Brier score	0.1523	0.1215	0.1352	0.1545	0.1362	0.1452	**0.1985**	0.1365	0.1547	0.1365	0.1587	0.1565	0.1548
Australian	accuracy	0.8587	0.8565	0.8415	0.8565	0.8425	0.8985	0.7856	0.8587	0.8214	0.8565	0.8958	0.8985	**0.9049**
	AUC	0.7585	0.8156	0.7458	0.8236	0.8165	0.8485	0.8658	0.8754	0.8854	0.8955	0.8456	0.8756	**0.9017**
	H-measure	0.5460	0.5365	0.5874	0.5698	0.5733	0.5987	0.5985	0.6054	0.5856	**0.6547**	0.6125	0.5895	0.5958
	Brier score	0.2654	0.3152	0.2658	0.2658	0.2722	0.3015	0.2985	0.3156	0.2985	**0.3215**	0.2895	0.3036	0.3205
Taiwan	accuracy	0.8782	0.8873	0.8524	0.8594	0.8614	0.9014	0.7964	0.8654	0.8352	0.8654	0.8958	0.8991	**0.9094**
	AUC	0.7654	0.8242	0.7541	0.8363	0.8356	0.8564	0.8783	0.8893	0.8883	0.8991	0.8487	0.8872	**0.9101**
	H-measure	0.5584	0.5583	0.5884	0.5694	0.5732	0.5998	0.5991	0.6102	0.5903	**0.6573**	0.6154	0.5899	0.5984
	Brier score	0.2693	0.3201	0.2695	0.2673	0.2769	0.3015	0.2985	0.3201	0.2984	**0.3301**	0.2935	0.3056	0.3268

Table 9 shows the results of the individual classifier, homogeneous ensemble, heterogeneous ensemble, and our automated EFCS. For each classifier type, several major findings have been found. In terms of individual classifiers, we able to find that all of the individual classifier methods performed worse overall than the ensemble classifier methods on different datasets. Furthermore, we can also find that the performance of the GPC (Gaussian process classification) method was relatively stable across different datasets, indicating that GPC is a powerful technique when the hyper-parameters are fine-tuned. The performance of the SVM (support vector machine), DNN (Deep Neural Networks), DT (decision tree) and LR (logistic regression) methods fluctuated over different datasets, mainly due to the relative simplicity of the models of these three methods, which makes it difficult to achieve better performance in the construction of credit scoring models through appropriate hyper-parameter settings. From the conventional binary choice models, the LR method performed well across the datasets, with relatively high AUC values indicating solid performance in the identification of bad customers even for the most imbalanced Taiwan dataset.

In terms of the ensemble methods, all base learners improved in most datasets. We can find that Bag-SVM (bagging SVM), Bstacking-al and Bag-GPC each achieved the best credit score performance once, which also indicates that the ensemble approach improves the performance of the credit model compared to the individual classifier approach. In addition, Bstacking-LR-AV and Bstacking-XGBoost-M each achieved the best credit score performance twice, which also indicates that the XGBoost method can greatly improve the classification performance of the model with a suitable ensemble approach.

We can clearly see from Table 9 that if the vanilla NAS method was used for credit model searching, the performance already outperformed the ensemble classifier method and achieved three times the optimal credit score performance. In contrast, our proposed EFCS method, which introduces *AP*-Triplet loss and an improved CE-NAS combination, achieved superior credit score performance to the vanilla NAS method, which also demonstrates the effectiveness of the EFCS method.

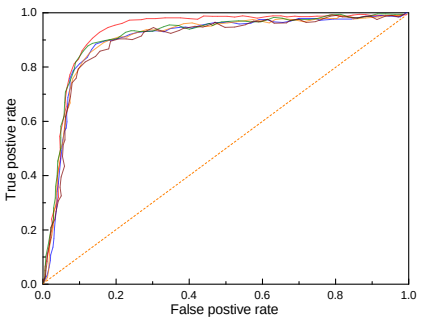
To summarize, while the individual classifier advantages from its simplicity, it cannot compete with the homogeneous and heterogeneous ensemble models in terms of classification performance. While compared to the performance of base learners when generating an individual classifier, the homogeneous ensemble considerably enhances their performance. Ensemble learning is especially beneficial to those base learners who have a lot of variability. The vanilla NAS method achieves better credit scoring performance than the ensemble classifier method. In comparison, our EFCS introduces a more efficient neural architecture search mechanism CE-NAS, which can substantially improve the credit scoring performance compared to the vanilla NAS method.

To statistically evaluate the classification performance of the credit scoring methods, a nonparametric Friedman test was conducted across the six datasets. The significance of the differences between the average ranking of the tested methods is indicated by the Friedman p-values presented in Table 10. For a more thorough comparison with the best performing method, the Iman-Davenport post hoc procedure was used to adjust the significance level of the results. The results in Table 10 show that the EFCS model achieved either the top or second position among the methods in terms of Accuracy, Brier score and AUC. In addition, EFCS performed significantly similarly to the best performer in terms of H-measure.

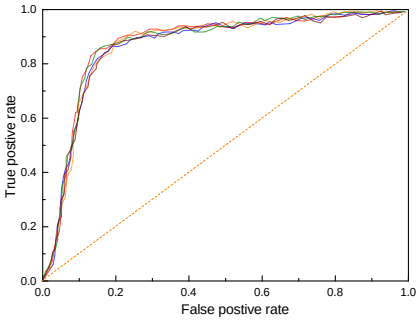
7.4.2. Analysis of ROC curves for different methods

The ROC curves of the proposed and benchmark models are shown in Fig. 10 for each dataset. We can observe that our EFCS achieved the highest AUC on all six credit datasets, with the different approaches achieving the best performance on the Australian dataset and slightly worse performance on the P2P-2 dataset than on the other datasets. We found that after data balancing, the distribution of the Australian dataset was more balanced, so the credit scores of the different models performed slightly better than the other datasets.

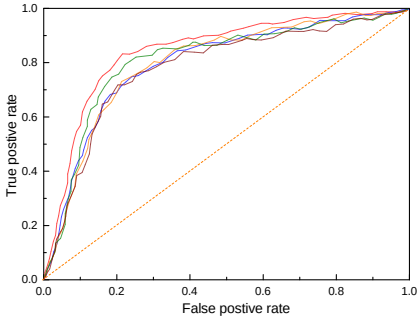
To summarize, our proposed EFCS beats all other classifiers in terms of AUC for most datasets, albeit its ROC curve is not always higher than the baseline models. When it comes to rejecting most loan applications, EFCS does a better job of identifying possible problematic applicants. In addition, the degree of imbalance of different datasets will also affect the credit scoring performance of the model to some extent.



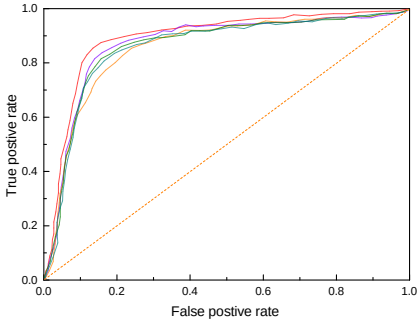
(a) ROC curves for Australian dataset



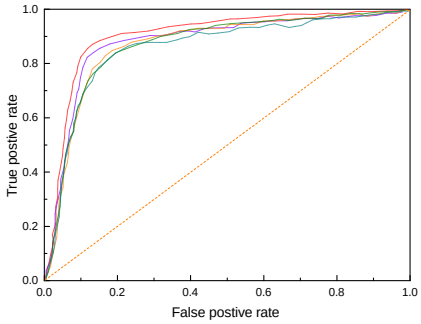
(b) ROC curves for credit card dataset



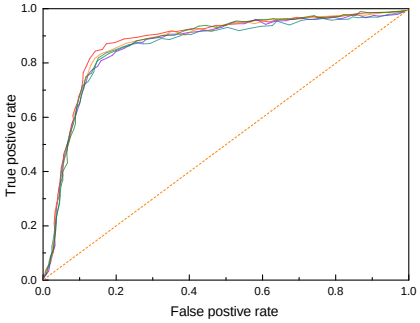
(c) ROC curves for German dataset



(d) ROC curves for P2P-1 dataset



(e) ROC curves for Taiwan dataset



(f) ROC curves for P2P-2 dataset

Figure 10: ROC curves for different credit datasets, where the red, purple, blue, yellow and green curves indicate the ROC curve results for the five credit scoring methods EFCS, Bstacking-XGBoost-MV, BAG-SVM, XGBoost and RF, respectively.

Table 10: Results of Friedman nonparametric test

Method	Ranking Accuracy	Ranking AUC	Ranking H-measure	Ranking Brier score
Bag-GPC	7.0	**4.9**	**4.0**	**5.0**
Bag-SVM	**5.0**	**5.2**	**6.3**	8.3
Bstacking-all	**5.2**	**4.5**	**4.6**	**4.7**
Bstacking-LR-AVP	7.4	**7.2**	8.1	**5.9**
Bstacking-XGBoost-MV	7.8	**7.0**	**4.0**	**7.1**
DNN	8.3	8.0	8.5	8.0
DT	6.8	**7.6**	8.9	11.8
LR	9.2	9.2	9.2	**7.2**
RF	9.7	**5.8**	**3.5**	7.8
SPC	9.4	11.5	10.4	10.7
SVM	9.6	8.7	10.9	**7.0**
vanilla NAS	**2.8**	**6.8**	7.8	**3.8**
EFCS	**2.8**	**4.8**	**4.8**	**3.8**
Friedman p-value	0.0063	0.0004	0.0013	0.0056

Significantly similar performance at $p < 0.10$ as the best performer is in bold.

8. Conclusion and future work

The use and evaluation of user credit data by financial institutions under the restriction of maintaining user privacy is the subject of this paper. At the moment, credit evaluation is growing in popularity and major Internet companies have access to enormous amounts of data that can be used to accurately evaluate user credit. In order to deal with the issue of credit data silos, in this research, we introduced EFCS, a federated learning and credit model sharing scheme that is explicable. The experimental results show that EFCS is secure and performs better than other competitive systems.

Our explainable federated learning strategy's performance and security could yet be enhanced. First, in future research, our EFCS method will consider incorporating credit model sharing mechanisms in a multi-client and distributed model, such as (Huang et al., 2021; Wang et al., 2022; Li et al., 2021; Qin et al., 2021), thus increasing the support to a different environment will allow for a wider range of applications. A distributed zero-knowledge proof system could be used in future research to increase productivity. Additionally, since the efficiency of verification at the underlying layer is higher than that of the zero-knowledge proof procedure used in this paper in an Ethereum smart contract, attention might be considered in the future, such as (Choi et al., 2020; Biswas et al., 2023; Jiang et al., 2022; Wang et al., 2023b), to improve the performance and efficiency of federated learning on the blockchain.

References

AbdulRahman, S., Tout, H., Ould-Slimane, H., Mourad, A., Talhi, C., and Guizani, M. (2021). A survey on federated learning: The journey from centralized to distributed on-site learning and beyond. *IEEE Internet of Things Journal*, 8(7):5476–5497.

Amini, H., Bichuch, M., and Feinstein, Z. (2022). Decentralized payment clearing using blockchain and optimal bidding. *European Journal of Operational Research*.

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115.

Berdik, D., Otoum, S., Schmidt, N., Porter, D., and Jararweh, Y. (2021). A survey on blockchain for information systems management and security. *Information Processing & Management*, 58(1):102397.

Biswas, D., Jalali, H., Ansaripoor, A. H., and De Giovanni, P. (2023). Traceability vs. sustainability in supply chains: The implications of blockchain. *European Journal of Operational Research*, 305(1):128–147.

Bodkhe, U., Tanwar, S., Parekh, K., Khanpara, P., Tyagi, S., Kumar, N., and Alazab, M. (2020). Blockchain for industry 4.0: A comprehensive review. *IEEE Access*, 8:79764–79800.

Bücker, M., Szepannek, G., Gosiewska, A., and Biecek, P. (2022). Transparency, auditability, and explainability of machine learning models in credit scoring. *Journal of the Operational Research Society*, 73(1):70–90.

Cheng, K., Fan, T., Jin, Y., Liu, Y., Chen, T., Papadopoulos, D., and Yang, Q. (2021). Secureboost: A lossless federated learning framework. *IEEE Intelligent Systems*, 36(6):87–98.

Choi, T.-M., Guo, S., Liu, N., and Shi, X. (2020). Optimal pricing in on-demand-service-platform-operations with hired agents and risk-sensitive customers in the blockchain era. *European Journal of Operational Research*, 284(3):1031–1042.

Dai, H.-N., Zheng, Z., and Zhang, Y. (2019). Blockchain for internet of things: A survey. *IEEE Internet of Things Journal*, 6(5):8076–8094.

Dastile, X., Celik, T., and Vandierendonck, H. (2022). Model-agnostic counterfactual explanations in credit scoring. *IEEE Access*, 10:69543–69554.

Dumitrescu, E., Hué, S., Hurlin, C., and Tokpavi, S. (2022). Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *European Journal of Operational Research*, 297(3):1178–1192.

Elsken, T., Metzen, J. H., and Hutter, F. (2019). Neural architecture search: A survey. *The Journal of Machine Learning Research*, 20(1):1997–2017.

Gai, K., Guo, J., Zhu, L., and Yu, S. (2020). Blockchain meets cloud computing: A survey. *IEEE Communications Surveys & Tutorials*, 22(3):2009–2030.

Gunnarsson, B. R., Vanden Broucke, S., Baesens, B., Óskarsdóttir, M., and Lemahieu, W. (2021). Deep learning for credit scoring: Do or don't? *European Journal of Operational Research*, 295(1):292–305.

Guo, Y. and Liang, C. (2016). Blockchain application and outlook in the banking industry. *Financial Innovation*, 2(1):1–12.

Hassija, V., Bansal, G., Chamola, V., Kumar, N., and Guizani, M. (2020). Secure lending: Blockchain and prospect theory-based decentralized credit scoring model. *IEEE Transactions on Network Science and Engineering*, 7(4):2566–2575.

Huang, K., Zhang, X., Mu, Y., Rezaeibagha, F., and Du, X. (2021). Scalable and redactable blockchain with update and anonymity. *Information Sciences*, 546:25–41.

Imteaj, A. and Amini, M. H. (2022). Leveraging asynchronous federated learning to predict customers financial distress. *Intelligent Systems with Applications*, 14:200064.

Jammalamadaka, K. R. and Itapu, S. (2022). Responsible ai in automated credit scoring systems. *AI and Ethics*, pages 1–11.

Jiang, S., Li, Y., Wang, S., and Zhao, L. (2022). Blockchain competition: The tradeoff between platform stability and efficiency. *European Journal of Operational Research*, 296(3):1084–1097.

Khan, L. U., Pandey, S. R., Tran, N. H., Saad, W., Han, Z., Nguyen, M. N., and Hong, C. S. (2020). Federated learning for edge networks: Resource optimization and incentive mechanism. *IEEE Communications Magazine*, 58(10):88–93.

Kim, H., Park, J., Bennis, M., and Kim, S.-L. (2019). Blockchained on-device federated learning. *IEEE Communications Letters*.

Kriebel, J. and Stitz, L. (2022). Credit default prediction from user-generated text in peer-to-peer lending using deep learning. *European Journal of Operational Research*, 302(1):309–323.

Latif, S., Idrees, Z., Ahmad, J., Zheng, L., and Zou, Z. (2021). A blockchain-based architecture for secure and trustworthy operations in the industrial internet of things. *Journal of Industrial Information Integration*, 21:100190.

Li, D., Han, D., Crespi, N., Minerva, R., and Li, K.-C. (2023). A blockchain-based secure storage and access control scheme for supply chain finance. *The Journal of Supercomputing*, 79(1):109–138.

Li, G., Ren, X., Wu, J., Ji, W., Yu, H., Cao, J., and Wang, R. (2021). Blockchain-based mobile edge computing system. *Information Sciences*, 561:70–80.

Lim, W. Y. B., Luong, N. C., Hoang, D. T., Jiao, Y., Liang, Y.-C., Yang, Q., Niyato, D., and Miao, C. (2020). Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*.

Lin, C., He, D., Huang, X., Khan, M. K., and Choo, K.-K. R. (2020). Dcap: A secure and efficient decentralized conditional anonymous payment system based on blockchain. *IEEE Transactions on Information Forensics and Security*, 15:2440–2452.

Liu, W., Fan, H., and Xia, M. (2022). Credit scoring based on tree-enhanced gradient boosting decision trees. *Expert Systems with Applications*, 189:116034.

Medina-Olivares, V., Calabrese, R., Crook, J., and Lindgren, F. (2022). Joint models for longitudinal and discrete survival data in credit scoring. *European Journal of Operational Research*.

Moscato, V., Picariello, A., and Sperlí, G. (2021). A benchmark of machine learning approaches for credit score prediction. *Expert Systems with Applications*, 165:113986.

Mothukuri, V., Parizi, R. M., Pouriyeh, S., Huang, Y., Dehghantanha, A., and Srivastava, G. (2021). A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115:619–640.

Pławiak, P., Abdar, M., and Acharya, U. R. (2019). Application of new deep genetic cascade ensemble of svm classifiers to predict the australian credit scoring. *Applied Soft Computing*, 84:105740.

Qin, X., Huang, Y., Yang, Z., and Li, X. (2021). Lbac: A lightweight blockchain-based access control scheme for the internet of things. *Information Sciences*, 554:222–235.

Shen, M., Liu, H., Zhu, L., Xu, K., Yu, H., Du, X., and Guizani, M. (2020). Blockchain-assisted secure device authentication for cross-domain industrial iot. *IEEE Journal on Selected Areas in Communications*, 38(5):942–954.

Štěpánková, B. (2021). Bank-sourced credit transition matrices: Estimation and characteristics. *European Journal of Operational Research*, 288(3):992–1005.

Tripathi, D., Edla, D. R., Cheruku, R., and Kuppili, V. (2019). A novel hybrid credit scoring model based on ensemble feature selection and multilayer ensemble classification. *Computational Intelligence*, 35(2):371–394.

Wang, C., Chen, X., Xu, X., and Jin, W. (2023a). Financing and operating strategies for blockchain technology-driven accounts receivable chains. *European Journal of Operational Research*, 304(3):1279–1295.

Wang, C., Chen, X., Xu, X., and Jin, W. (2023b). Financing and operating strategies for blockchain technology-driven accounts receivable chains. *European Journal of Operational Research*, 304(3):1279–1295.

Wang, Q. and Su, M. (2020). Integrating blockchain technology into the energy sector—from theory of blockchain to research and application of energy blockchain. *Computer Science Review*, 37:100275.

Wang, Y., Wang, Z., Zhao, M., Han, X., Zhou, H., Wang, X., and Koe, A. S. V. (2022). Bsm-ether: Bribery selfish mining in blockchain-based healthcare systems. *Information Sciences*, 601:1–17.

Xia, Y., Liu, C., Da, B., and Xie, F. (2018). A novel heterogeneous ensemble credit scoring model based on bstacking approach. *Expert Systems with Applications*, 93:182–199.

Yang, F., Qiao, Y., Abedin, M. Z., and Huang, C. (2022a). Privacy-preserved credit data sharing integrating blockchain and federated learning for industrial 4.0. *IEEE Transactions on Industrial Informatics*, 18(12):8755–8764.

Yang, F., Qiao, Y., Huang, C., Wang, S., and Wang, X. (2021a). An automatic credit scoring strategy (acss) using memetic evolutionary algorithm and neural architecture search. *Applied Soft Computing*, 113:107871.

Yang, F., Qiao, Y., Qi, Y., Bo, J., and Wang, X. (2022b). Bacs: blockchain and automl-based technology for efficient credit scoring classification. *Annals of Operations Research*, pages 1–21.

Yang, F., Qiao, Y., Qi, Y., Bo, J., and Wang, X. (2022c). Bmp: A blockchain assisted meme prediction method through exploring contextual factors from social networks. *Information Sciences*, 603:262–288.

Yang, F., Qiao, Y., Wang, S., Huang, C., and Wang, X. (2021b). Blockchain and multi-agent system for meme discovery and prediction in social network. *Knowledge-Based Systems*, 229:107368.

Yfanti, S., Karanasos, M., Zopounidis, C., and Christopoulos, A. (2023). Corporate credit risk counter-cyclical interdependence: A systematic analysis of cross-border and cross-sector correlation dynamics. *European Journal of Operational Research*, 304(2):813–831.

Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., and Gao, Y. (2021). A survey on federated learning. *Knowledge-Based Systems*, 216:106775.

Zhang, J., Tan, R., Su, C., and Si, W. (2020). Design and application of a personal credit information sharing platform based on consortium blockchain. *Journal of Information Security and Applications*, 55:102659.

Zhang, W., He, H., and Zhang, S. (2019). A novel multi-stage hybrid model with enhanced multi-population niche genetic algorithm: An application in credit scoring. *Expert Systems with Applications*, 121:221–232.

Appendix 1: Verification procedure

Proof of correctness:

$$
\begin{aligned}
(X,Y) &= \left(\prod_{n=1}^{n=|\mathcal{D}_3|} X_n, {}^{n=|\mathcal{D}_3|}_{n=1} Y_n \right) \\
&= \left(w^{\sum_{n\in\mathcal{D}_3} HF_{\delta,\rho}(x_n)}, h^{\sum_{n\in\mathcal{D}_3} HF_{\delta,\rho}(x_n)} \right) \\
&= \left(w^{HF_{\delta,\rho}\left(\sum_{n\in\mathcal{D}_3} x_n\right)}, h^{HF_{\delta,\rho}\left(\sum_{n\in\mathcal{D}_3} x_n\right)} \right) \\
&= (X', Y') \\
e(X,h) &= e\left(w^{HF_{\delta,\rho}(\sigma)}, h\right) = e\left(w, h^{HF_{\delta,\rho}(\sigma)}\right) \\
&= e(w,Y)
\end{aligned}
\tag{16}
$$

$$
\begin{aligned}
e(L,h) &= e\left(g^{\sum_{n\in\mathcal{D}_3} \gamma_n\gamma+\nu_n\nu-HF_{\delta,\rho}(x_n)}, h\right)^{1/d} \\
&= e\left(w, h^{\sum_{n\in\mathcal{D}_3} \gamma_n\gamma+\nu_n\nu-HF_{\delta,\rho}(x_n)}\right)^{1/d} \\
&= e(w,Q) \\
\Phi &= e(X,h)\cdot e(L,h)^d \\
&= e\left(w^{HF_{\delta,\rho}(\sigma)}, h\right)\cdot e\left(w^{\sum_{n\in\mathcal{D}_3} \gamma_n\gamma+\nu_n\nu-HF_{\delta,\rho}(x_n)}, h\right) \\
&= e(w,h)^{\sum_{n\in\mathcal{D}_3} \gamma_n\gamma+\nu_n\nu}
\end{aligned}
\tag{17}
$$