FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Privacy and Profit: Monetization Strategies for Privacy-Preserving Collaborative Forecasting with Blockchain Support

José Luís Rodrigues

Mestrado em Engenharia Informática e Computação

Supervisor: Prof. Carlos Soares

Company Supervisors:
Carla Gonçalves
José Ricardo Andrade

July 30, 2025

Privacy and Profit: Monetization Strategies for Privacy-Preserving Collaborative Forecasting with Blockchain Support

José Luís Rodrigues

Mestrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

President: Prof. Daniel Mendes
Examiner: Doctor João Vinagre
Supervisor: Prof. Carlos Soares

July 30, 2025

Abstract

Forecasting is a critical task for the integration of renewable energy sources. It helps address their inherent unpredictability by enabling better planning, reducing reliance on fossil backups, and supporting a more stable and efficient grid. Studies have shown that forecasting accuracy improves when energy producers collaborate by sharing data. However, since these producers often compete in the same market, privacy concerns discourage such cooperation.

Although recent systems have proposed privacy-preserving collaborative learning frameworks, privacy alone is not sufficient to sustain participation. These frameworks remain vulnerable to free-riding and unbalanced gains, where some agents contribute valuable data without receiving proportional benefits. To address this, this thesis proposes a system that integrates a contribution-based incentive mechanism into an existing privacy-preserving forecasting protocol. Payments are distributed through blockchain based on the quantified impact of each participant.

Moreover, introducing monetary incentives can motivate dishonest behaviour, as participants may attempt to manipulate their reported contributions to increase rewards. To address this, the system includes two validation strategies: one based on perturbations, and another using zero-knowledge proofs, particularly zk-SNARKs.

Experimental results on both synthetic and real-world datasets show that the proposed system preserves predictive accuracy while enabling fair compensation and resistance to dishonest behaviour. In addition, it was shown that the system's forecasting performance is comparable to centralized approaches, while preserving privacy and promoting collaboration.

Keywords: Federated Learning, Renewable Energy Forecasting, Data Privacy, Incentive Mechanism, Blockchain Technology, Decentralized Systems, Zero-Knowledge Proofs, LASSO-VAR Models, Collaborative Models

Resumo

A previsão é uma tarefa fundamental para a integração de fontes de energia renovável. Esta ajuda a lidar com a inerente imprevisibilidade destas fontes, permitindo um planeamento mais eficiente, reduzindo a dependência de reservas fósseis e promovendo uma operação mais estável da rede elétrica. Estudos demonstraram que a qualidade das previsões melhora quando os produtores de energia colaboram através da partilha de dados. No entanto, dado que estes produtores competem frequentemente no mesmo mercado, as preocupações com a privacidade dificultam esta cooperação.

Embora tenham sido recentemente propostos sistemas que permitem aprendizagem colaborativa com preservação da privacidade, esta, por si só, não é suficiente para garantir a participação. Estes sistemas continuam vulneráveis a comportamentos oportunistas, em que alguns agentes beneficiam do modelo global sem contribuírem de forma proporcional. Para colmatar este problema, esta dissertação propõe um sistema que integra um mecanismo de incentivos baseado nas contribuições individuais num protocolo já existente de previsão colaborativa com preservação da privacidade. Os pagamentos são distribuídos através de tecnologia *blockchain*, com base no impacto quantificado de cada participante.

Além disso, a introdução de incentivos monetários pode motivar comportamentos desonestos, uma vez que os participantes podem tentar manipular as suas contribuições para aumentar as recompensas. Para mitigar este risco, o sistema inclui duas estratégias de validação: uma baseada em perturbações e outra baseada em provas de conhecimento zero usando zk-SNARKs.

Os resultados experimentais, obtidos com dados sintéticos e reais, mostram que o sistema proposto preserva a exatidão da previsão, permitindo simultaneamente uma compensação justa e resistência a comportamentos desonestos. Adicionalmente, foi demonstrado que o desempenho de previsão do sistema é comparável ao de abordagens centralizadas, enquanto preserva a privacidade e promove a colaboração.

Palavras-chave: Aprendizagem Federada, Previsão de Energia Renovável, Privacidade de Dados, Mecanismo de Incentivos, Tecnologia *Blockchain*, Sistemas Descentralizados, Provas de Conhecimento Zero, Modelo LASSO-VAR, Modelos Colaborativos

UN Sustainable Development Goals

The United Nations Sustainable Development Goals (SDGs) provide a global framework to achieve a better and more sustainable future for all. It includes 17 goals to address the world's most pressing challenges, including poverty, inequality, climate change, environmental degradation, peace, and justice.

Renewable Energy Sources (RES), such as wind and solar, are central to global decarbonization goals, but their variability poses major challenges for reliable grid operation. Accurate forecasting is essential to balance supply and demand, reduce waste, and avoid unnecessary reliance on fossil backups. This work contributes to a more efficient and cooperative use of RES by proposing a privacy-preserving and incentive-compatible framework for collaborative forecasting across independent energy producers.

The specific Sustainable Development Goals addressed in this work are:

SDG 7 Ensure access to affordable, reliable, sustainable and modern energy for all

SDG 8 Promote sustained, inclusive and sustainable economic growth, full and productive employment and decent work for all

SDG 9 Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation

SDG 12 Ensure sustainable consumption and production patterns

SDG 13 Take urgent action to combat climate change and its impacts

Table 1 outlines how this thesis contributes to each of the identified SDGs, detailing the relevant targets, specific contributions, and performance indicators used to assess impact.

Table 1: Contributions of this thesis to selected UN Sustainable Development Goals, including relevant targets, corresponding system contributions, and performance indicators.

SGD	**Target**	**Contribution**	**Performance Indicators and Metrics**
7	7.1	Enables secure data sharing between energy producers, improving forecast coordination and reducing uncertainty in energy supply. This supports more reliable and wider access to energy.	Improvement in forecast error metrics (e.g., RMSE, MAE); reduction in renewable energy waste through improved forecasting; increased share of population reliably served by clean energy sources.
	7.2	By improving the accuracy of renewable energy forecasting using federated learning, the system helps reduce reliance on non-renewable backup sources and enables greater integration of renewables into the grid.	Increase in renewable energy share in the total final energy consumption; improvement in forecast error metrics (e.g., RMSE, MAE).
8	8.2	Enables safe data sharing and monetization between energy companies, supporting digital innovation and promoting the creation of new services. Blockchain ensures transparent and verifiable compensation.	Number of contributors monetizing data; volume of forecast-sharing transactions; emergence of new roles or services based on data collaboration.
9	9.5	Proposes a novel, privacy-preserving framework for collaborative forecasting, lowering barriers to innovation and helping modernize digital infrastructure in the energy sector.	Number of organizations adopting collaborative forecasting; integration of privacy/fairness mechanisms in industrial tools; references in research or technical standards.
12	12.2	Improves the efficiency of renewable energy use by reducing wasted production and backup activation, through better forecasting and secure data collaboration.	Decrease in unused renewable output; fewer reserve activations; better match between production and demand.
13	13.2	Supports climate strategies by improving the reliability of renewable energy, helping reduce fossil fuel use, and enabling cleaner grid planning.	Reduction in fossil energy demand; improved forecast-based energy planning; decreased greenhouse gas emissions through increased share of renewables in daily operation.

Acknowledgments

Getting to the end of this project was never a straightforward process. What follows is not only the result of countless hours of my own, but also the effort, support, and presence of a group of people who made it possible to get here at all.

First and foremost, my deep thanks go to my supervisors. To Prof. Carlos Soares, for all the thought-provoking insights that pushed this work forward in meaningful ways. To Carla Gonçalves, for the care, attention, and relentless effort you invested in every part of this work, as if it were your own. And to Ricardo Andrade, for making this thesis possible from the start. I've learned a great deal from your feedback and support, not just throughout this project, but over the whole time working under your guidance.

Then, on an institutional level, I'd like to express my gratitude to INESC TEC, and in particular to everyone at CPES, for welcoming me from day one. It has been a pleasure to work alongside all of you.

Finally, on a personal note, I want to acknowledge all of my family, but particularly my parents, for always giving everything they had to ensure I received the best education possible. To Maria, for being able to both inspire and challenge me on a daily basis. Your support during the hardest moments meant more than I can put into words. To all the Pupils and extended Pupils, for making this degree about more than just work. And to Gonçalo, João, Maria João, and all the xeminols, for being such a defining part of my life and for shaping me into who I am today.

José Luís Rodrigues

"Live to the point of tears."

Albert Camus

Contents

List of Figures

List of Tables

Abbreviations and Symbols

ADMM	Alternating Direction Method of Multipliers
CV	Coefficient of Variation
DO	Data Owner
FL	Federated Learning
LASSO	Least Absolute Shrinkage and Selection Operator
MAE	Mean Absolute Error
MSE	Mean Squared Error
MO	Model Owner
nMAE	Normalized Mean Absolute Error
nRMSE	Normalized Root Mean Squared Error
P2P	Peer-to-Peer
RC	Remote Contributions
RES	Renewable Energy Sources
RMSE	Root Mean Squared Error
SD	Standard Deviation
SMPC	Secure Multi-Party Computation
VAR	Vector Autoregressive Model
VCG	Vickrey–Clarke–Groves
ZKP	Zero-Knowledge Proof
zk-SNARK	Zero-Knowledge Succinct Non-Interactive Argument of Knowledge

Chapter 1

Introduction

Renewable Energy Sources (RES) represented 25% of the energy consumed in the EU in 2023, according to data from Eurostat [13]. This share has steadily increased in recent years, driven by the growing adoption of technologies such as solar and wind power, which are highly dependent on weather conditions. Such sensitivity means that production can fluctuate significantly with factors such as irradiance or wind speed. This natural variability introduces uncertainty into energy system planning, potentially leading to energy shortages or, alternatively, periods of energy surplus. The unpredictability of these generation sources affects both electricity grids and markets, creating challenges in maintaining a stable balance between supply and demand. A common implication is the need to activate balancing reserves, which introduces additional costs and can ultimately influence electricity market prices.

Given the uncertainty and cost imbalances introduced by RES variability, accurate forecasting becomes essential. To address this challenge, energy companies may benefit from forecasting algorithms that can help in dealing with the variable nature of these resources. A promising aspect of this approach is that a large amount of geographically distributed RES data is regularly produced, and combining this data can greatly enhance forecasting. Because renewable energy sources are geographically dispersed, integrating data from multiple sites helps capture broader weather patterns, leading to more accurate forecasting. For example, forecast accuracy was improved by 1.5% to 4.6% across different lead times in a case study where data from 19 geographically dispersed wind farms was combined, as opposed to relying on a single site [64].

Despite the benefits of collaborative forecasting, companies are often wary of sharing data, as other participants are most likely their competitors operating in the same market. The sensitive nature of this information creates a barrier to cooperation, even when a distributed prediction model could lead to better forecasting for all participants. To overcome this, it is crucial to develop collaborative models that preserve data privacy while enabling joint forecasting. Recent work has explored such approaches, showing that privacy can be maintained even in distributed learning settings [20].

While privacy-preserving systems address a key challenge of cooperation, they may not be sufficient to encourage competitors to participate. In many forecasting scenarios, the benefits

of collaboration are unevenly distributed. Depending on spatial-temporal patterns, some agents may contribute valuable information that improves the model for others without gaining a direct advantage themselves. For instance, a wind station that reports passing wind events may help downstream stations improve their short-term forecasts, but gains little to no advantage from the collaboration. This type of imbalance raises the need for mechanisms that promote fairness and ensure that participants are properly recognized for their contributions.

1.1 Motivation

Current privacy-preserving collaboration models, such as Federated Learning, often lack mechanisms to accurately quantify and reward individual contributions. This lack of incentive structures can discourage participation, especially in competitive markets where data is a valuable asset. Meanwhile, existing data markets, which are platforms designed to facilitate data exchange, typically rely on centralized authorities and thus compromise data privacy and limit trust among participants.

In light of the limitations identified in existing privacy-preserving collaboration models and data markets, there is a clear need for a new framework that supports three essential properties: distributed collaborative model training, preservation of data privacy, and fair incentive mechanisms based on participants' contributions.

These three components are closely interrelated. Distributed training depends on strong privacy guarantees to ensure that sensitive data remains within local domains. At the same time, the ability to assess and reward individual contributions is critical to sustain engagement, particularly in competitive environments. However, this becomes significantly more difficult when privacy constraints limit the visibility of individual inputs. Bringing these elements together in a decentralized setting introduces further challenges, particularly around the secure and transparent allocation of incentives without relying on a central authority.

1.2 Research Objectives

The goal of this thesis is to design and evaluate a distributed forecasting system that supports collaborative model training while ensuring data privacy and fair participant compensation. To achieve this, the proposed framework integrates Federated Learning with contribution-based rewards, supported by blockchain technology to provide transparency, auditability, and a secure mechanism for rewarding participants.

A key challenge lies in accurately evaluating individual contributions under strict privacy constraints. This could be accomplished without relying on full centralization, which could compromise trust and scalability. Instead, the system explores how far central coordination can be minimized while still preserving the essential properties of fairness, privacy, and predictive performance.

Ultimately, the aim of this research is to enable broader collaboration in the RES sector by ensuring that contributors are rewarded proportionally to their impact, without exposing sensitive data.

Hypothesis

Building on the problem defined in the previous section, this work is guided by the following hypothesis:

> *A distributed system for generating Renewable Energy forecasts can achieve comparable performance to centralized algorithms, while preserving data privacy and offering fair compensation to participants.*

Research Questions

The hypothesis leads to the following research questions:

RQ1 Can a distributed forecasting system be extended with a mechanism to offer participants fair compensation for contributions while preserving accurate predictions and data privacy?

RQ2 How do the contribution assessments derived from the distributed system compare in effectiveness and fairness to those generated by existing centralized algorithms under similar conditions?

1.3 Document Structure

The remainder of this document is structured as follows. Chapter 2 reviews the state of the art in privacy-preserving collaborative forecasting, including incentive mechanisms, zero-knowledge proofs, and blockchain-based approaches. Chapter 3 presents the proposed framework, outlining design decisions and expected outcomes. Chapter 4 describes how the framework was implemented and tested, detailing the experimental setup and the datasets used. Chapter 5 reports the results of the conducted experiments and evaluates the system with respect to the defined research questions. Chapter 6 concludes the thesis by summarizing the main contributions and proposing directions for future work. Annex A details the systematic literature review used to support the State of the Art analysis, Annex B contains sequence diagrams of the adopted prototype, and Annex C provides additional experimental results that complement the ones discussed in the main chapters.

Chapter 2

State of the Art

The growing adoption of Renewable Energy Sources in power systems has introduced significant challenges in data management and collaborative forecasting. Accurate modeling is essential for managing the inherent variability of these resources, as it plays a key role in ensuring reliable operation, minimizing inefficiencies, and supporting short-term planning [4].

Over the years, a wide range of forecasting techniques have been developed to meet the growing complexity of energy systems. These methods have evolved from classical statistical models to more advanced machine learning and deep learning approaches, driven by improvements in data availability, computational power, and system requirements [46].

Earlier forecasting work relied on simple univariate or multivariate time series models [28]. The integration of multisensor data and digitalization of the power grid significantly increased data volume and availability [58]. While this often led to improved model performance, it also raised concerns with accessibility and data privacy [40].

In addition to the different types of data available, another challenge to forecasting in the energy field comes from its decentralized origin. In many cases, the large volume and sensitive nature of the data make centralized collection impractical or undesirable. Moreover, data in the Energy Sector is frequently distributed across multiple entities and organizations, each operating with its own privacy policies and regulatory constraints [20]. This decentralized distribution poses significant challenges to data access, inhibiting the collaboration needed to build better models. The need to protect privacy while enabling effective data usage has become a central challenge in the development of AI systems [90]. This prompted the development of Federated Learning and data marketplaces to enable secure and equitable data sharing.

Even with mechanisms for secure collaboration, understanding the value of data remains a complex problem. Data owners often struggle to quantify the impact or usefulness of their information, which complicates decisions about sharing and compensation. This further reinforces the need for incentive mechanisms that can attribute value based on actual contributions [89].

These interconnected topics are explored throughout this chapter. Section 2.1 covers forecasting methods in the Energy Sector. Section 2.2 introduces Federated Learning and its core components. Incentive mechanisms are discussed in Section 2.3, followed by blockchain-based

approaches in Section 2.4. Section 2.5 outlines the foundations of zero-knowledge proofs. A critical analysis is presented in Section 2.6.

2.1 Forecasting in the Energy Sector

Forecasting is a core task in energy systems. It supports operational decisions, resource allocation, and market participation by anticipating the future behavior of key variables such as demand or generation [4]. In this context, forecasting typically involves analyzing time series data, which contains the evolution of system variables such as electricity production, wind speed, or solar irradiance over time.

Formally, let $\{\mathbf{y}_t\}_{t=1}^{T}$ denote a multivariate time series, where each vector $\mathbf{y}_t \in \mathbb{R}^n$ represents the state of n variables at time t. The goal of forecasting is to estimate future values $\mathbf{y}_{T+h}$ for one or more lead times (or horizons) $h > 0$, using historical observations $\{\mathbf{y}_t\}_{t=1}^{T}$.

Over time, different modeling approaches have been proposed to address the forecasting problem. Initially, traditional statistical methods such as Autoregressive Integrated Moving Average (ARIMA) were employed to model trends and seasonality. More sophisticated models like Seasonal ARIMA (SARIMA) and Holt-Winters managed to capture more complex seasonal patterns, while the Kalman Filter introduced the flexibility of state-space models [46].

With the rise of Machine Learning, methods such as linear regression, k-nearest neighbors (KNN), Support Vector Machines (SVM), Random Forest (RF), and Gradient Boosting Trees (GBT) became prominent, especially when exogenous variables were present. Deep Learning (DL) further revolutionized the field with Recurrent Neural Networks (RNNs), Long-Short-Term Memory (LSTMs), and Gated Recurrent Unit (GRUs), which are adept at capturing temporal dependencies. Meanwhile, Convolutional Neural Networks (CNNs) and related architectures enabled the detection of spatial patterns. Recent innovations such as hybrid models, attention mechanisms, and transfer learning have further improved the precision and adaptability of forecasting systems [46]. This evolution was driven by improvements in modeling techniques, but a key challenge for the sector lies in dealing with the decentralized nature of its data.

Advances in data sharing and communication technologies have enabled collaborative forecasting, where multiple agents or data holders contribute to a shared predictive model without necessarily exchanging raw data. These approaches are able to capture spatial-temporal dependencies observed across locations, thus improving forecasting accuracy. Vector Autoregressive (VAR) models are a common choice for this setting, supporting both centralized and privacy-preserving decentralized implementations [5, 19]. Privacy concerns have motivated the use of secure protocols and distributed optimization techniques, such as the Alternating Direction Method of Multipliers (ADMM), that allow agents to cooperate without exposing raw data [7, 19]. These concepts will be revisited with more detail in Chapter 3, as they form the basis of the proposed system.

To evaluate forecasting performance, predicted values are compared against held-out observations from a test set. The most commonly used error metrics are the Mean Absolute Error (MAE),

$$\text{MAE} = \frac{1}{N}\sum_{t=1}^{N}|\hat{y}_t - y_t| \tag{2.1}$$

and Root Mean Squared Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(\hat{y}_t - y_t)^2}. \tag{2.2}$$

Here, $\hat{y}_t$ denotes the predicted value at time t, while y_t is the corresponding observed value. The metrics are computed over N time steps from the test set. When comparing series with different scales, normalized metrics such as normalized MAE (nMAE), normalized RMSE (nRMSE), or Coefficient of Variation of RMSE (CVRMSE) can be used. These scale the obtained metrics by the magnitude of the observed values, making the results comparable across datasets. The Coefficient of Determination (R^2) can also be used to assess how well the predictions capture the variance in the observed data, by comparing it to the variance observed in the test set. Execution Time (ET) measures how long a model takes to produce forecasts. While not an accuracy metric, it can be useful in scenarios where inference speed matters, though it varies with hardware and implementation [46].

2.2 Federated Learning

Distributed computation of models has been a longstanding approach in AI and machine learning [7]. However, the term Federated Learning (FL) was only introduced in 2017 by McMahan et al. [45]. Unlike centralized learning, which often requires collecting data into a single location, Federated Learning enables model training to occur across multiple devices holding local data. More precisely, it enables decentralized data holders, such as smartphones, IoT devices, or organizational data silos, to participate in model training without transferring raw data to a central server.

This section begins by defining Federated Learning and describing its main components. It then presents several approaches designed to address different application requirements and use cases.

2.2.1 Definition

Federated Learning is a decentralized machine learning paradigm in which multiple entities collaboratively train a shared model without exchanging their local data [90]. These entities are commonly referred to as *clients*, *participants*, or *agents*, depending on the application domain. The coordination role is often played by a central *server* or *aggregator* [45], but decentralized

implementations are also possible, in which participants communicate directly through peer-to-peer (P2P) protocols or other distributed mechanisms [20]. More generally, participants can be distinguished as *data owners* (DOs), who provide local data and computation, and *model owners* (MOs), who initiate training requests and benefit from the resulting model, even if they do not provide data directly.

A common formulation of Federated Learning is defined in Eq. (2.3), which is based on the Federated Averaging (FedAvg) algorithm originally proposed by McMahan et al. [45] It defines the global objective as minimizing a combination of local loss functions, each weighted according to the participant's data share.

$$\min \sum_{i=1}^{N} p_i \mathscr{L}_i \tag{2.3}$$

In this expression, $\mathscr{L}_i$ represents the local loss function of participant i, evaluated on its private dataset $\mathscr{D}_i$. The coefficients $p_i \geq 0$ are weights that sum to one, and correspond to the proportion of data held by participant i, given by $|\mathscr{D}_i|$ relative to the total dataset size across all N participants, $\sum_{j=1}^{N} |\mathscr{D}_j|$. Although this objective function forms the basis for many standard approaches in FL, alternative formulations may be more appropriate depending on the specific goals of each system.

The training process begins with the initialization of a shared model, which is made available to all participants. Each participant then trains a local copy using its private data and computes model updates, which are then combined. This aggregation step, handled by either a central server or a decentralized consensus mechanism, results in an updated global model. This process is repeated over multiple communication rounds until convergence.

Unlike centralized learning, FL avoids raw data exchange entirely and allows each participant to retain full control over their dataset. This decentralization is especially valuable in settings with strict privacy regulations or data governance requirements.

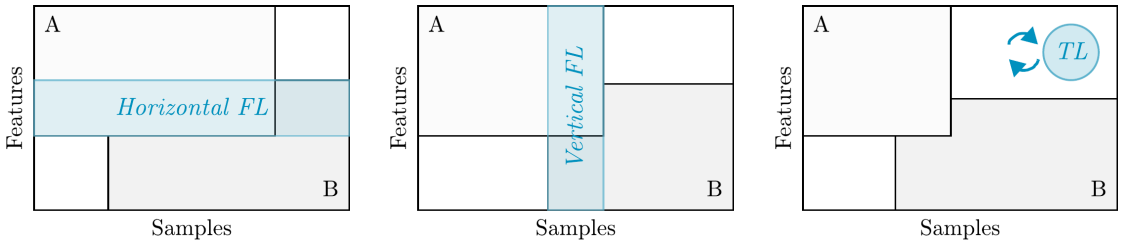
A key determinant of FL system behavior is how data is partitioned across participants. This is examined in the next section.

2.2.2 Data Partitioning

Data partitioning refers to the way in which datasets are distributed across the different nodes (or participants) in a Federated Learning system. Based on the distribution of samples and features, FL can be classified into three main types, as shown in Figure 2.1 [81, 90]:

Horizontal FL , also known as sample-based partitioning, occurs when agents have datasets with the same feature space but distinct samples. In this setup, all nodes train local instances of the same model architecture using their local data. In energy systems, for example, multiple producers might collect the same set of features (e.g., temperature, wind speed, or solar irradiance) from different locations. The local model updates are then aggregated (e.g., by averaging weights) to refine a global model [1, 2].

Vertical FL , also referred to as feature-based partitioning, occurs when participants have datasets with overlapping samples but distinct feature spaces. In this scenario, different organizations



(a) Horizontal Federated Learning (b) Vertical Federated Learning (c) Federated Transfer Learning

Figure 2.1: Federated Learning Data Partitioning schemes, according to features and samples distribution. Adapted from Zhang et al. [90].

have complementary features for the same data instances. For example, wind farms may collect wind speed, while solar plants track solar irradiance, with both recording data for the same time intervals. Vertical FL enables these organizations to collaborate and train a model together by aligning on shared samples while protecting the confidentiality of their distinct features [20, 38]. Unlike Horizontal FL, feature coefficients cannot be calculated independently by each participant, as no single party has access to all relevant features. As a result, complex secure multi-party computations are needed for gradient estimation, making Vertical FL significantly more challenging in terms of privacy, computation, and communication overhead.

Federated Transfer Learning is a more general approach to FL, where participants have datasets that differ in both their feature spaces and sample sets. This approach is useful when knowledge from one domain can still be transferred and utilized to benefit another [81].

2.2.3 Data Privacy Techniques

Data locality and model transmission are key concepts in Federated Learning. The approach by which model updates are shared significantly impacts the privacy guarantees of the system. Depending on the data partitioning and forecasting model, different systems may require stricter privacy measures, as the model updates shared with the central server may still leak sensitive information.

The literature typically categorizes methods for ensuring data privacy into the three main groups in Figure 2.2: decomposition techniques, data transformation, and secure multi-party computation. A brief description of each is provided below:

Decomposition techniques In early literature, distributed computation followed by model aggregation is a common method for ensuring privacy in FL. In this approach, participants compute local updates, typically in the form of gradients, and send them to an aggregator. The aggregator combines these updates to create a new global model, which is then shared with the participants. This process ensures that the raw data never leaves the client's devices [45, 57]. However, it is not entirely secure. Attackers can still reconstruct sensitive

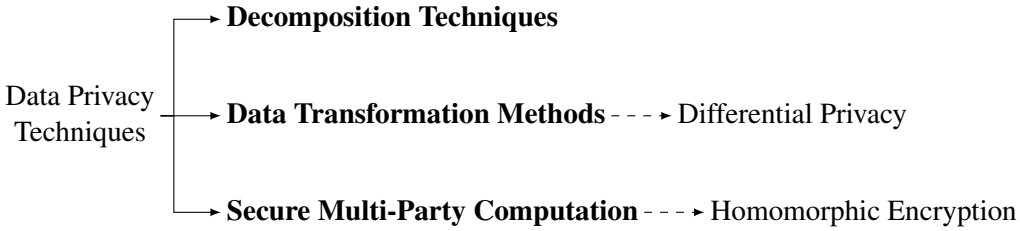


Figure 2.2: Overview of data privacy techniques relevant to FL.

information from the gradients through techniques like gradient inversion attacks, where the local gradients are used to approximate the original training data [54].

Data Transformation Methods These techniques operate by modifying raw data or updates to prevent information from being leaked to other parties. A typical approach in FL is the usage of differential privacy, which is a widely used privacy-preserving technique that limits the risk of individual data points being identified, even if an attacker has access to the output of an algorithm. It achieves this by adding random noise (usually from a Laplace distribution) to the data or to intermediate values such as gradients, making it mathematically difficult to determine whether a specific individual's data was included in the dataset. In FL, differential privacy can be applied to data or local model updates (like gradients) before sharing information with the central server, protecting user privacy even against gradient inversion attacks [33, 44]. The privacy level is controlled by the parameter ε (epsilon), where lower values of ε correspond to stronger privacy but can also degrade model accuracy, requiring a trade-off between data privacy and utility.

Secure multi-party computation Allows multiple parties to jointly compute a function over their inputs while keeping them private. A commonly used protocol is Homomorphic encryption, which allows mathematical operations (like additions and products) to be performed directly on encrypted data. In FL, each participant encrypts their model updates before sending them to the server. The server aggregates these encrypted updates and sends back an encrypted global model, which each client can decrypt [26]. This method provides strong privacy guarantees since neither the server nor any adversary can access individual updates. One setback is that it comes with significant computational overhead, as operations on encrypted data are computationally expensive [53]. Additionally, most encryption schemes rely on a third-party trust to generate the encryption keys.

While the above techniques focus on preserving data privacy during collaborative learning, they do not prevent dishonest behavior during the computation process. To address this, zero-knowledge proofs can be used to verify the correctness of local model training without revealing any private information [16]. These are discussed in detail in Section 2.5.

2.2.4 Network Topology

The original proposal for Federated Learning [45] relies on a centralized architecture where a single server coordinates the entire training process. This server facilitates communication with distributed agents and aggregates the model updates sent by them. By combining these updates, the server refines the global model, which is then shared with the agents. This process repeats iteratively, enabling collaborative training without sharing raw data. In this setting, the server typically acts as the MO, while the clients serve as DOs.

While effective in managing and coordinating distributed learning, the centralized approach has notable flaws. One key issue is the potential exposure of private information, as methods like gradient inversion allow a curious server to infer private data from the shared updates [27, 73]. Moreover, the reliance on a single server introduces a single point of failure, as system outages or breaches can disrupt the entire process. Centralized systems are also more vulnerable to security attacks, such as data poisoning, where malicious participants can manipulate their local training data to mislead the global model during aggregation [66].

While these concerns are valid, using a central aggregator remains a practical and efficient way to coordinate Federated Learning in many settings. A trusted third party, such as a company or a cloud service provider, can act as the aggregator. However, this approach is not always possible or desirable, especially in privacy-sensitive applications, where data owners may be reluctant to trust an external entity with access to their model updates.

Decentralized Federated Learning addresses these limitations by removing the central coordinator and allowing participants to communicate directly with each other in a peer-to-peer (P2P) fashion. This setup achieves model aggregation without relying on a central entity, thus improving privacy. While coordination becomes more complex in this setting, it offers greater resilience against single points of failure and information leakage. In decentralized settings, there can be multiple MOs and DOs, and participants may take on either role or both, depending on the scenario.

Different methods can be used for model aggregation between participants. In Blockchain-based aggregation, agents send their local updates to the blockchain, and the model is created in the ledger. These updates can be previously encrypted using Differential Privacy or Homomorphic Encryption to preserve privacy [11, 95]. In Gossip protocols, agents share their updates with neighboring peers, sequentially propagating the information through the network, which gradually leads to model convergence [25, 97]. In chain-like or ring structures, each agent sequentially transmits its model updates to the next agent in a fixed sequence. The process continues until the final agent aggregates all updates to obtain the final model [20].

2.2.5 Real Word Applications

Similarly to traditional machine learning techniques, Federated Learning has been applied across a broad range of domains. It is particularly promising in scenarios where data is valuable but cannot be centralized, either due to privacy constraints, institutional boundaries, or limitations in data

transferring capabilities. This section highlights several key applications of FL, with a specific focus on its relevance to the Energy Sector, particularly in RES forecasting, which is the main application domain of this thesis.

One of the first and most well-known applications of Federated Learning is Google's use to enhance the Gboard virtual keyboard. The primary objective was to improve next-word prediction. The text users type on their keyboards often contains sensitive information. Thus, Google used FL to train the prediction model directly on users' devices, resulting in a model that could outperform one trained on a central server [23]. Since then, FL has found its application in different domains, such as Healthcare [41, 50, 60], Internet of Things [49, 80], Internet of Vehicles [61, 70], and Industry [22, 35].

The Energy Sector is a particularly well-suited use case for FL, given the sensitive nature of the data. The increasing digitalization of grid-edge devices and the consequent increase in data volume have paved the way for FL to be applied in forecasting, non-intrusive load monitoring, and predictive maintenance, among others [92]. Briggs et al. proposed a federated learning approach for short-term load forecasting that achieved a $\sim$5% performance improvement and a $\sim$10x reduction in computational cost compared to localized learning [9]. Furthermore, Ahmadi et al. explored a deep federated learning-based framework for wind power forecasting, demonstrating 7.42% higher accuracy than models fed with local data, while preserving data privacy across multiple wind farms [1].

2.2.6 Challenges and Limitations

Since Federated Learning is a relatively recent concept, it faces several unresolved challenges. This section briefly describes some of the problems currently associated with this method.

One of the most significant issues is the communication overhead inherent in its distributed nature. FL requires frequent exchanges of model updates between participants and a central server, which can result in high bandwidth consumption. Since models are trained locally and only parameter updates are shared, a large number of participants and frequent communication rounds create a bottleneck. Methods to reduce communication include model compression, quantization, and techniques that limit the frequency of updates, but these approaches introduce trade-offs between communication efficiency and model accuracy [90].

Privacy, although often highlighted as a key advantage of FL, also presents significant challenges. While raw data never leaves local devices, the gradients shared with the server during training can still reveal private information to a curious server [54]. As previously discussed in Section 2.2.3, various methods have been proposed to address this issue, including homomorphic encryption for secure aggregation and differential privacy. It is important to note that these techniques often introduce significant computational overhead or degrade model accuracy, highlighting the need to balance performance with privacy guarantees [86].

Another major concern is the presence of malicious participants in the learning process, especially in open FL environments, as it is difficult to guarantee that all participants act honestly.

Malicious actors can introduce "poisoned" data or send inaccurate model updates to the aggregator, degrading the quality of the final model. Methods to counteract malicious participants include Byzantine-resilient aggregation techniques, which aim to identify and exclude faulty updates, and anomaly detection methods that flag unusual behavior. Additionally, some systems use blockchain technology to maintain a transparent record of participant behavior, allowing reputation-based exclusion of unreliable participants [31].

Finally, one of the most persistent challenges in FL is participant motivation. FL's structure poses the question of why participants would willingly offer their computing resources, bandwidth, and learned knowledge without receiving clear benefits. This issue is particularly relevant in commercial and competitive environments where companies may be unwilling to share information with their competitors [89]. Since this is the main concern of this thesis, it will be covered separately in subsequent Section 2.3.

2.3 Incentive Mechanisms

Although privacy is often the main concern in Federated Learning, it is not always enough to motivate and sustain collaboration. In practice, the benefits of a shared model are often unevenly distributed across participants. For some situations, participating in the collaboration may lead to significant improvements to the models of other agents, yet offer minimal benefits in return. In more extreme cases, this leads to what is known as a free-rider attack, where a participant contributes little to the global model but still gains access to the same high-quality predictions as other contributors. This asymmetry may reduce the appeal of collaborating, particularly in competitive settings. Incentive mechanisms contribute to balancing individual and collective gains, with monetization providing a practical approach to motivate participation.

The importance of incentives applies to both cross-device and cross-silo FL scenarios. In cross-device FL, participants are individual users or edge devices that contribute with locally trained model updates derived from private data. These are a valuable asset with inherent monetary value. In some cases, the value of data is not known to its owner, and incentives can help uncover it [21]. For these clients, participation also comes with extra costs in computation, energy consumption, and bandwidth usage, all of which discourage voluntary collaboration [89]. Incentives also provide a way to agree on the value of resources, helping clients decide how much they are willing to contribute in exchange for a given reward [67]. On the other hand, cross-silo FL brings together large organizations with substantial datasets that could all benefit from a shared collaborative model. However, since these organizations are often competitors operating in the same market, they can be reluctant to participate, even if privacy guarantees are in place. This setting is especially vulnerable to free-riding, making participants less willing to take part [62].

These limitations show why incentive mechanisms are crucial for enabling collaboration in practical FL systems. However, designing them effectively is not a straightforward task. One major challenge lies in how to evaluate each agent's contribution to the global model while also ensuring that participants behave honestly. This becomes particularly difficult given that the agent's data

is private and thus cannot be directly accessed. Many incentive mechanisms have been proposed to address this problem, with a focus on how to evaluate individual contributions in Federated Learning. Since there is no established or widely accepted taxonomy in the literature [47, 88, 89], the grouping shown in Figure 2.3 was adopted to support the discussion. Each category will be described in detail throughout the section.

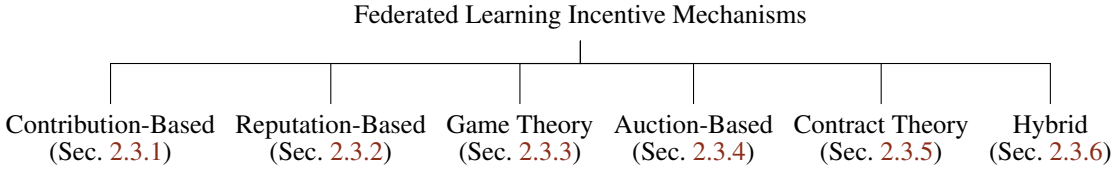


Figure 2.3: Main categories of FL incentive mechanisms discussed in this section.

2.3.1 Contribution-Based

Contribution-based incentive mechanisms aim to reward participants in proportion to their individual impact on the global model. The central challenge of this approach lies in accurately evaluating the contribution of each participant. Once contributions are estimated, compensation can be distributed to encourage participation and promote fairness. In addition, this must be achieved without violating the system's core constraints; namely, ensuring privacy, maintaining trust between participants, and keeping the overall process reliable and efficient [89].

One common approach within this type of mechanism is the use of the Shapley value, a concept from cooperative game theory that aims to reward each participant based on their marginal contribution across all possible subsets of collaborators. It has been widely applied in Federated Learning to evaluate the relative importance of individual clients in training the global model [71, 84]. In this context, the Shapley value estimates a client's contribution by comparing model performance with and without their participation across different subsets of participants. More specifically, the Shapley value computes the average marginal gain a client brings when added to all possible combinations of others. Formally, let N be the set of all clients, and let $v(S)$ denote the utility of a subset $S \subseteq N$ (e.g., model accuracy). For a given client i, its contribution is estimated by averaging the marginal differences $v(S \cup \{i\}) - v(S)$ over all subsets $S \subseteq N \setminus \{i\}$,

$$\phi_i = \frac{1}{|N|} \sum_{S \subseteq N \setminus \{i\}} \left[v(S \cup \{i\}) - v(S) \right]. \tag{2.4}$$

This formula captures how much client i contributes on average when participating in different coalitions. In practice, computing the exact Shapley value is computationally expensive due to the exponential number of possible combinations. Therefore, most systems rely on approximation methods using a small sample of subsets.

While Shapley value considers the quality of data, other methods rely on different metrics to estimate contributions. Among these, the quantity of data samples is sometimes used as an approximation of the client's contributions, particularly in horizontal Federated Learning, where the

amount of data is related to model performance [24, 82, 89]. Additionally, some systems integrate resource-awareness into their evaluation, rewarding clients not only for what they contribute but also for the efficiency with which they do so. This includes factors such as computational capacity, energy consumption, and responsiveness over time [24, 30, 55]. Another common alternative involves simplified approximations resembling the Shapley value, where a client's contribution is evaluated based on the isolated effect their updates have on the global model. For example, Hassija et al. [24] estimate the contribution by measuring the change in model accuracy when a single participant's update is applied. In contrast, Zhang et al. [91], focusing on model owners, evaluate the quality of model updates based on the similarity to the updates of other participants, which indirectly reflects their alignment with the global objective. It is important to note that, in practice, most systems combine several of these metrics to better capture the complexity of client contributions [24, 84, 89, 91].

One limitation of contribution-based evaluation methods is their potential conflict with privacy-preserving techniques such as those discussed in Section 2.2.3. Introducing noise with mechanisms like differential privacy reduces the quality of client updates for model training [93], which consequently makes it harder to accurately assess each participant's contribution [84]. This highlights an inherent trade-off between preserving privacy and providing reliable contribution evaluation.

In summary, contribution-based mechanisms provide a basis for many incentive schemes, though their design involves trade-offs between accuracy, computational cost, fairness, and privacy constraints.

2.3.2 Reputation-Based

In reputation-based incentive mechanisms, the system tracks the historical behavior of each agent and uses it to inform reward allocation. The central component is the computation of a reputation score, which serves as the basis for calculating the agent's compensation. Most reputation-based systems determine the reputation score R using a common update structure [32, 96],

$$R = \varepsilon R_d + (1-\varepsilon)R_h, \tag{2.5}$$

where R_h represents the agent's historical or indirect reputation, typically obtained from a previous computation of R, allowing the system to retain memory of past behavior. The term R_d denotes the direct reputation, computed from the agent's most recent contribution. Although this structure is widely adopted, the main point of divergence across systems lies in how R_d is calculated. The parameter $\varepsilon \in [0,1]$ controls the balance between recent and historical behavior, effectively acting as a decay factor.

An established approach to computing R_d adopts the same types of metrics used in contribution-based incentives (Section 2.3.1). Several systems estimate direct reputation based on the marginal improvement the updates of each client bring to the global model performance [51, 55, 96]. Other

designs rely on simpler indicators, such as the amount of data held by each client [82], or incorporate resource metrics like computational cost and energy consumption [55, 75]. A few systems also make use of Shapley-based estimators [32, 75]. In practice, these evaluation strategies are often combined to improve the robustness of the estimations.

An alternative approach is to arrive at R_d based on peer evaluation, where the reputation of each client is derived from the opinions of other participants. In these designs, agents evaluate peer updates using various criteria, such as performance impact or similarity to expected behavior. Some systems use subjective logic to combine direct observations with third-party recommendations into a unified reputation score [30]. Others compute reputation from weighted assessments, often assigning higher credibility to nodes with a history of reliable evaluations [75, 82].

Reputation scores can also be used for participant selection, in parallel with incentives, by determining which clients are eligible to join future training rounds or should be excluded. Some systems integrate this with the reward mechanism, using reputation both to allocate compensation and to enforce participation rules [30, 51, 75]. A common enforcement rule is to ban agents whose reputation drops below a minimum threshold [55], promoting honest behavior and mitigating disruption from malicious nodes.

Blockchain and reputation mechanisms are a common pairing in Federated Learning, which work well by complementing each other [32, 82, 83]. Blockchain can be used to publicly store reputation information, making past behavior verifiable and persistent. This transparency allows participants to audit how reputation scores are computed, promoting trust in the system and encouraging honest participation.

2.3.3 Game Theory

Game theory provides a way to model strategic interactions between participants, called players, who choose actions to maximize their own benefit. Each player has a set of strategies and a utility function, and the outcome of the game depends on the combination of their choices. This framework is often used in FL to design incentive mechanisms, especially in settings where roles are split between model owners and data providers [67].

Some works assume cooperation between players, where participants coordinate and share rewards. These approaches often rely on fairness concepts rather than strategic interaction. For example, Shapley values (discussed in Section 2.3.1) are used to compute rewards based on contribution.

In most cases, however, players are self-interested and act independently. Non-cooperative game theory is used to model this kind of setting, where participants negotiate, compete, or respond to incentives without coordination [67].

Non-cooperative game theory models situations where players act independently to maximize their own utility. Each player chooses a strategy based on the expected actions of others, and the system reaches a stable outcome when no one can improve their situation by changing strategy alone. This is known as a Nash equilibrium, and most non-cooperative incentive mechanisms are designed so that the system naturally converges to one [67].

A common example used in practice is the Stackelberg game, which introduces a sequential structure to decision-making. One player, the leader, moves first, and the remaining players, the followers, respond after observing the leader's choice. The leader anticipates how followers will react and chooses its action accordingly. In FL, this often maps to the model owner setting a reward or pricing scheme, while the clients respond by deciding how much data or effort to contribute [29, 82, 85].

In short, game theory in FL provides a way to structure negotiations over incentive pricing, contribution level, effort, or resource usage, even when participants act selfishly. As an illustration, Liu et al. use a game where clients charge the model owner based on the estimated energy cost of local training and communication [39].

2.3.4 Auction-Based

An auction is a process where buyers and sellers compete for resources or services by submitting bids under a defined set of rules. There are many ways an auction can be structured, and these can be categorized using four main criteria:

Direction Defines who is making offers and who is receiving them, depending on whether participants are bidding to get a resource or to provide one.

Cardinality Describes how many participants are on each side of the bidding, such as one buyer with many sellers, or many buyers and many sellers.

Bidding format Explains how bids are submitted, either privately without knowing others' bids, publicly where all bids are visible, or over multiple rounds.

Pricing rule Determines how the payment is set for the winners, such as paying your own bid, the second highest, or accounting for the impact on others.

Within the context of Federated Learning, auctions can be used both to select participants and negotiate how to share the incentives received. This mechanism is particularly useful in FL when MOs and DOs are separate entities, as it allows them to agree on payments, participation terms, and how much each side is willing to commit in terms of resources. Auction mechanisms in FL differ according to roles played by participants, how incentives are negotiated, and what the system is optimizing for.

Forward auctions are used when multiple MOs compete to obtain services from DOs, who act as sellers. Each buyer submits a bid to access data or training resources, and the allocation is determined by comparing these bids. This structure fits scenarios where data access is costly or scarce [63].

A recurrent type of auction in the FL literature is the reverse auction. In this setup, the FL server acts as the buyer and selects among multiple clients, who act as sellers and submit asks to offer local training. This structure is effective because the server can select clients based on

their price, and clients are compensated according to how much they ask for. In sealed-bid formats, clients submit their bids privately and are paid just enough to stay selected over the next-best alternative, which encourages truthful bidding while minimizing cost [14, 52]. This payment corresponds to the client's critical value, the highest price they could have asked while still being selected. Other mechanisms use online bidding, where bids are submitted over time across multiple rounds, allowing the server to adapt selection as the process evolves [87]. Vickrey–Clarke–Groves (VCG) pricing works by measuring the benefit of each client to the system and paying them based on that contribution, which helps ensure honest bidding and leads to better overall outcomes [77].

Double auctions allow both sides of the market to submit bids simultaneously. This setup is well-suited for decentralized or peer-to-peer FL markets, where multiple MOs and DOs interact [67].

Combinatorial auctions are used when participants bid for bundles of resources instead of single items. This is useful in FL when participation requires allocating multiple resources at once, such as bandwidth, energy, and training time. In these settings, clients bid for combinations of resources that meet their own goals [65].

All the auction variations discussed are summarized in Table 2.1, organized according to the criteria introduced earlier.

Table 2.1: Common auction types categorized by key criteria and representative examples.

Variable	**Examples**
Direction	Forward [63], Reverse [52, 87, 14, 42, 51, 77], Double [67]
Cardinality	Single-buyer / multiple-sellers [52], Multiple-buyers / single-seller [63], Many-to-many [67]
Bidding format	Sealed-bid [52, 14], Iterative / Online [87, 77]
Pricing rule	First-price [63], Second-price [63], Critical value [52, 14], VCG [77]
Resource structure	Single-item [52, 77], Combinatorial [65]

2.3.5 Contract Theory

Contract theory is an area of economics that studies how to design agreements between parties with potentially conflicting interests and unequal access to information. It is particularly focused on solving situations involving asymmetric information, where one party holds private knowledge that the other cannot directly observe or verify. The typical setup involves two separate roles, with the *principal* defining a set of contract options and the *agent* choosing one to accept based on their private characteristics. The principal designs the contracts to appeal to different possible agent types, without knowing in advance which type each agent is. Then, each agent selects the contract that best matches their needs. This allows agents to act in their own interest while preserving their private information, effectively resolving the asymmetry [67].

The use of contract theory in Federated Learning was first introduced by Kang et al. [30]. Since participants hold private information, contract theory provides a way to design incentive

mechanisms that work without requiring them to reveal it. Typically, the MO plays the role of the principal and issues contracts to a set of DOs, who act as agents. This setting naturally models common asymmetries in FL, such as participants' computational resources, data quality, local latency constraints, willingness to participate, or desired compensation.

Contracts used in incentive mechanisms can be either static or dynamic. In static contracts, the principal offers a fixed set of choices that the agent selects on a single interaction [36, 59]. Dynamic contracts extend over multiple rounds, allowing both sides to adapt as contracts evolve [10, 69].

Implementations differ in terms of what type of asymmetry is being handled. Some works consider data-related asymmetries, such as local quality or volume [36, 59], while others focus on resource constraints like energy consumption, latency, or computation delay [37, 70].

2.3.6 Hybrid Strategies

From a practical perspective, the ideas presented above can be combined to design more effective incentive mechanisms. In fact, many of the systems discussed earlier adopt hybrid strategies that integrate multiple approaches in order to address different needs of incentive design. There are many possible ways in which to combine the mechanisms. The most recurrent examples are illustrated below.

Reputation scores are sometimes used in support of other incentive mechanisms. They can be integrated into auction processes, where reputation influences either bidding eligibility or winner selection [56, 51]. They also complement contribution-based systems by using past behavior or score history to assess participant quality [32, 82]. In contract theory, reputation is used to differentiate between reliable and unreliable agents and adjust rewards accordingly [30]. Finally, game-theoretic approaches incorporate reputation into decision-making and strategy formation [75].

Besides supporting reputation systems [32, 82], contribution-based scores can also complement other mechanisms by providing data-driven measures of participant value. In auction schemes, they influence bid evaluation by reflecting the expected utility of a participant's contribution [14]. Game-theoretic approaches use them as input for utility or payoff functions [29, 39]. At last, in contract theory, contribution metrics are integrated into the value functions of contracts [68, 10].

2.4 Blockchain for Federated Learning

Blockchain is a distributed ledger system that securely records transactions without relying on any centralized authority. At its core, it provides a transparent way to store data across a network of independent participants, where previously recorded information cannot be changed without collective agreement. Transactions are grouped into blocks, which are linked together to form a chronological chain shared by all participants. Once data is recorded in a block and added to the chain, it cannot be altered without consensus from the majority of the network, as each block depends on the previous one. Initially introduced as the foundation for Bitcoin's monetary system [48], blockchain has since evolved beyond cryptocurrencies. Its decentralized nature has

opened up numerous opportunities across various applications. Instead of relying on a central server or authority, blockchain networks operate through distributed consensus, where participants collectively verify and agree on the state of the ledger. By removing the dependence on central servers, blockchain effectively addresses critical issues such as single points of failure and the need for trust in third-party intermediaries, which is useful for many different areas [94].

To extend the functionality of blockchain systems beyond simple data recording, smart contracts were introduced as a computational layer on top of the ledger. Beyond the basic storage of transaction records, smart contracts allow for the execution of logic directly on the blockchain. Operating as self-executing scripts, these contracts run on a decentralized virtual machine, enabling automated computation and enforcement of predefined rules. This change allows blockchain systems to not only store data but also process and react to it through embedded logic. With this technology, it becomes possible to create fully decentralized applications that can execute complex logic by relying on the blockchain [76].

Some of the specific properties of Blockchain can help address problems in Federated Learning. Its decentralized structure removes the need for a central coordinator, and blockchain-based mechanisms can be used to validate contributions and the manage shared state. The following sections discuss specific aspects of integrating blockchain into Federated Learning, focusing on relevant challenges and design choices.

2.4.1 Aggregation

Most Federated Learning systems rely on a central server to perform model aggregation. In each round, participants send their local model updates to this server, which computes the global model by aggregating the received weights. However, this centralized architecture introduces a number of limitations. Since aggregation happens inside the server, participants have no visibility into how the model is updated, which raises concerns about trust and transparency. In addition, the server can become a single point of failure and an attractive target for adversarial behavior, such as data poisoning or the extraction of private information from model updates [15].

An alternative approach is to perform the aggregation process directly on the blockchain. In this setting, participants submit their model updates to the blockchain, where aggregation is handled through smart contracts. When this approach is executed in a public ledger, the aggregation process becomes fully transparent, since the contents of the ledger are visible to all participants.

Because the blockchain is public, this architecture must be combined with privacy-preserving techniques. As discussed in Section 2.2.3, agents can locally apply methods such as homomorphic encryption or differential privacy to their model updates before sending them to the blockchain. This ensures that, although the aggregation is performed in a transparent and verifiable way, the underlying data remains private and secure. Only the agents hold the necessary keys to interpret their own updates [11, 15].

2.4.2 Incentive Mechanisms

Another application of blockchain in Federated Learning lies in supporting incentive mechanisms. A natural first step for this integration is to use blockchain to handle payments, recording transactions directly on a distributed ledger. The transparency of blockchain allows all participants to see the recorded transactions and verify how rewards are being distributed [79].

Beyond handling payments, blockchain technology can be integrated to support the incentive mechanisms discussed throughout Section 2.3. One such example is the use of blockchain to manage reputation scores. By recording model quality scores and evaluation outcomes on the public ledger, the system can store the information needed to track the historical behavior of each participant. Smart contracts can then use this information to compute reputation values over time [30, 56, 82]. Blockchain has also been used to support game-theoretic approaches, for example, by implementing Stackelberg games through smart contracts [72, 74]. In addition, smart contracts have been used to implement contract-theoretic incentive mechanisms, allowing agreements between participants to be enforced automatically [30, 72]. Finally, blockchain has been used to manage auction-based incentive mechanisms, where smart contracts execute the bidding process and determine outcomes automatically [51, 14].

2.5 Zero-Knowledge Proofs

Incentive mechanisms introduce the risk of dishonest behavior, as agents may attempt to manipulate their contributions in order to receive higher rewards. To address this, it is important to verify that reported values are correct without compromising data privacy. This section introduces Zero-Knowledge Proofs (ZKPs), a class of cryptographic techniques that enable such verification without revealing sensitive information.

Zero-knowledge proofs are cryptographic protocols that allow one party, known as the *prover*, to prove to another party, the *verifier*, that a statement is true, without revealing any information beyond the validity of the statement itself. ZKPs were first introduced by Goldwasser et al. [18] who initially formalized this proof system.

In a typical setting, the prover holds some secret information, referred to as the *witness*, and wants to convince the verifier that they know a valid solution to a given problem. The verifier checks the proof and accepts it if it is convinced that the prover knows a solution, without learning anything about the witness.

Formally, ZKPs are defined over a language $L \subseteq \{0,1\}^*$, where each $x \in L$ represents a valid statement. The prover and verifier engage in an interactive protocol in which the prover attempts to convince the verifier that $x \in L$.

A zero-knowledge proof system must satisfy three important properties:

1. **Completeness:** If the statement is true ($x \in L$) and both parties follow the protocol honestly, the verifier accepts the proof with high probability.

2. **Soundness:** If the statement is false ($x \notin L$), no (even malicious) prover can convince the verifier to accept the proof, except with negligible probability.

3. **Zero-knowledge:** If the statement is true, the verifier learns nothing beyond the fact that the statement is true.

To better illustrate the notion of ZKPs, consider the following example, based on the graph isomorphism problem [17]. Two graphs G_0 and G_1 are isomorphic if they have the same structure, meaning there is a one-to-one correspondence between their vertices that preserves all edge connections. Such is illustrated in Figure 2.4, where isomorphism can be verified by replacing the labels *a* through *d* by the numbers *1* to *4*. Formally, this means there exists a mapping f such that $f(G_0) = G_1$.

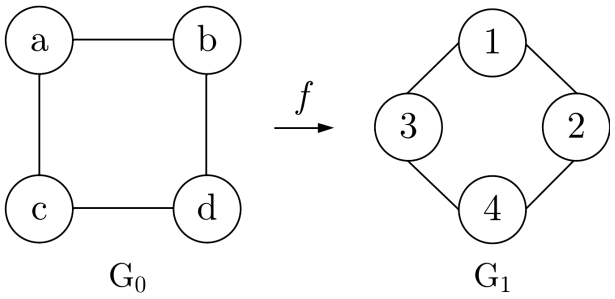


Figure 2.4: Two isomorphic graphs that differ only in the labeling of their vertices.

Suppose the prover wants to prove to the verifier that they know the function f, without revealing any information about it. In this scenario, G_0 and G_1 are both public, but the transformation between them is unknown to the verifier.

The protocol proceeds as follows (repeated m times):

1. The prover chooses a random permutation π of the vertices and applies it to G_1, generating a new graph $H = \pi(G_1)$, as seen in Figure 2.5. The new graph H is isomorphic to G_1, but will only be isomorphic to G_0 if the prover knows f.

2. The prover sends the graph H to the verifier.

3. The verifier chooses a random challenge bit $\alpha \in \{0,1\}$ and sends it to the prover.

4. To prevent the prover from preparing a single fixed response, the verifier sends a random challenge bit $\alpha \in \{0,1\}$, forcing the prover to demonstrate knowledge in two distinct ways:

   - If $\alpha = 0$, the prover reveals the composition $\pi \circ f$, which maps G_0 to H, thereby proving that H is isomorphic to G_0 without revealing any information about f.
   - If $\alpha = 1$, the prover reveals π, which directly maps G_1 to H, proving that H is isomorphic to G_1.

5. The verifier checks that the revealed mapping correctly transforms G_α into H, increasing the probability that the prover knows f. A dishonest prover could have constructed H to be isomorphic to either G_0 or G_1, but not both, so their chance of guessing the correct challenge is at most $1/2$ in each round.

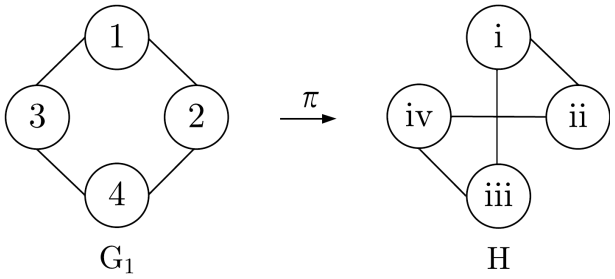


Figure 2.5: Step 1 of the protocol. The prover applies a random permutation π to the graph G_1, producing a new graph $H = \pi(G_1)$. The graph H is isomorphic to G_1, with the nodes relabeled from numbers to Roman numerals.

Repeating this process m times makes it increasingly unlikely that a prover who does not know f can consistently respond to the verifier's challenges and succeed in convincing them.

While this protocol demonstrates how interaction enables zero-knowledge proving, it also shows that repeated communication between the prover and verifier can be limiting. To address this problem, Blum et al. [6] later showed that interaction could be removed entirely by relying on a shared random string, leading to the first construction of non-interactive zero-knowledge proofs. This advancement made adopting this mechanism significantly more practical, making it useful in real-world applications where interaction is undesirable.

More recently, significant progress has been made in improving the efficiency and scalability of non-interactive zero-knowledge proofs. These developments led to the introduction of zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge), a class of cryptographic proofs that retain the zero-knowledge and non-interactivity properties while being extremely compact and efficiently verifiable. In zk-SNARKs, the computation to be verified is first represented as an *arithmetic circuit*, which is a directed acyclic graph composed of addition and multiplication gates. The circuit takes a set of inputs, processes them through a sequence of arithmetic operations, and produces one or more outputs. A simple example is shown in Figure 2.6. The relationships between inputs, intermediate values, and outputs are encoded as a system of arithmetic constraints, which is then transformed into a polynomial representation suitable for cryptographic proving. This enables efficient communication, making zk-SNARKs particularly useful in contexts such as distributed systems and blockchains, where there are constraints on both computation and communication [3].

ZKPs have been applied to a wide range of practical domains. These include proof of identity systems in which users are able to prove membership or certain attributes without revealing sensitive information. In machine learning, ZKPs enable the verification of inference results while keeping both the input and model parameters private. In blockchain systems, they are fundamental

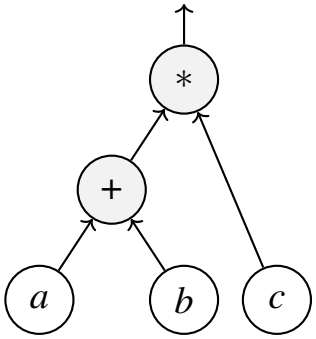


Figure 2.6: Example of an arithmetic circuit for computing $(a+b)\cdot c$. Each circle represents a gate or input node, with directed edges indicating the flow of computation.

to privacy-preserving cryptocurrencies like Zcash, which supports confidential transactions without revealing the sender, receiver, or amount [34]. In the context of federated learning, ZKPs and zk-SNARKs have been used to verify the correctness of local model computation and aggregation [16, 78]. However, the application of zero-knowledge proofs in incentivized federated learning, where participants are rewarded based on their contributions, remains relatively unexplored in the current literature [43]. This represents a promising area for further research, particularly for ensuring that rewards are allocated fairly without compromising user privacy.

Many of these applications, particularly those involving federated learning and blockchain, require mechanisms to ensure that secret values are fixed and verifiable without being revealed. This is often achieved through the use of *commitment schemes*, which are used within ZKPs to allow provers to bind themselves to hidden inputs while preserving privacy and integrity. Commitment schemes are briefly introduced below, as they form a core building block in many zero-knowledge constructions.

2.5.1 Commitment Schemes

A commitment scheme allows a prover to commit to a secret input value x, denoted $\text{Commit}(x)$, while keeping x hidden. The process is composed of two separate commit and reveal stages, and should be both hiding and binding. This means that x remains secret, and that the prover cannot change their mind about x once the commitment is made.

As a practical example, consider a guessing game where the prover wants to challenge the verifier to guess a number, either 0 or 1. The prover first chooses a number and locks it in a sealed box, analogous to computing $\text{Commit}(x)$. After the verifier tries to guess the number, the prover opens the box and reveals the original number. This allows the verifier to check whether the guess was correct, while ensuring that the prover couldn't have changed their chosen number. In a concrete implementation, this box corresponds to a cryptographic one-way function, such as a hash function, that is easy to compute but hard to invert with any meaningful probability.

Commitment schemes are a fundamental building block in the construction of zero-knowledge proofs [8, 17]. Their properties allow the verifier to check that the prover behaves consistently with

values that were previously committed from the private witness. Moreover, commitment schemes can also be used to extend a proof in scenarios where it must be reused over time, ensuring that the same private values remain fixed across different stages.

2.6 Critical Analysis

Federated Learning is an emerging field that enables the use of valuable but previously inaccessible data due to factors such as privacy or confidentiality constraints. By allowing collaborative model training without requiring raw data to be shared, FL addresses key barriers to effective data sharing. However, despite the promising concept, its practical implementation demands context-specific adaptations and trade-offs. As discussed in previous sections, real-world applications must handle challenges such as competitive data ownership and heterogeneous data sources. These conditions require tailoring the FL framework to address domain-specific constraints on privacy, security, and incentive compatibility.

Forecasting plays a critical role in the renewable energy sector, yet data owners are often unwilling to engage in collaborative model training due to concerns over data privacy. This reluctance is driven by the sensitivity of the data and the consequent competitive implications of sharing it. To address these limitations, Gonçalves et al. [20] proposed a privacy-preserving framework for distributed time series forecasting that allows participants to collaboratively train VAR models without exposing their raw data. A key challenge in this setting is that multivariate time series models, such as VAR, require agents to share covariates derived from their own power measurements, thus compromising confidentiality. To overcome this, each data owner applies private row-wise and feature-wise transformation matrices to their local time series before sharing. This masking step preserves the structure necessary for model training while ensuring that raw data and intermediate parameters remain hidden from both other participants and the central coordinator. As a result, the framework enables accurate collaborative forecasting in settings where privacy concerns would otherwise prevent data sharing. Given that each participant contributes different features for a shared prediction task, the framework can be classified as an instance of vertical Federated Learning.

However, the proposed framework assumes that participants are inherently motivated to collaborate in the forecasting task. In practice, this assumption may not hold, particularly in competitive or asymmetric environments where some data owners benefit less from the collaborative model than others. Without explicit incentives, participants may still choose to withhold data, even when privacy is preserved. This suggests that the framework would benefit from the integration of an incentive mechanism to promote sustained and fair participation across diverse agents.

This gap is addressed by extending the original framework with an incentive mechanism designed to ensure participation even in scenarios where individual benefits from the global model are limited or unevenly distributed.

Chapter 3

Methodology

In response to the increasing demand for transparent and collaborative renewable energy forecasting, this chapter presents a novel system that extends an existing privacy-preserving Federated Learning framework by introducing a monetary incentive mechanism.

The chapter begins with a review of the VAR model in Section 3.1. This model is the basis for the privacy-preserving system proposed by Gonçalves et al. [20], presented in Section 3.2, that serves as the foundation for this work. Section 3.3 outlines the gaps in the existing framework that motivated the proposed extensions, namely the incentive mechanism in Section 3.4 and validation algorithms in Section 3.5.

3.1 VAR Model

The VAR model is a statistical model used to capture the relationship between multiple time series that influence each other over time. Unlike simpler models that treat each time series separately, VAR models allow all variables in the system to depend linearly not only on their own past values but also on the past values of the other variables in the system.

More formally, let $\{\mathbf{y}_t\}_{t=1}^{T}$ be an n-dimensional multivariate time series, where n represents the number of agents or data owners. This series follows a vector autoregressive process of order p, denoted as $\text{VAR}_n(p)$, when:

$$\mathbf{y}_t = \eta + \sum_{\ell=1}^{p} \mathbf{y}_{t-\ell}\mathbf{B}^{(\ell)} + \varepsilon_t, \tag{3.1}$$

for each time point $t \in \{1,\ldots,T\}$.

In this formulation, $\eta = [\eta_1,\ldots,\eta_n]$ is the intercept vector (a row vector) in $\mathbb{R}^n$, $\mathbf{B}^{(\ell)} \in \mathbb{R}^{n\times n}$ is the coefficient matrix at lag ℓ, where the entry $B_{i\to j}^{(\ell)}$ captures the effect of the j-th series on the i-th series at lag ℓ. The noise term $\varepsilon_t = [\varepsilon_{1,t},\ldots,\varepsilon_{n,t}]$ is an n-dimensional white noise vector, assumed to be independently and identically distributed with zero mean and a nonsingular covariance matrix.

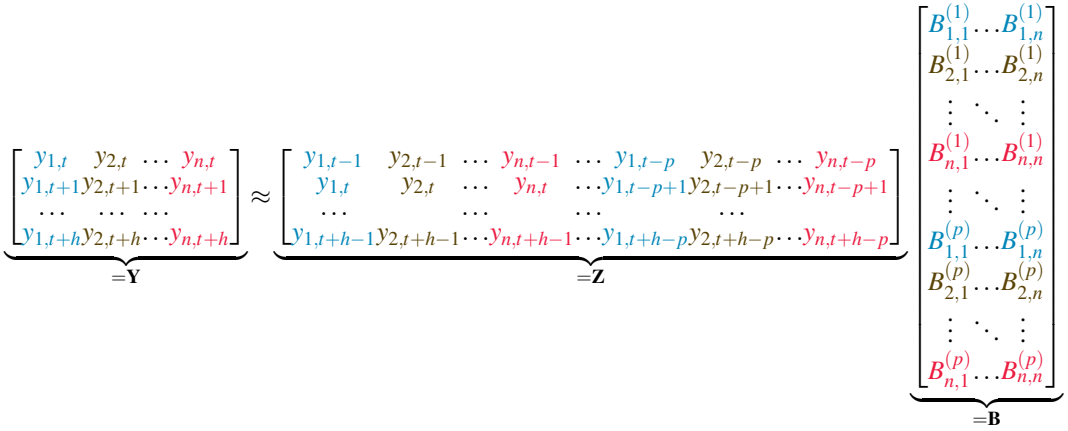


Figure 3.1: Visual layout of the $\text{VAR}_n(p)$ model structure, highlighting lagged dependencies and agent-specific data partitions using distinct colors.

For simplicity, the process is assumed to be centered, so $\eta = \mathbf{0}$. Under this setup, the $\text{VAR}_n(p)$ model can be compactly written in matrix form, as illustrated in Figure 3.1:

$$\mathbf{Y} = \mathbf{ZB} + \mathbf{E}, \tag{3.2}$$

where $\mathbf{Y}$ is the $T \times n$ target matrix, $\mathbf{B}$ is the $np \times n$ matrix of coefficients, $\mathbf{Z}$ is the $T \times np$ covariate matrix composed of lagged observations, and $\mathbf{E}$ is the $T \times n$ matrix of residual errors.

Figure 3.1 also illustrates how the VAR process operates across multiple lags, and shows that each column of the data matrices $\mathbf{Y}$ and $\mathbf{Z}$ is owned by a unique data provider. Correspondingly, each row block in the coefficient matrix $\mathbf{B}$ interacts only with a specific subset of features. This partitioning suggests a natural alignment with distributed data ownership, making the VAR model particularly well-suited for collaborative estimation in privacy-preserving environments.

Although the standard VAR model is linear and only models endogenous, or internal, variables, it can be extended to handle nonlinear relationships through preprocessing and by incorporating exogenous, or external, inputs, resulting in a VAR-X formulation. For example, the relationship between wind speed forecasts and future wind power output is nonlinear. To capture this within a VAR-X framework, wind speeds from multiple locations can be first transformed using cubic splines. These transformed series then serve as exogenous inputs, allowing the VAR-X to account for spatial and nonlinear effects while retaining its linear structure in the endogenous dynamics.

3.1.1 Least Squares and LASSO Approaches

When the number of covariates is considerably smaller than the number of observations, $np << T$, the VAR model is typically estimated using multivariate least squares:

$$\hat{\mathbf{B}}_{\text{LS}} = \arg\min_{\mathbf{B}} \|\mathbf{Y} - \mathbf{ZB}\|_2^2, \tag{3.3}$$

where $\|\cdot\|_r$ denotes the L_r norm, applicable to both vectors and matrices.

However, in high-dimensional settings, such as spatiotemporal data involving many correlated time series and large lag orders, the number of parameters increases rapidly, making standard least squares unreliable. In such cases, LASSO is particularly effective, as it encourages sparsity in the coefficient matrix, helping to manage strong interdependencies between time series. In the LASSO-regularized VAR (LASSO-VAR) approach, the coefficients are estimated by solving:

$$\hat{\mathbf{B}} = \arg\min_{\mathbf{B}} \left(\frac{1}{2} \|\mathbf{Y} - \mathbf{ZB}\|_2^2 + \lambda \|\mathbf{B}\|_1 \right), \tag{3.4}$$

where $\lambda > 0$ is a regularization parameter that controls the level of sparsity. The first term, $\frac{1}{2}\|\mathbf{Y} - \mathbf{ZB}\|_2^2$, measures how well the model fits the observed data. The second term, $\lambda\|\mathbf{B}\|_1$, adds a penalty that forces many of the coefficients in $\mathbf{B}$ to shrink to exactly zero, helping to select only the most relevant variables.

3.1.2 Alternating Direction Method of Multipliers (ADMM)

While effective, the LASSO regularization in Eq. (3.4) introduces non-differentiability into the loss function, which limits the use of standard optimization techniques. A widely used alternative is the Alternating Direction Method of Multipliers (ADMM), which is both computationally efficient and naturally suited to parallel and distributed settings.

As illustrated in Figure 3.1, the data is distributed across different data owners. This allows the optimization problem in Eq. (3.4) to be reformulated as:

$$\underset{\mathbf{B}}{\operatorname{argmin}} \left(\frac{1}{2} \left\| \mathbf{Y} - \sum_i \mathbf{Z}_i \mathbf{B}_i \right\|_2^2 + \lambda \sum_i \|\mathbf{B}_i\|_1 \right), \tag{3.5}$$

where each $\mathbf{B}_i$ corresponds to a local coefficient matrix maintained by data owner i. This decomposition enables each agent to independently estimate its local parameters using only its own data, without sharing raw observations.

ADMM coordinates these local updates by combining them at each iteration to guide the global model toward minimizing prediction error for $\mathbf{Y}$. The iterative updates follow the system of equations below:

$$\mathbf{B}_i^{k+1} = \underset{\mathbf{B}_i}{\operatorname{argmin}} \left(\frac{\rho}{2} \left\| \mathbf{Z}_i \mathbf{B}_i^k + \overline{\mathbf{H}}^k - \overline{\mathbf{ZB}}^k - \mathbf{U}^k - \mathbf{Z}_i \mathbf{B}_i \right\|_2^2 + \lambda \|\mathbf{B}_i\|_1 \right), \tag{3.6a}$$

$$\overline{\mathbf{H}}^{k+1} = \frac{1}{N+\rho} \left(\mathbf{Y} + \rho \overline{\mathbf{ZB}}^{k+1} + \rho \mathbf{U}^k \right), \tag{3.6b}$$

$$\mathbf{U}^{k+1} = \mathbf{U}^k + \overline{\mathbf{ZB}}^{k+1} - \overline{\mathbf{H}}^{k+1}, \tag{3.6c}$$

where $\overline{\mathbf{ZB}}^{k+1} = \frac{1}{n}\sum_{j=1}^n \mathbf{Z}_j \mathbf{B}_j^{k+1}$, and the variables have the following dimensions: $\mathbf{B}_i^{k+1} \in \mathbb{R}^{p \times n}$, $\mathbf{Z}_i \in \mathbb{R}^{T \times p}$, and $\mathbf{Y}, \overline{\mathbf{H}}^k, \mathbf{U}^k \in \mathbb{R}^{T \times n}$. Each local update of $\mathbf{B}_i$ uses standard ADMM steps [21].

In this formulation, $\overline{\mathbf{H}}$ serves as a global consensus variable that aggregates the local contributions of all agents, ensuring that their updates align toward a common prediction target. The matrix $\mathbf{U}$ is the dual variable, which acts as a running estimate of the discrepancy between the global model and the average of local estimates. Through iterative updates, $\mathbf{U}$ helps enforce agreement across distributed agents, facilitating convergence of the overall optimization process.

3.2 Privacy-Preserving Distributed Learning

Although the ADMM-based LASSO-VAR solver allows agents to compute local updates independently, it requires repeated exchange of intermediate results over multiple iterations. In time series applications, where the response variable $\mathbf{Y}$ is composed of lagged values from the covariates $\mathbf{Z}$, this structural overlap introduces notable privacy risks. As noted by Gonçalves et al. [19], such settings can allow sensitive information from local datasets to be indirectly inferred, compromising privacy guarantees.

These limitations motivated the development of the privacy-preserving protocol adopted in this work, which combines data transformation techniques, such as multiplicative randomization, with the ADMM. This combination allows distributed learning across multiple parties, enabling the estimation of a joint LASSO-VAR model while ensuring that no individual agent or central node can reconstruct the original private datasets.

The proposal supports two collaboration schemes: a centralized communication model with a neutral central hub and a peer-to-peer (P2P) scheme where agents directly exchange encrypted updates. Although the full system proposed in this thesis later introduces a central node for validation, this component is only used in the final stage of the process. The initial collaboration protocol adopts a decentralized peer-to-peer scheme, preserving the original structure and privacy guarantees outlined above. Figure 3.2 illustrates an example topology for a network with three agents, which will be used throughout this chapter to describe the system.

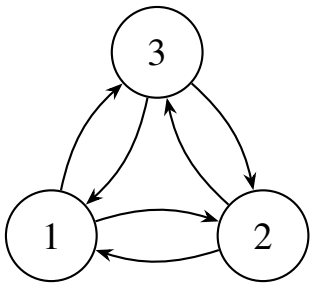


Figure 3.2: Network topology of the collaborative learning protocol in [20]. Arrows indicate the flow of information.

As outlined in Figure 3.3, the adopted process is divided into three main stages leading up to forecasting: encryption, model fitting, and prediction. The details of each stage are discussed throughout this section.

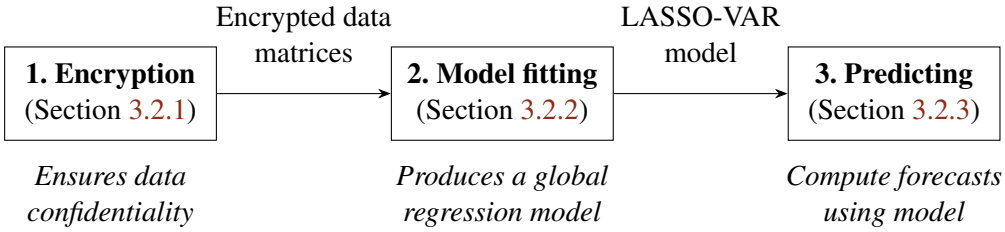


Figure 3.3: Overview of the collaborative learning protocol in [20].

3.2.1 Encryption

To ensure data confidentiality, the system integrates an encryption mechanism based on multiplicative randomization. This approach allows each agent to transform their data independently before any exchange occurs, ensuring that raw data never leaves the local domain. Under certain data dimensionality constraints, as proven by Gonçalves et al. [20], this approach also prevents any participant from reconstructing another's original data.

The encryption process involves two types of transformations:

- A **feature-wise transformation**, where each data owner i multiplies their feature matrix (e.g., $\mathbf{Z}_i$ and $\mathbf{Y}_i$) by a private invertible matrix $\mathbf{Q}_i$, thereby encrypting the feature space;
- A **record-wise transformation**, which adds extra obfuscation by applying a jointly constructed matrix $\mathbf{M}$, created collaboratively without any party knowing its full structure (as detailed below).

These transformations result in the following encrypted matrices $\widetilde{\mathbf{Z}}_i$ and $\widetilde{\mathbf{Y}}_i$:

$$\widetilde{\mathbf{Z}}_i = \mathbf{M}\mathbf{Z}_i\mathbf{Q}_i, \quad \widetilde{\mathbf{Y}}_i = \mathbf{M}\mathbf{Y}_i \tag{3.7}$$

This encryption makes it significantly more difficult to reconstruct individual data points or infer cross-correlations, even if intermediate results or updates are intercepted during training.

A key challenge with this scheme lies in constructing $\mathbf{M}$. Since it operates across all time steps and mixes data from every agent, $\mathbf{M}$ must be collaboratively built in a way that prevents any participant from knowing its full structure. To achieve this, the protocol uses an iterative, peer-based data transformation process in which each agent contributes a private component $\mathbf{M}_i$, such that $\mathbf{M} = \mathbf{M}_1 \cdots \mathbf{M}_n$. Figure 3.4 provides a simplified illustration of this process, highlighting its two main stages. In the first stage, shown with dashed arrows, agents send their locally anonymized data to a designated peer. In the second stage, shown with solid arrows, these values are passed back through an encryption chain, with each agent sequentially applying its private transformation. This mechanism ensures that each agent computes its encrypted inputs $\mathbf{M}\mathbf{Z}_i\mathbf{Q}_i$ and $\mathbf{M}\mathbf{Y}_i$ without ever accessing the original private matrices or the full structure of $\mathbf{M}$. A complete sequence diagram of the protocol is provided in Figure B.1.

This scheme is compatible with both centralized and peer-to-peer communication setups and supports asynchronous execution. It ensures that agents are unable to infer the original data of

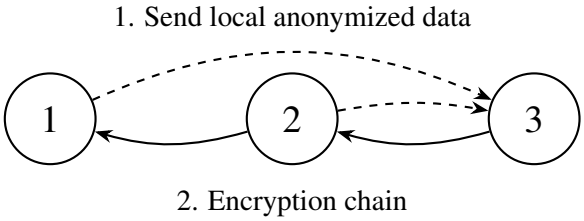


Figure 3.4: Simplified overview of the encryption protocol [20], illustrating the two-stage data transformation process used to collaboratively construct the shared matrix $\mathbf{M}$. Dashed arrows represent the initial exchange of anonymized local data, while solid arrows depict the encryption chain where each agent applies its private transformation.

other participants, while preserving the integrity of the global learning objective. Moreover, the use of obfuscated data does not interfere with the application of regularization techniques, such as LASSO, which remain fully operational in the transformed feature space [20].

Most importantly, the transformations applied during encryption are fully invertible by the data owner. This allows each participant to locally interpret or recover model outputs without compromising the privacy of others. Overall, the encryption framework introduces minimal degradation in model performance while significantly strengthening data privacy within the learning process [20].

3.2.2 Model Fitting

After the data is initially encrypted, the model coefficients can be estimated. This is done using the LASSO-VAR approach, which combines least squares fitting with L1 regularization to promote sparsity in the model.

Given the transformed covariate $\widetilde{\mathbf{Z}}_i = \mathbf{M}\mathbf{Z}_i\mathbf{Q}_i$ and target $\widetilde{\mathbf{Y}}_i = \mathbf{M}\mathbf{Y}_i$ matrices from Eq (3.7), the problem to solve is

$$\underset{\mathbf{B}'}{\operatorname{argmin}} \left(\frac{1}{2} \left\| \widetilde{\mathbf{Y}} - \sum_i \widetilde{\mathbf{Z}}_i \mathbf{B}_i' \right\|_2^2 + \lambda \sum_i \left\| \mathbf{Q}_i \mathbf{B}_i' \right\|_1 \right). \tag{3.8}$$

To support understanding of the distributed model fitting process, Figure 3.5 provides a simplified illustration of a single step in the iterative optimization. Each agent updates its local coefficients using encrypted inputs and communicates the results to its peers. For a complete view of the fitting procedure, refer to the full sequence diagram in Figure B.2.

To ensure the integrity, the convergence of the aggregated residual error is monitored. In particular, if the value $\left\| \mathbf{M}\mathbf{Y} - \mathbf{M}\mathbf{Z}\mathbf{Q}\mathbf{B}^k \right\|_2^2$ does not decrease smoothly during the ADMM iterations (with k being the current step), this indicates that at least one agent is injecting noise or sharing distorted estimates. This mechanism enables the detection of malicious behavior without requiring access to the private data of other participants and thus ensures honesty during fitting. To automate such detection, simple techniques such as monitoring first-order differences are used to flag irregular convergence patterns [20].

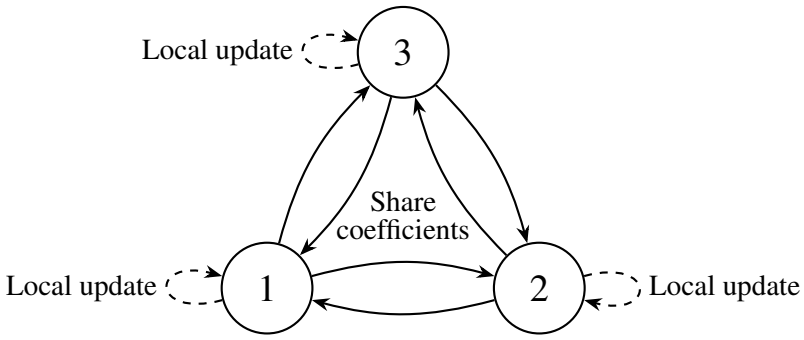


Figure 3.5: Simplified overview of a step in the fitting process [20].

3.2.3 Predicting

Once the model is estimated, predictions are made by aggregating the contributions from all agents. While the model coefficients remain encrypted during training, each agent can recover their local portion of the model and use it in conjunction with private data to compute their contribution to the global forecast.

Let $c_{i\to j,t+h}$ denote the contribution of agent i to the prediction of the jth target variable at time $t+h$. This contribution is calculated based on the lagged observations owned by agent i, as well as their corresponding model coefficients. Formally, if agent i holds a subset of the lagged variables, its contribution is given by:

$$c_{i\to j,t+h} = \sum_{\ell=h+1}^{p} y_{i,t+h-\ell} \hat{\mathbf{B}}_{i\to j}^{(\ell)}, \tag{3.9}$$

where $\mathbf{y}_{i,t+h-\ell}$ represents the lagged values from agent i, and $\hat{\mathbf{B}}_{i\to j}^{(\ell)}$ is the effect of the j-th series on the i-th series at lag ℓ.

The final prediction for target agent j is obtained by summing the contributions from all agents:

$$\hat{y}_{j,t+h} = \sum_{i=1}^{n} c_{i\to j,t+h}. \tag{3.10}$$

In this decentralized setting, each peer is responsible for collecting the contributions required to generate its own forecast. As illustrated in Figure 3.6, agent 1 requests predictions from its peers and then aggregates the received contributions to produce its final output. A full sequence diagram describing the interaction in more detail is shown in Figure B.3. Although not present at this stage, a central coordinating node could also be introduced to handle the request and aggregation process. This extension will be discussed in a later section.

Despite the transformations applied during training, the resulting predictions are expressed in the original data domain. Since the encryption protocol is invertible by the data owner, each agent can interpret the forecast and derive actionable insights without compromising the privacy of others. Specifically, once the encrypted problem is solved, each agent receives their encrypted coefficients $\mathbf{B}_i'$, as computed in Eq. (3.8). By applying their private matrix $\mathbf{Q}_i$, they can locally

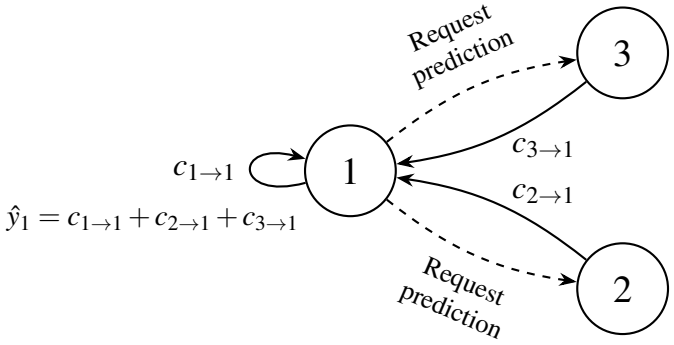


Figure 3.6: Simplified overview of the prediction process [20].

reconstruct the original coefficients from Eq. (3.5) using $\mathbf{B}_i = \mathbf{Q}_i\mathbf{B}_i'$. This ensures that only the corresponding agent can recover their original model. Once reconstructed, each agent can perform forecasting in the original feature space.

3.3 Proposed approach

Although the framework developed by Gonçalves et al. [20] addresses key challenges in privacy-preserving collaborative forecasting, several limitations remain unaddressed.

First, the system lacks an incentive mechanism, relying entirely on voluntary participation without economic justification. This assumption neglects the commercial sensitivities of real-world data owners. Furthermore, this opens the door to free riding, where some agents may benefit from the collaborative model without meaningfully contributing to it.

Second, the framework does not quantify individual contributions to forecasting performance. The encrypted data structure makes it difficult to assess and reward participation fairly and transparently.

Finally, despite strong privacy guarantees, the protocol assumes that agents behave honestly. This issue becomes critical in the presence of financial compensation, where dishonest behavior could result in misdirected rewards.

These limitations motivate the design choices in this thesis, which extends the original framework with the additions highlighted in blue in Figure 3.7. The main contributions of this work, detailed in the following sections, are as follows:

Incentive Mechanism A method for quantifying each agent's contribution to forecasting accuracy and distributing payments accordingly.

Validation Algorithms Two distinct strategies to prevent the manipulation of reported contributions.

As part of the proposed extensions, a central server was introduced to support the prediction and validation stages of the protocol. While the encryption and model fitting phases remain

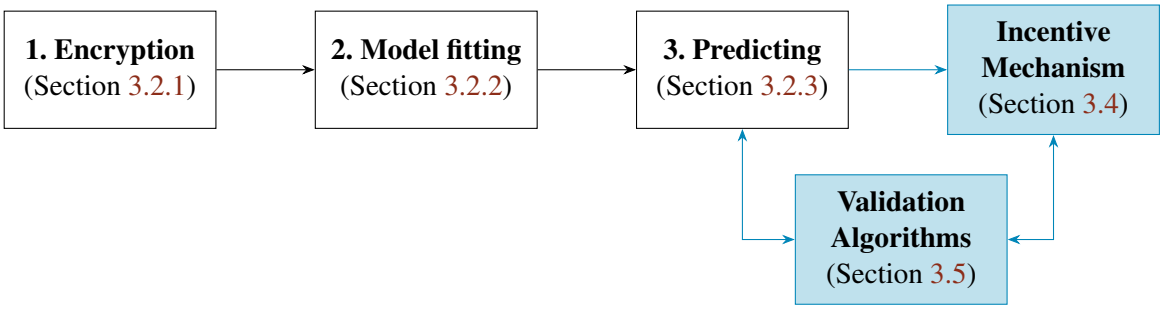


Figure 3.7: Extended overview of the collaborative learning protocol, highlighting the contributions introduced in this work.

fully decentralized, this server is responsible for collecting forecast contributions, distributing payments, and executing validation algorithms, as detailed in the following sections. Even though these operations could be implemented in a fully decentralized manner, doing so would require substantial additional complexity in both design and implementation. Thus, a centralized component was adopted, and full decentralization, potentially supported by blockchain infrastructure, was left as future work.

3.4 Incentive Mechanism

Designing a fair incentive structure requires specific knowledge of the individual contributions of each agent to the prediction. This is enabled by the additive structure introduced in the previous section, where the forecast is expressed as a sum of individual contributions, as defined in Eq. (3.10).

In the extended system, as a first step toward enabling the incentive mechanism, the introduced server takes responsibility for managing the prediction stage. Instead of requesting remote contributions directly from each other, agents now contact the server, which coordinates and aggregates these inputs. The server is thus responsible for computing the remote contribution (RC), which represents the total input from all agents other than the forecast target:

$$RC_{j,t+h} = \sum_{\substack{i=1 \\ i \neq j}}^{n} c_{i \to j,t+h}. \tag{3.11}$$

Figure 3.8 illustrates the prediction process updated with the central node. In this example, agent 1 no longer requests contributions directly from the other agents, but instead contacts the server, which then gathers the necessary values on its behalf.

The remote contribution, $RC_{j,t+h}$, captures the portion of the forecast that is only made possible through collaboration with other agents. The absolute magnitude of this term, $|RC_{j,t+h}|$, determines the overall value of the external assistance provided.

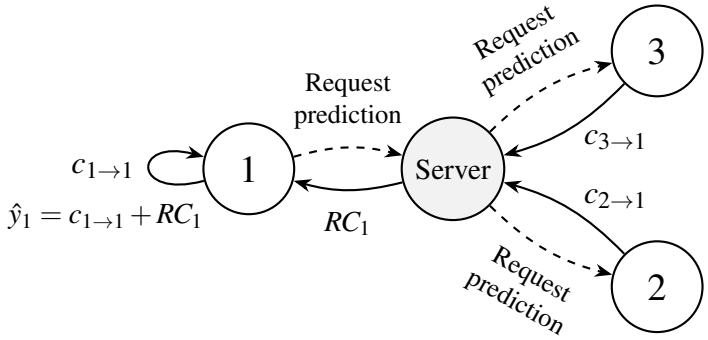


Figure 3.8: Overview of the prediction process with the addition of a central node. Agent 1 requests remote contributions to the server in order to compute its own prediction $\hat{y}_1$. The server aggregates them and then replies with $RC_1 = c_{2\to1} + c_{3\to1}$.

From this, the *total payment* $P_{j,t+h}$ from agent j is defined as:

$$P_{j,t+h} = |RC_{j,t+h}| \cdot fee, \tag{3.12}$$

where *fee* is a fixed parameter agreed in advance, reflecting the assumption that the marginal value of the forecast target remains constant across forecasts. As shown in Eq. (3.12), the payment is proportional to the collaborative portion of the forecast.

To distribute $P_{j,t+h}$ among the contributing agents, the *contribution factor*, $cf_{i\to j}$, is defined for each $i \neq j$ as:

$$cf_{i\to j,t+h} = \frac{|c_{i\to j,t+h}|}{\sum_{\substack{k=1 \\ k\neq j}}^{n} |c_{k\to j,t+h}|}. \tag{3.13}$$

This factor expresses the proportion of the external forecasting effort attributed to agent i, relative to the other contributors.

Finally, the reward allocated to agent i for their contribution to the prediction of target agent j is given by:

$$R_{j\to i,t+h} = cf_{i\to j,t+h} \cdot P_{j,t+h}. \tag{3.14}$$

This ensures that the agent's compensation is directly tied to their quantified impact, rewarding only those who add value and in proportion to their contribution.

3.5 Validation Algorithms

In parallel with the introduction of the incentive structure, a set of related challenges also emerges. One such challenge is the possibility of dishonest behavior from participating agents, aiming to disrupt the system through manipulation or misreporting.

In the context of this work, validation strategies were developed for the case of contribution manipulation. This is the scenario in which contributing (seller) agents attempt to misreport their

contributions in order to receive a higher reward. With respect to Eq. (3.13) and Eq. (3.14), if an agent deliberately increases its own $|c_{i\to j,t+h}|$, this will lead to a proportional increase in the agent's reward, $R_{j\to i,t+h}$. In order to ensure trust in the system, there is a need for a mechanism to monitor the honesty of the sellers.

To simplify the introduction of regulation strategies, these are presented assuming the existence of the supporting central entity presented in Section 3.3, referred to here as the *validator*. Notably, the validator never accesses raw measurement data and operates only on reported contributions, thus maintaining the privacy guarantees of the system.

Two distinct approaches were explored for this purpose and are detailed in Sections 3.5.1 and 3.5.2.

3.5.1 Perturbations

This strategy introduces perturbations, that is, small, controlled changes to reported contributions, in order to detect dishonest behavior among contributing agents. Since the validator has access to the reported values $c_{i\to j,t+h}$, it can leverage this information to assess whether each contribution reflects honest reporting. In order to receive a larger reward, dishonest agents will attempt to inflate their own $|c_{i\to j,t+h}|$, as defined in Eq. (3.13) and Eq. (3.14). In such cases, if reducing any $c_{i\to j,t+h}$ leads to a model with improved error, this suggests that the original contribution may have been overstated. Since the original model was computed before any perturbation, it should be optimal or near-optimal. Any significant improvement under perturbation can thus signal manipulation.

A simple illustration of the process behind this strategy is presented in Figure 3.9, which shows the validation of the remote contributions made to the prediction of agent 1, $\hat{y}_1$. More generally, for every target agent j, the validator creates a perturbed version of the set of contributions shared by the other agents $i \neq j$, that is, $\{c_{i\to j,t+h}\}_{i\neq j}$. This is done by applying controlled modifications to individual values in the set, resulting in a perturbed version $\{c'_{i\to j,t+h}\}_{i\neq j}$. From this, a new remote contribution value RC$'$ is computed by aggregating the values in the perturbed set, according to Eq. (3.11). This value is then sent to the target agent, along with the original RC, since only agent j possesses the true observation needed to compute the forecasting error. Agent j then evaluates the predictive performance of each version using a metric such as MAE in Eq. (2.1). These evaluations are sent back to the validator, who can then draw conclusions about the validity of the received contributions. If the forecast computed using RC$'$ results in a lower error than the one using RC, this suggests that the original contributions may have been misreported.

The perturbation mechanism described above can be generalized in several ways. Instead of generating a single RC$'$, the validator can produce multiple perturbed versions RC^k. As before, if any of these yield a lower prediction error than the original RC, the contributions can be flagged as potentially dishonest. There are also many ways to perturb a set: for example, by modifying only a subset of agents' contributions, or by applying different scaling factors to each. This work explores the following strategies as representative examples for assessing the impact of such perturbations:

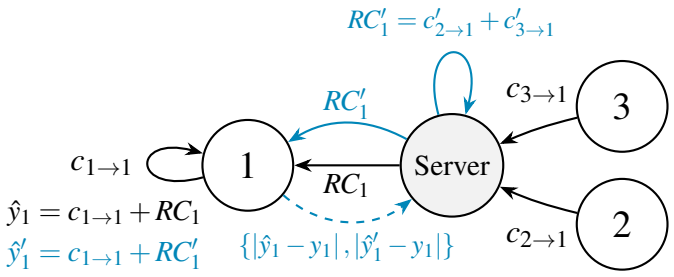


Figure 3.9: Overview of the perturbation-based validation process. Black arrows represent the unaltered prediction flow from Figure 3.8. Blue arrows indicate the additional steps introduced for validation: the server creates a perturbed remote contribution, RC'_1, and sends it to agent 1, who then returns the corresponding prediction errors, shown along the dashed line.

- Scaling all contributions in the set by a common factor θ, resulting in a perturbed set $\{c'_{i\to j,t+h}\} = \{c_{i\to j,t+h} \cdot \theta\}$.
- Scaling a single contribution at a time by a factor θ, generating one perturbed set per agent: $\{c'_{i\to j,t+h}\} = \{c_{k\to j,t+h} \cdot \theta\} \cup \{c_{i\to j,t+h}\}_{i\neq k}$, for each $k \neq j$.
- Scaling each contribution individually across multiple predefined factors θ, generating one perturbed set per θ and per agent: $\{c'_{i\to j,t+h}\} = \{c_{k\to j,t+h} \cdot \theta\} \cup \{c_{i\to j,t+h}\}_{i\neq k}$, for each $k \neq j$ and each θ.
- Assessing individual impact using a Shapley-inspired approach: for each $k \neq j$, evaluate the marginal change in forecasting error when adding $c_{k\to j,t+h}$ to multiple subsets of $\{c_{i\to j,t+h}\}_{i\neq k, i\neq j}$, with the results weighted according to the standard Shapley value formulation.

3.5.2 Zero-Knowledge Validation

An alternative strategy for validating contributions uses zk-SNARKs. Zero-knowledge proof systems, detailed in Section 2.5, allow proving the correctness of a computation without having to reveal its inputs. In the context of this thesis, an agent could prove the correctness of their contribution, $|c_{i\to j,t+h}|$, without disclosing the lagged values, $\sum_{\ell=h+1}^{p} y_{i,t+h-\ell}$, or the estimated coefficients, $\hat{B}_{i\to j}^{(\ell)}$, used to compute it, as defined in Eq. (3.9).

To simplify the implementation of this idea in a zero-knowledge proof circuit, the computation is expressed in the matrix form introduced in Eq. (3.2), where Z represents the covariate matrix and B the coefficient matrix. This abstraction allows the contribution matrix to be written as $Y = ZB$, from which the relevant entry corresponding to $c_{i\to j,t+h}$ is extracted and its absolute value taken.

The goal of the proof system to be developed is to demonstrate that the agent correctly computed the matrix product $Y = ZB$, using the appropriate covariate and coefficient matrices, without revealing their contents. The arithmetic circuit illustrated in Figure 3.10 expresses the computation $Y = ZB$ using private inputs Z and B, and serves as the basis for generating a zk-SNARK proof.

In practice, the matrix multiplication operation is decomposed into a series of scalar additions and multiplications.

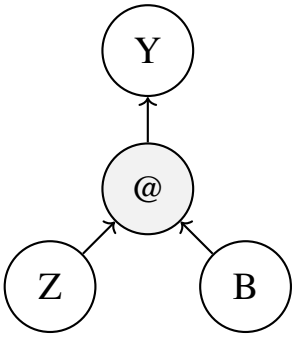


Figure 3.10: Arithmetic circuit for validating the computation of $Y = ZB$ using private inputs Z and B. The symbol @ denotes matrix multiplication.

This circuit proves that the prover possesses some matrices Z and B that satisfy the condition $Y = ZB$. However, this condition alone is not enough to validate a contribution. A malicious agent could still push a fake contribution matrix Y' by fabricating corresponding fake inputs Z' or B' that satisfy the equation, and then generate a valid proof for this incorrect output. To prevent this, the protocol is extended by requiring the prover to commit to both Z and B before generating any proofs.

As detailed in Section 2.5.1, commitment schemes allow a prover to fix private values in advance while keeping them hidden, ensuring that the same Z and B are used consistently across different proofs. Values are typically committed by applying a one-way cryptographic hash function, which produces an output that can later be used for verification. In most cases, the prover can later reveal Z and B directly, allowing the verifier to check that they match the originally committed values. However, this is not the case in the current setting, since both Z and B are meant to remain private and should not be revealed, even after the proof is consumed. Instead, the protocol verifies that the prover used the correct inputs by incorporating the hash functions directly into the circuit. This ensures that the output Y is computed from the same Z and B used to generate the commitment, without requiring those values to be exposed. The final circuit structure is shown in Figure 3.11, where the hashes of Z and B are computed internally and can then be matched against their previously shared commitments, with $\text{Commit}(Z) = h(Z)$ and $\text{Commit}(B) = h(B)$.

Such a structure is particularly effective in the current setting, as it allows commitments to be reused across multiple forecasts. If a new one were to be created for every proof, it would be impossible to verify whether the same inputs were used consistently. In practice, both Z and B are reused several times.

In the case of Z, each agent uses the same input matrix when forecasting for different targets, so sharing both $\text{Commit}(Z)$ and $h(Z)$ allows other agents to verify among themselves that this input remains consistent. Likewise, B is not only shared across target agents but can also be verified across time, since the model is not retrained for every forecast.

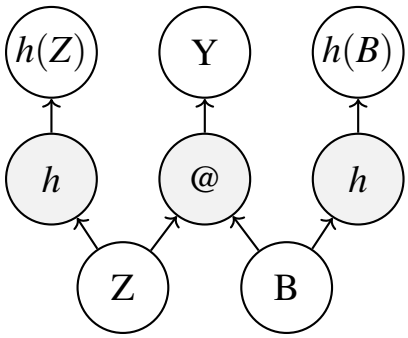


Figure 3.11: Arithmetic circuit for validating the computation $Y = ZB$, with the addition of commitments. The symbol @ denotes matrix multiplication, and h represents a cryptographic hash function.

These constraints significantly increase the difficulty of cheating. Since the commitments are implemented via cryptographic hash functions, agents cannot modify their inputs while still producing matching commitments, as $h(x') \neq \text{Commit}(x)$ for any $x' \neq x$. As a result, to successfully deceive the system, an agent would have to alter an input matrix either across all target agents or throughout a full forecasting period. Such a strategy would lead to a widespread and persistent degradation in forecasting quality, making the manipulation both evident and difficult to sustain without detection.

3.6 Summary

This chapter first described the collaborative forecasting system originally proposed by Gonçalves et al. [20], which enables privacy-preserving model estimation through encrypted data sharing and distributed optimization. The system allows multiple agents to jointly train a LASSO-VAR model without revealing raw data, using a combination of peer-to-peer encryption and ADMM-based learning.

However, the original framework assumes honest behavior, offers no incentive for participation, and lacks mechanisms to quantify or validate individual contributions. To address these limitations, this work introduces two key extensions: an incentive mechanism and two validation algorithms. The incentive mechanism allows forecast targets to reward contributors proportionally to their impact on prediction accuracy. In parallel, two validation strategies are proposed to detect dishonest behavior and ensure the correctness of reported contributions.

Chapter 4

Experimental Setup

This chapter describes the experimental setup used to address the research questions introduced in Section 1.2. More specifically, the goal is to evaluate whether the adopted distributed forecasting system [20] can be extended with a mechanism that offers fair compensation to participants without compromising predictive accuracy or data privacy (RQ1), and to compare the resulting contribution assessments with centralized alternatives (RQ2).

To support this evaluation, two complementary systems were developed. The first is a custom testing framework designed for controlled experimentation with the incentive mechanism and validation algorithms. It enables systematic assessment of predictive accuracy and fairness, both in the distribution of compensation and in the ability to detect dishonest behavior under varied collaborative scenarios. The second is a functional prototype integrated into INESC TEC's existing collaborative forecasting platform, described in Section 3.2. This prototype extends the system by incorporating the proposed incentive mechanism alongside the previously implemented components, which include encryption, model estimation, and prediction.

This chapter describes both frameworks, along with the datasets used to evaluate them.

4.1 Test Framework

The test framework was developed to evaluate forecasting performance and to assess individual contributions and their associated fairness. It was developed in Python 3.10[1] to facilitate the integration of existing predictive algorithms with the novel mechanisms proposed in this thesis.

The framework supports experiments with configurable numbers of agents, validation strategies, and dataset setups, as detailed in Section 4.3. Each run produces an automatic report with detailed metrics on forecasting performance, contribution distribution, and validation outcomes.

4.1.1 LASSO-VAR Model

The model testing component of the framework supports the estimation and evaluation of the LASSO-VAR model in a decentralized setting.

[1] https://www.python.org/

The generated reports include forecasting performance metrics, such as MAE and RMSE. For deeper analysis, the system also outputs the estimated model coefficients per agent, the individual contribution values, $c_{i\to j,t+h}$ defined in Eq. (3.9), along with the resulting contribution factors, $cf_{i\to j,t+h}$, defined in Eq. (3.13) and used in the incentive mechanism. These outputs support a clear interpretation of each agent's impact on the forecast and the fairness of the resulting reward distribution.

4.1.2 Validation Algorithms

The framework also includes a module for evaluating the effectiveness of the validation algorithms developed to handle contribution manipulation, as introduced in Section 3.5. It allows testing all the validation strategies across configurable scenarios, enabling a comparative analysis of their performance under varying conditions.

Perturbations

As described in Section 3.5.1, a perturbation strategy generates a set of manipulated models whose forecasting performance can be evaluated. Several strategies were implemented by varying key parameters and exploring different perturbation schemes based on the ideas discussed earlier. The complete list of implemented strategies is presented in Table 5.5.

Zero-Knowledge Validation

This alternative validation strategy, based on zero-knowledge proofs, was implemented using the `PySNARK` library[2]. It provides a Python interface for constructing zk-SNARK circuits, making it possible to integrate the zero-knowledge validation logic directly into the existing system. For this implementation, a circuit was designed to prove the correctness of a matrix multiplication operation, specifically verifying that $Y = ZB$ holds, without revealing the contents of Z or B.

Evaluation

The validation strategies were evaluated using a set of test scenarios generated by varying key parameters:

- **Number of agents** and the dataset characteristics, including size, train/test split, and model hyperparameters.
- **Number of malicious agents** that artificially inflate their reported contribution values to increase their rewards.
- **Degree of manipulation**, i.e., the scaling factor by which malicious agents amplify their reported contributions.

[2] https://github.com/meilof/pysnark

By varying these parameters, the system is able to simulate both honest and adversarial environments, allowing for a comprehensive assessment of each algorithm's robustness. Specifically, it evaluates whether the strategy can accurately detect the presence of manipulation when malicious agents are involved, avoid false positives in fully honest scenarios, and correctly identify which participants are dishonest.

4.1.3 Centralized Baseline Comparison

To assess how the contribution assessments derived from the distributed system compare in effectiveness and fairness to those produced by centralized algorithms (RQ2, Section 1.2), a baseline must be established under a centralized setting. This baseline can be obtained using a standard contribution estimation algorithm, assuming global access to all data.

Shapley values (Section 2.3.1) are used for this comparison, given their established role in Federated Learning as a reference for quantifying individual impact under full information [21]. These values reflect the average marginal contribution of each participant across all possible coalitions and thus offer a strong baseline for evaluating fairness in contribution-based mechanisms. If the distributed contribution factors align closely with the obtained Shapley values, this indicates that the proposed method provides incentives that are comparable in fairness to the established centralized reference allocation, while still maintaining system properties such as privacy preservation.

For each simulated agent, Shapley values are approximated by repeatedly measuring the change in forecasting performance when subsets of external features are added to the agent's own input data. This is done over multiple random permutations, providing an estimate of each external agent's marginal contribution under a centralized setting. The resulting values can then be compared to the *contribution factors*, $cf_{i \to j}$ as in Eq. (3.13), produced by the proposed system.

4.2 Prototype

The prototype serves to demonstrate the integration of the incentive mechanism into the existing privacy-preserving forecasting workflow, enabling end-to-end evaluation under realistic operational conditions. Its development builds upon the methodology proposed by Gonçalves et al. [20], as outlined in Section 3.2, and was implemented on top of INESC TEC's existing forecasting framework, originally developed for the European project ENERSHARE[3].

The overall behavior of the adopted system is illustrated in Figure 4.1, which presents a sequence diagram with the three main operations in the process: encryption, model fitting, and prediction. The illustrated scenario involves three agents. Agent 1 is denoted as the buyer, while Agents 2 and 3 act as sellers. Although in practice all agents assume both roles, this naming convention is used throughout the document to simplify the diagrams and isolate specific behaviors.

[3] https://enershare.eu/

The first two stages of the interaction, *Encryption* and *Model Fit*, follow the procedures defined in the original framework. These stages are encapsulated as references in the diagram to maintain focus on the thesis's main contributions. The third stage, *Prediction*, is where this thesis introduces new mechanisms related to contribution-based incentives and validation. On the original system, when the buyer wants to compute a forecast for its own target y_1, it initiates prediction requests to the other agents in the system. Each seller responds with its local contribution, denoted $c_{2\to1}$ and $c_{3\to1}$ in the diagram. After receiving the remote contributions, the buyer aggregates them with its own contribution to compute $\hat{y}_1$. This structure allows the prediction process to be additive and distributed, setting the foundation for a contribution-based incentive mechanism.

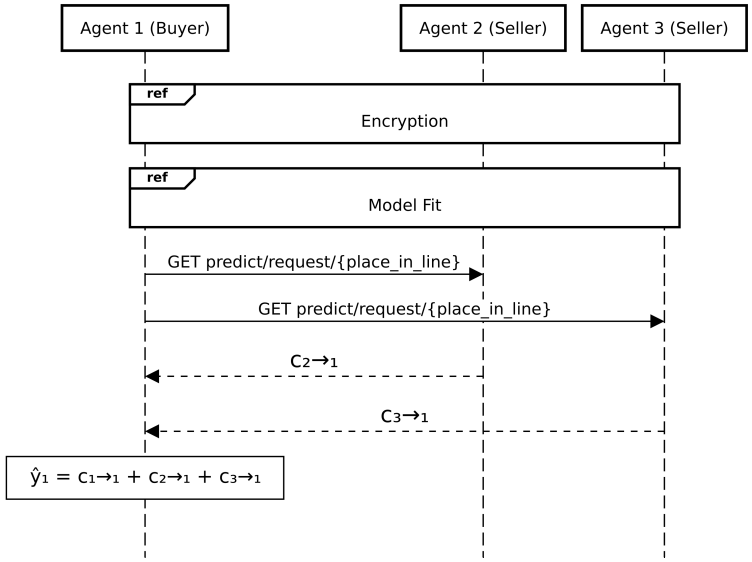


Figure 4.1: Sequence diagram of the initial prototype. After encryption and model fitting stages, agent 1 requests contributions from other agents to compute its own prediction $\hat{y}_1$.

Having outlined the baseline implementation, the experimental integration of the incentive mechanism will now be described. Figure 4.2 illustrates the modifications introduced to support a payment system within the prediction workflow. To support the integration of the incentive mechanism, a centralized server is introduced at this stage to manage remote contributions and coordinate payments. As discussed in Section 3.5, this design choice simplifies implementation, deferring full decentralization to future work.

In this updated flow, the buyer no longer requests remote contributions directly from each individual agent. Instead, the request is sent to the server, which gathers the required values $(c_{2\to1}, c_{3\to1})$ and computes the total remote contribution RC_1, as defined in Eq. (3.11). It also calculates the total cost using Eq. (3.12), which is communicated to the buyer along with a unique prediction identifier. This identifier is used to link the forecast request to its corresponding payment and to ensure that rewards are correctly attributed once the transaction is complete. Upon receiving this information, the buyer must execute the payment through the specified channel to get access to the prediction result. After successful payment (verified via an acknowledgment), the server distributes rewards to contributing agents proportionally, according to Eq. (3.14).

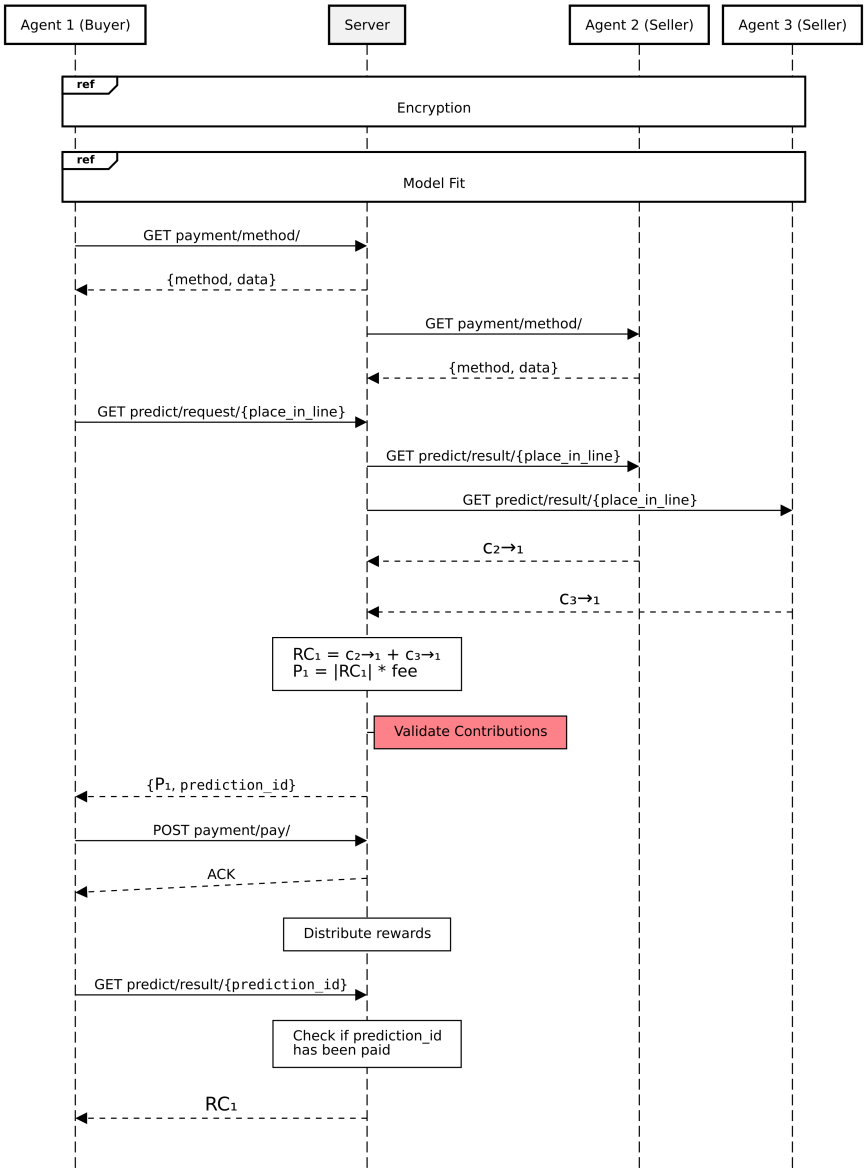


Figure 4.2: Sequence diagram of the updated prototype. Now, the request for RC is intercepted by a central server that must receive payment before delivering the data. The red rectangle highlights the stage where validation mechanisms are intended to be integrated.

Blockchain was selected over traditional payment systems to provide a trustless and verifiable way to automate reward distribution. More specifically, Ethereum[4] was adopted for handling payments, and the prototype was tested using a local testnet provided by Ganache.[5] Although Ethereum was chosen as the default payment mechanism, the system was designed with flexibility in mind. Each participant communicates their preferred payment method at the beginning of the interaction (as shown in the initial '`GET payment/method/`' exchanges in the diagram). This design allows for seamless integration of alternative payment systems in future versions of the

[4] https://ethereum.org/
[5] https://archive.trufflesuite.com/ganache/

prototype.

A natural next step would be to integrate the validation strategies discussed in Section 3.5. These mechanisms have not yet been included in the function prototype due to the project's timeframe, although they have already been designed and validated for this setting using the complementary testing framework. Nonetheless, their inclusion would bring the prototype closer to the intended end-to-end design. Validation should be performed by the Server after receiving the remote contributions, as highlighted in red in Figure 4.2.

4.3 Datasets

The experimental evaluation of the proposed framework relies on the usage of appropriate datasets. Both synthetic and real-world data were used in order to assess the system's capabilities under controlled and realistic conditions. This dual approach ensures a comprehensive evaluation, allowing for scenario-specific testing while also verifying performance in realistic settings.

4.3.1 Synthetic

Synthetic data was used to simulate controlled scenarios and isolate specific system behaviors. This approach allows systematic testing of the incentive mechanisms and validation algorithms under known conditions. By precisely defining the data distributions and inter-agent relationships, it becomes possible to manipulate and evaluate how individual agents influence each other's forecasts. This is critical for assessing whether the contribution-based reward system correctly identifies and compensates meaningful input. Additionally, synthetic data enables the injection of adversarial behaviors in a systematic way, allowing for repeatable testing of the system's resilience against dishonest agents.

Two synthetic datasets were developed to evaluate the behavior of the system under different conditions:

Linear Dataset: The first dataset consists of a simple linear relationship between inputs and outputs. The covariate matrix Z and the coefficient matrix B were generated using a uniform distribution over $[0, 10]$ and $[0, 1]$, respectively. The target matrix Y was then computed as $Y = ZB + \varepsilon$, where the additive white noise term ε was also sampled from a uniform distribution over $[0, 1]$. This construction results in a dataset with an easily interpretable linear dependency.

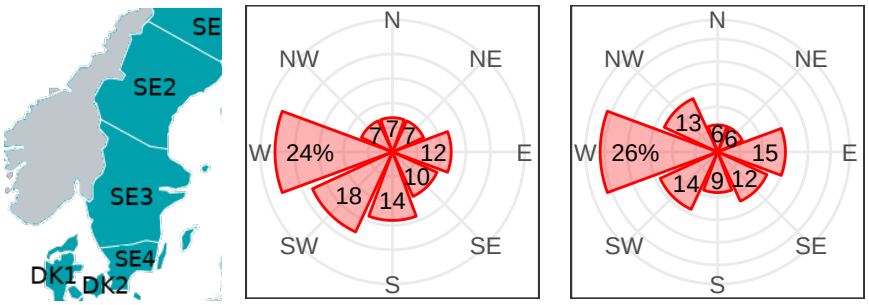
VAR Dataset: The second dataset simulates a more complex scenario using a VAR process with multiple lags, as defined in Eq. (3.1). The covariate matrix Z and target matrix Y are derived from a simulated stationary VAR process, where each observation depends on p lagged values of all variables. This design incorporates temporal dependencies commonly found in real-world time series, making the forecasting task more challenging and suitable for testing the system's performance.

Together, these synthetic datasets allow the proposed system to be evaluated under both simplified and controlled conditions, making it possible to isolate specific behaviors without the variability present in real-world data.

4.3.2 Nord Pool

The use of real-world data introduces challenges that cannot be fully captured through synthetic simulations. Empirical datasets inherently contain noise, irregular patterns, and potentially missing values, all of which are crucial for evaluating the system's performance under practical conditions. This aspect is essential to avoid overly optimistic conclusions that may arise from testing only on clean or idealized data, and to better understand the system's performance in real-world scenarios.

Nord Pool[6] operates the leading power exchange in Europe, covering countries in the Nordic and Baltic regions. The dataset used in this work contains wind power generation data from Sweden and Denmark, divided across the regions shown in Figure 4.3. It contains hourly measurements that span two years between September 2016 and 2018.



(a) Regions. (b) Wind direction in Copenhagen (DK2) and Malmo (SE4).

Figure 4.3: Nord Pool dataset regions and wind rose diagrams for two of the six regions. Adapted from [21].

The six regions in this dataset correspond to six agents in the system, each collaborating to compute their one-hour ahead local forecasts. An important advantage of this setup is that data from neighboring regions can help improve local forecast accuracy, since wind patterns often affect nearby areas in similar ways. One important particular aspect is that wind typically flows from Denmark towards southern Sweden, as depicted in Figure 4.3(b), suggesting that regions DK1 and DK2 are likely to have a strong influence on the forecasts for SE3 and SE4.

Figure 4.4 shows the one-hour ahead correlation matrix for the Nord Pool dataset. Each value reflects the correlation between a region's current value and the one-hour lagged values of other regions. As expected, the strongest correlations appear along the diagonal, indicating that each

[6] https://www.nordpoolgroup.com/

region is most influenced by its own recent history. Secondly, neighboring regions also tend to exhibit stronger lagged correlations, which reflects the local structure of wind patterns. This spatial pattern becomes especially clear when observing how the intensity of the correlations generally fades as one moves away from the diagonal. Moreover, some exceptions are particularly informative. For instance, DK1 shows a stronger influence on SE4 than SE4 does on DK1, even though the correlation is high in both directions. This aligns with the suggested wind patterns and is expected to be reflected in the reward distribution later on.

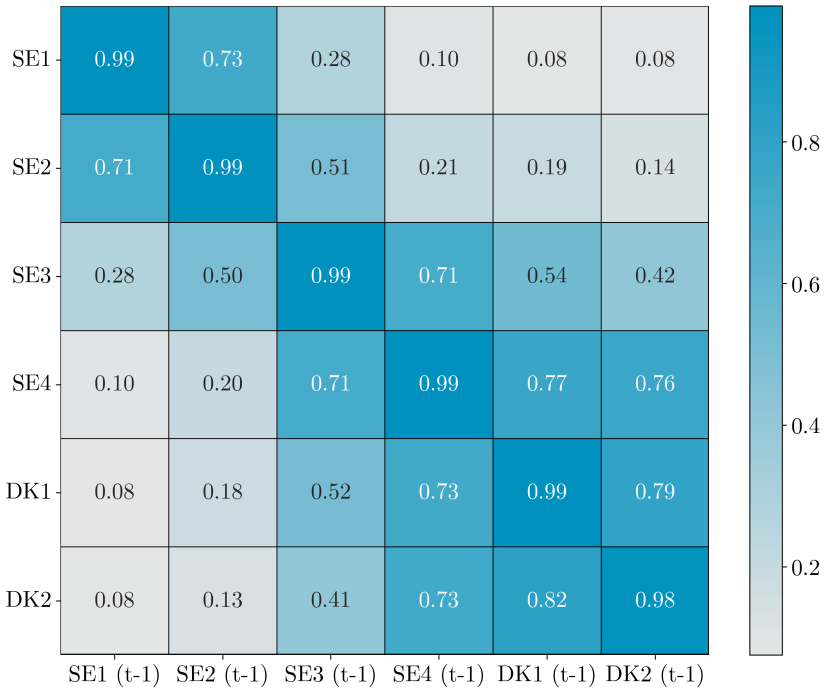


Figure 4.4: Nord Pool one-hour ahead correlation matrix.

Another important aspect of the Nord Pool dataset is its high variability, which is an inherent characteristic of renewable energy sources. This variability increases the complexity of the forecasting task, meaning that slightly lower accuracy may be expected compared to scenarios with more predictable or less volatile data. Table 4.1 reports the standard deviation and the coefficient of variation (CV) for each region. The relatively high CV values across all regions confirm the presence of significant fluctuations in the data. Moreover, the maximum power values differ notably between regions, indicating significant variation in installed wind capacity. DK1, in particular, stands out with a maximum output of 3771 MWh, the highest among all regions.

Table 4.1: Standard deviation and coefficient of variation for each region of the Nord Pool dataset.

Region	**Standard Deviation [MWh]**	**Coefficient of Variation**	**Maximum Power [MWh]**
SE1	127.660	0.830	517
SE2	509.914	0.830	1982
SE3	482.397	0.734	2071
SE4	362.209	0.769	1468
DK1	891.742	0.740	3771
DK2	262.493	0.831	1027

Chapter 5

Results

This chapter presents the experimental outcomes of the proposed system, organized according to the research questions defined in Section 1.2.

The first section addresses **RQ1**, which investigates whether a distributed forecasting system can be extended with a mechanism to offer participants fair compensation for their contributions, without compromising predictive accuracy or data privacy. This is a complex question that involves several different aspects of the proposed framework. To properly answer it, a variety of results are presented, covering the different components developed throughout the work.

The second section focuses on **RQ2**, which evaluates how the contribution assessments produced by the distributed system compare to those generated by centralized methods. Here, the goal is to assess whether the distributed approach is comparable in fairness to centralized baselines, such as Shapley value attribution.

5.1 Research Question 1

This section presents the results related to RQ1, from Section 1.2:

> *Can a distributed forecasting system be extended with a mechanism to offer participants fair compensation for contributions while preserving accurate predictions and data privacy?*

To address this question, the section is organized into three key areas, each corresponding to one of the core requirements: accuracy, privacy, and fairness.

First, accuracy is assessed by analyzing the forecasting performance of the system across synthetic and real-world datasets. Second, privacy is considered in terms of the system's ability to protect sensitive data while maintaining predictive utility. Third, fairness is examined through the lens of the incentive mechanism, evaluating whether rewards are distributed in proportion to each agent's contribution and whether dishonest behavior can be effectively discouraged.

The structure of this section, along with the specific evaluation perspectives associated with each component, is outlined in Figure 5.1. It is important to note that, in this section, the incentive

mechanism was evaluated using the functional prototype, while the remaining components were assessed using the testing framework.

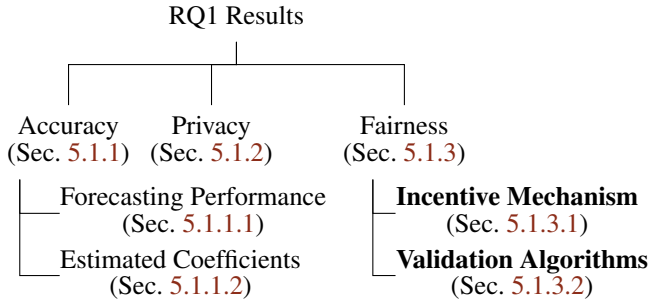


Figure 5.1: Structure of the section addressing RQ1, organized by core evaluation components. Bold elements refer to the evaluation of components that were originally developed as part of this work.

5.1.1 Accuracy

Although the VAR model itself is not a contribution of this work, its integration is central to evaluating whether the extended system can deliver accurate forecasts. The results presented here assess the model's ability to capture temporal and geographic dependencies between participants and demonstrate that forecast quality is preserved within the extended, incentive-compatible framework.

The model was tested on the three datasets described in Section 4.3, each containing data from six agents, with one covariate per agent and a total of eight thousand time steps. A VAR model was fitted separately for each dataset, with hyperparameters λ and ρ tuned via Bayesian optimization using 10-fold cross-validation.

To evaluate the model, both its forecasting performance (Section 5.1.1.1) and the accuracy of the estimated coefficients (Section 5.1.1.2) were considered.

5.1.1.1 Forecasting Performance

This subsection evaluates how well the LASSO-VAR model performs as a forecasting tool within the proposed framework. The analysis focuses on the model's ability to produce accurate predictions across datasets with varying levels of complexity and realism.

The corresponding results are shown in Table 5.1, which reports the forecasting errors obtained for each dataset. To allow for meaningful comparison across datasets with different scales, raw error metrics (MAE, RMSE) were normalized using both the standard deviation (std-nMAE, std-nRMSE) and the maximum value (max-nMAE, max-nRMSE) of each dataset. This normalization ensures that the results reflect relative performance rather than being influenced by data magnitude.

As expected, the Linear dataset resulted in very low error rates, due to the simplicity and low variability of the underlying synthetic data. In contrast, the VAR dataset, although also artificially generated, presents greater complexity due to temporal dependencies and interactions between

variables. This makes it a more challenging forecasting task. Nonetheless, the model was able to maintain low error rates based on the normalized metrics. The Nord Pool dataset represents real-world renewable energy data, which is inherently noisy and highly variable. Despite this, the model achieved strong results, with std-nMAE and std-nRMSE values below 0.15.

Table 5.1: Forecast metrics for the VAR model across synthetic and real datasets.

Dataset	**std-nMAE**	**std-nRMSE**	**max-nMAE**	**max-nRMSE**
Linear	0.048	0.055	0.007	0.008
VAR	0.252	0.316	0.049	0.061
Nord Pool	0.081	0.129	0.013	0.020

Another relevant assessment of the model's performance is to examine how the system behaves without the values from contributing agents. That is, by relying only on each agent's locally generated predictions. For that, the same experimental setup was used, and performance reports were generated for scenarios without external input. The resulting MAE and RMSE for each configuration are presented in Table 5.2.

Table 5.2: Forecast performance of the VAR model across datasets, with and without remote contributions. The percentage change is relative to the remote setting. Positive values indicate increased error when remote contributions are excluded.

Dataset	**MAE**			**RMSE**		
	Remote	**No Remote**	**Change (%)**	**Remote**	**No Remote**	**Change (%)**
Linear	0.252	11.562	+4488.1%	0.291	12.465	+4183.5%
VAR	0.181	0.363	+100.55%	0.226	0.476	+110.62%
Nord Pool	42.372	47.957	+13.181%	67.580	74.748	+10.607%

As expected, the results show that removing remote contributions leads to a significant decline in forecasting performance across all datasets. Interestingly, the linear dataset is the most affected by this change, showing an increase in error of over 4488.1% in MAE and 4183.5% in RMSE when remote contributions are excluded. This can be explained by the fact that, unlike the other datasets, each agent is not necessarily the most relevant contributor to its own prediction, and thus removing remote contributions has a much greater impact.

A detailed visualization of these results is provided in Figure C.1. It illustrates that, in both the VAR and Nord Pool datasets, the forecasts are mainly driven by the target agent's own data. Incorporating remote contributions leads to small but meaningful adjustments that enhance overall accuracy, demonstrating the value of collaborative forecasting.

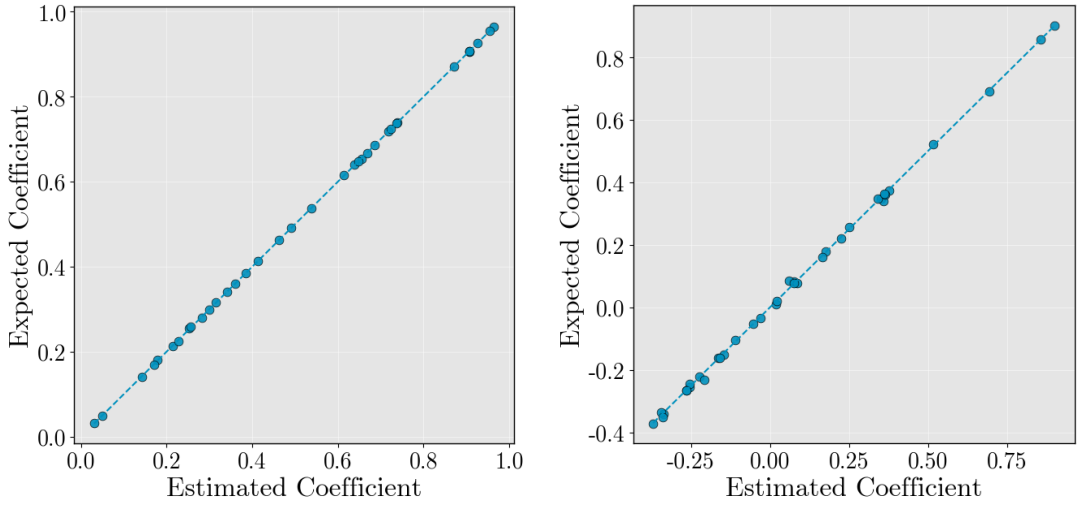
5.1.1.2 Estimated Coefficients

A complementary way to assess the model is to compare the estimated coefficients, obtained after model training, with the known values used to generate the data. This is only possible in the synthetic datasets, where the true coefficients are known by design. Still, this comparison is

particularly relevant, as the estimated coefficients (B matrices) form the foundation for the reward calculations in the incentive mechanism, as detailed in Section 3.4.

Figure 5.2 illustrates the comparison between the expected and estimated coefficients for both synthetic datasets obtained from the previous experiment. As shown, the model accurately approximates the B matrix, with most points lying close to the reference $y = x$ line. This strong alignment in the Linear dataset confirms that the estimation process is functioning correctly, providing a reliable foundation for the incentive mechanism.

In the VAR dataset, the estimation is only slightly less precise, which is expected given the increased complexity and temporal dependencies present in the data. Even so, the model successfully captured these patterns with a high level of accuracy.



(a) Linear Dataset (b) VAR Dataset

Figure 5.2: Estimated coefficients plotted against expected values for both synthetic datasets.

To complement the visual alignment between estimated and expected coefficients, the Pearson correlation coefficient was computed for each synthetic dataset. For the Linear dataset, the result was $r = 1.0000$ ($p = 6.05 \times 10^{-81}$), indicating perfect linear agreement. For the VAR dataset, the correlation remained extremely high at $r = 0.9997$ ($p = 3.81 \times 10^{-56}$), despite the increased complexity. These values confirm the precision of the estimation process and reinforce the reliability of using the resulting coefficients in the reward mechanism.

5.1.2 Privacy

Although privacy is not directly assessed through empirical testing, it remains a core aspect of RQ1 and is addressed through the system's architecture. This section provides a qualitative evaluation of the mechanisms that support privacy throughout both the training and forecasting phases. Rather than being a performance outcome, privacy in this context is preserved by design, through the use of encryption protocols and structural safeguards.

Secure model training

As discussed in Section 3.2.1, the training protocol ensures that each agent performs local computations on encrypted data. Only masked or aggregated intermediate results are ever exchanged, preventing the leakage of raw inputs during model estimation. The use of multiplicative randomization and jointly constructed transformation matrices makes it highly unlikely that any participant, including a central coordinator, can infer private values from the shared information.

Secure forecasting

Forecasting operations are also designed to maintain data confidentiality, as detailed in Section 3.2.3. The validator node receives only remote contributions from collaborating agents and never observes the target agent's local input or full forecast. Because each prediction is decomposed into additive parts, as shown in Eq. (3.11), the validator only has access to remote contributions and thus cannot reconstruct individual forecasts or access sensitive agent-specific patterns using the LASSO-VAR model structure.

5.1.3 Fairness

Fairness was evaluated by first running the full prototype on real-world data to analyze how rewards were distributed under the proposed incentive mechanism, and then by testing the validation strategies in controlled scenarios involving dishonest behavior. The following sections report the results of both evaluations.

5.1.3.1 Incentive Mechanism

Running the prototype provides a valuable opportunity to evaluate the behavior of the incentive mechanisms under realistic conditions. Additionally, it enables an end-to-end assessment of the complete system, validating whether the proposed solution effectively meets the requirements outlined in the problem definition.

In this experiment, the prototype was executed using real data from the Nord Pool dataset (Section 4.3.2). Six agents were used, corresponding to all six geographical regions in the dataset. The model was trained using data from the entirety of 2017. Following the training, forecasts were generated for the first month of 2018.

The system is expected to produce accurate forecasts and a fair distribution of rewards. More specifically, influential regions, such as those located in Denmark, are anticipated to accumulate higher compensation due to their stronger impact on neighboring regions.

To gain insight into the relative influence of each region, the total contribution factor associated with each agent was computed. That is, for each agent, the sum of its contribution factors to all other agents' prediction targets was aggregated. Although this metric is not derived from a formal model interpretation, it serves as a useful visual tool to understand the influence each agent has in

the system and to anticipate how rewards are likely to be distributed. Figure 5.3 presents a bar plot of the aggregated contribution factors, with regions ordered by descending influence.

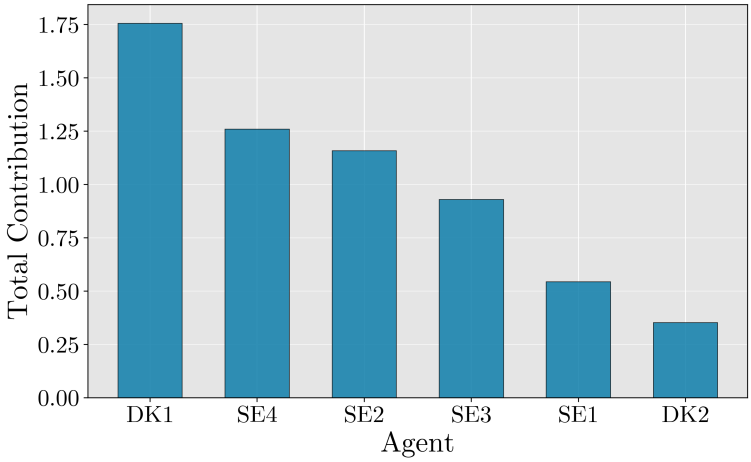


Figure 5.3: Sum of contribution factors for each region, indicating their relative influence on the predictions of other agents.

Each agent that participated in the simulation, as well as the server, was represented by a dedicated account, with a starting balance of 100 ETH. Each agent is mapped to a specific region code, as shown in Table 5.3. The *fee* parameter used for reward calculation was set to 0.1 ETH.

Table 5.3 displays the final account balances and transaction counts after the prototype was run. As expected, the server executed the highest number of transactions, acting as the coordinating entity responsible for managing predictions and distributing rewards.

Table 5.3: Final account balances and transaction counts per agent, ordered by balance.

Region	**Final Balance (ETH)**	**Received (ETH)**	**Sent (ETH)**	**TX Count**
DK1	**102.99**	**5.71**	2.72	1
SE2	101.79	2.82	1.03	1
SE3	101.42	2.90	1.48	1
SE1	98.74	0.94	2.20	1
SE4	97.96	3.35	5.39	1
DK2	97.11	0.92	3.81	1
Server	100.00	16.64	16.64	30

The final balances are consistent with the influence patterns shown in Figure 5.3, confirming that agents with stronger contributions received higher rewards. This suggests that the reward mechanism behaves as expected, fairly distributing value based on each participant's impact on the forecasts.

To ensure the accuracy of the forecasts produced by the prototype, time series plots were generated for each agent, comparing the expected values against the model's predictions over the month of January 2018. As shown in Figure C.2, the predicted curves closely follow the expected

values, demonstrating strong alignment and confirming the system's ability to generate reliable forecasts under realistic conditions.

5.1.3.2 Validation Algorithms

The next step in evaluating fairness is to assess the performance of the proposed validation algorithms. However, before introducing the results from these strategies and evaluating their effectiveness in detecting dishonest agents, it is important to first illustrate the practical impact that manipulated contributions can have on forecasting accuracy. Figure 5.4 provides a visual comparison between forecasts under different agent behaviors using Nord Pool data, showing the forecast results for region SE1. The blue curve represents the expected values, serving as a reference. The yellow curve shows the predicted values generated under normal conditions, closely aligning with the expected results. To simulate dishonest behavior, two additional scenarios were introduced by modifying the coefficient matrix of agent SE2.

In the first case, represented by the green line, the agent's coefficients were scaled by a factor of 1.1. This minor manipulation leads to forecasts that differ only slightly from the predicted values. In contrast, the red line illustrates the outcome of a more extreme manipulation, where the coefficients were scaled by a factor of 5. This leads to significantly distorted forecasts, clearly diverging from the expected and predicted results.

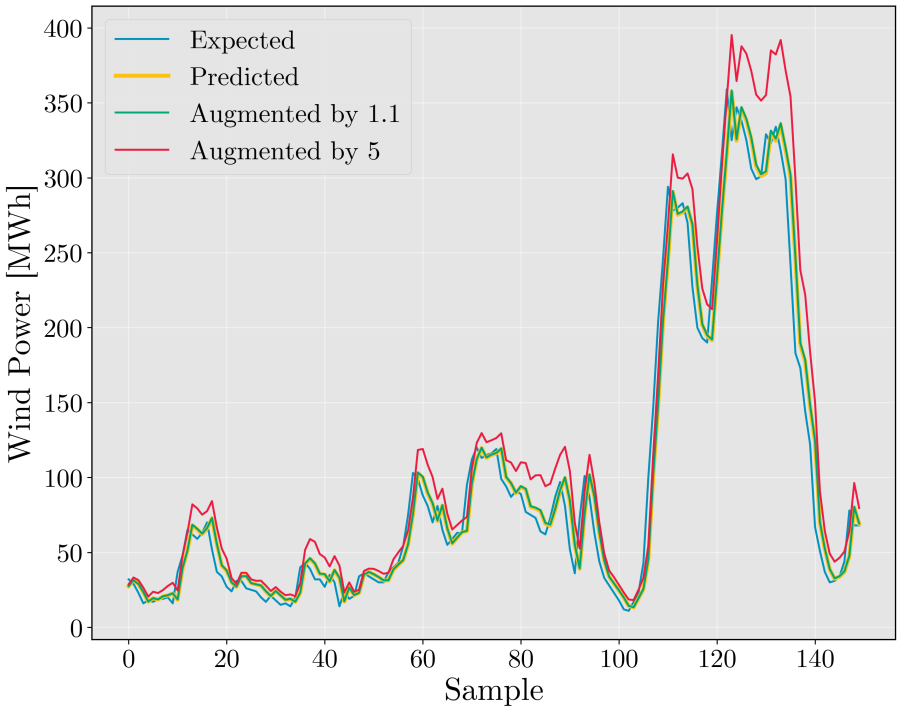


Figure 5.4: Agent SK1 forecast comparison across different agent behaviors. The expected and predicted values are contrasted with results from models where agent SK2's coefficients were scaled by 1.1 and 5, respectively.

These results are supported by the quantitative error metrics presented in Table 5.4. When no dishonest agents are present, the model achieves an MAE of 42.373 and an RMSE of 67.580. Manipulations of matrix B show a significant degradation of forecasting quality.

Table 5.4: Error metrics under varying levels of contributor dishonesty.

Scenario	**MAE**	**RMSE**
No dishonest agents	42.373	67.580
One dishonest agent (scale 1.1)	45.611	72.130
One dishonest agent (scale 5)	331.376	959.370

These results emphasize the necessity of implementing validation mechanisms to regulate the contributions. What follows is an assessment of the two proposed strategies, evaluated using the test scenario described in Section 4.1.2.

The testing scenarios were configured by varying the following parameters:

Number of agents Simulations were conducted with either 3 or 5 participating agents.

Malicious participation Scenarios included no dishonest agents, a single dishonest agent (agent 1 or agent 2), or two dishonest agents acting simultaneously.

Manipulation intensity Dishonest agents inflated their contributions by scaling their coefficients by factors of 1.1 or 5.

Each validation strategy was analyzed according to the following criteria:

Presence The strategy's ability to detect the existence of dishonest behavior within the forecasting process.

Identity The strategy's ability to correctly determine which agents were responsible for the manipulation.

Based on the combinations of parameters described above, a total of 14 tests were conducted to assess presence detection, and 12 tests were carried out to evaluate identity detection.

Perturbations Each of the previously defined perturbation strategies was evaluated across all test scenarios described above. A summary of the results, along with the corresponding strategies, is presented in Table 5.5.

The results show that most perturbation strategies are effective at detecting the presence of dishonest behavior. This suggests that manipulating reported contributions and observing the resulting forecast error is a reliable approach for identifying when something is wrong.

However, identifying exactly *which* agent is responsible for the manipulation proves to be a more difficult task, namely when using perturbations. The challenge lies in the nature of the forecast itself, which is computed as a summation of all agents' contributions. In this additive

Table 5.5: Detection performance of different perturbation strategies for contribution validation. The table reports the number of successful detections of dishonest behavior (Presence), correct identification of responsible agents (Identity), and the total number of tests passed.

Strategy	**Presence (out of 14)**	**Identity (out of 12)**	**Total (out of 26)**
Reduce contributions of all agents by 1%.	**14**	2	16
Reduce each agent's contribution, one at a time, by 1%.	**14**	**7**	**21**
Remove each individual contribution, one at a time.	8	4	12
Reduce each agent's contribution, one at a time, by 50%.	8	4	12
Reduce each agent's contribution, one at a time, by multiple predefined factors (0, 0.01, 0.25, 0.5, 0.75, 0.99).	**14**	4	18
Reduce each agent's contribution, one at a time, based on the agent's average impact.	**14**	**7**	**21**
Reduce contributions for multiple agents based on their average impact.	**14**	5	19
Decrease one agent's contribution by 10% while increasing others by 10%, one at a time.	8	4	12
Use Shapley-like value estimation based on combinations of agents.	12	**7**	19

structure, altering the contribution of any single agent can affect the overall error, sometimes improving it, even if that agent was not the one behaving dishonestly.

As a result, while the perturbation validator works well for detecting the *presence* of manipulation, it is less reliable for pinpointing the *identity* of the malicious participant.

Zero-Knowledge Validator The application of zero-knowledge proofs (ZKPs) to validate contributions, as described in Section 3.5.2, was also evaluated within this framework.

The results, shown in Table 5.6, indicate that all tests were successfully passed. This outcome is expected, given the design of the validator and the properties of zero-knowledge proofs. In particular, the *Soundness* property ensures that a false statement cannot produce a valid proof. Under the current test structure, this implies that dishonest agents are unable to produce valid proofs for manipulated contributions and can therefore be consistently identified.

Table 5.6: Detection performance of applying ZKPs to contribution validation. The table reports the number of successful detections of dishonest behavior (Presence), correct identification of responsible agents (Identity), and the total number of tests passed.

Strategy	**Presence (out of 14)**	**Identity (out of 12)**	**Total (out of 26)**
Use zk-SNARKs to prove matrix multiplication.	14	12	26

While these results suggest that ZKPs are a highly reliable tool for contribution validation in this setting, it is important to note that they reflect the assumptions and structure of the current implementation. Further testing under different conditions may be needed to fully assess the robustness of this approach.

5.2 Research Question 2

This section presents the results related to RQ2, from Section 1.2:

> *How do the contribution assessments derived from the distributed system compare in effectiveness and fairness to those generated by existing centralized algorithms under similar conditions?*

To address it, Section 4.1.3 introduces a centralized baseline used to compute agent influence with full data access. This section presents the results of that experiment and compares them with the values produced by the testing framework.

Shapley values were computed for both the Nord Pool and VAR datasets. These were used as a centralized reference to evaluate the distributed contribution estimates generated by the proposed framework. While the Nord Pool dataset captures realistic forecasting conditions, the VAR dataset enables more controlled testing using the known generated coefficients.

It is worth noting that this comparison implicitly treats the Shapley values as a ground truth for the contribution of each agent. While this is not entirely accurate, as the Shapley method relies on a different estimation process, it does benefit from full data access and the ability to simulate marginal effects by re-training on various feature subsets. Even so, it provides a strong benchmark to evaluate how well the distributed system captures individual impact under both real and controlled conditions.

To compare the contribution factors with the Shapley values, MAE and three other evaluation metrics were used: cosine similarity, L1 distance, and Spearman correlation. These metrics help compare how similar the distributed contributions are to the Shapley values from the centralized setup. L1 distance and MAE are error-based metrics, where lower values indicate better alignment. Cosine similarity and Spearman correlation range from -1 to 1, where values closer to 1 indicate stronger agreement in direction and ranking, respectively. Together, they capture how well the shape, scale, and order of contributions match between the two methods.

The results, shown in Table 5.7, highlight a clear difference between the two datasets. In the synthetic VAR dataset, the distributed contributions are very close to the Shapley values across all metrics. Cosine similarity reaches 0.940, and the Spearman correlation is 0.733, indicating that both the overall contribution pattern and the relative importance of each agent are accurately captured. The MAE is low, meaning the actual values are also close.

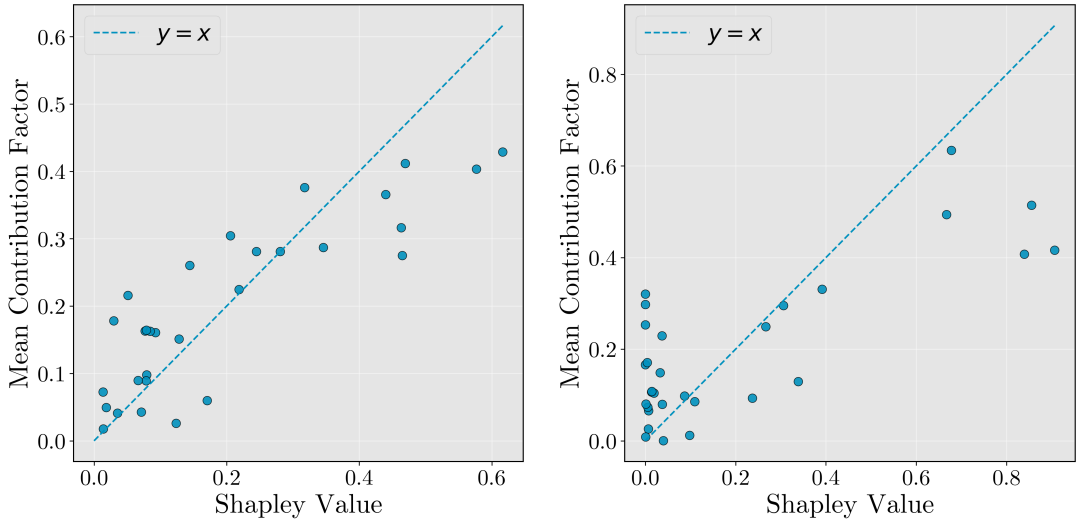
In the Nord Pool dataset, the comparison is less precise, which is expected given the complexity of real data. A cosine similarity of 0.855 indicates that the relative distribution of contributions

across agents is largely preserved. However, the higher L1 distance and lower Spearman score indicate that there are some differences in the actual values and their relative order.

Table 5.7: Evaluation metrics comparing the distributed contribution values with Shapley-based values from a centralized approach, for both the Nord Pool and VAR datasets.

Metric	**Nord Pool**	**VAR**
Mean Absolute Error (MAE)	0.138	0.0749
L1 Distance	0.690	0.375
Cosine Similarity	0.855	0.940
Spearman Correlation	0.376	0.733

Moreover, Figure 5.5 provides a visual comparison between the contribution factors estimated by the distributed system and the Shapley values computed in the centralized setting. In the VAR dataset, the estimated contributions exhibit a strong alignment with the reference values. This visual coherence confirms the numerical results presented above, indicating that the distributed system can accurately capture the relative importance of each agent. By comparison, the Nord Pool dataset presents a more dispersed distribution, reflecting its increased complexity, where agent contributions are harder to isolate and estimate precisely. This difference also reflects how the two methods handle redundancy. Shapley values promote cooperation by rewarding similar features equally, even if only one is used in the final model. In contrast, the proposed method compensates only agents whose data is directly used for prediction, as encouraged by the sparsity of the LASSO regularization. This distinction is particularly relevant in the VAR dataset, where the absence of redundancy leads both approaches to yield similar contribution estimates.



(a) VAR Dataset (b) Nord Pool Dataset

Figure 5.5: Comparison of mean contribution factors and Shapley values for the VAR and Nord Pool datasets. The dashed line shows the $y = x$ reference, where both values would match exactly.

To further assess the behavior of the reward system, an additional experiment was performed

using the VAR dataset. In this case, instead of estimating the model, predictions were generated using the original coefficient matrix B that was used to construct the dataset. This approach removes the effect of estimation errors and allows for a clearer view of how the reward distribution behaves when the true model is known. This test allows for a direct comparison between the contribution factors obtained from the estimated model and those computed with the true coefficients. If the reward distribution remains consistent across both scenarios, it indicates that the intermediate stages of the system do not introduce errors that affect fairness.

The results of this experiment are summarized in Table 5.8, which compares the distributed contributions against Shapley values for both versions of the VAR dataset. Additionally, Figure 5.6 provides a visual comparison of the contribution factors obtained using the true B matrix.

Table 5.8: Evaluation metrics comparing the distributed contribution values with Shapley-based values from a centralized approach, for both experiments with the VAR dataset.

Metric	**VAR (estimated)**	**VAR (true)**
Mean Absolute Error (MAE)	0.0749	0.0758
L1 Distance	0.375	0.379
Cosine Similarity	0.940	0.939
Spearman Correlation	0.733	0.733

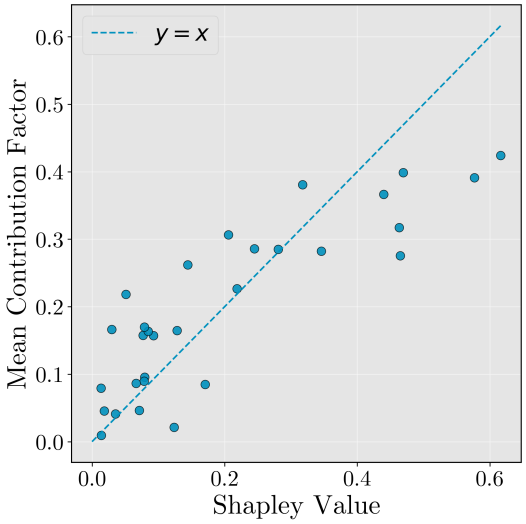


Figure 5.6: Comparison between mean contribution factors and Shapley values for the VAR dataset using the true coefficient matrix.

As shown in both the table and the figure, the results obtained using the true coefficients are very similar to those from the estimated model. This suggests that the model was able to recover the underlying contribution structure present in the data with reasonable accuracy. While it does not fully confirm that the reward system is free from error, it shows that using estimated coefficients does not significantly distort the reward distribution in this case.

Chapter 6

Conclusion

This work set out to create a distributed system to enable collaboration between energy producers while ensuring privacy and fairness. The goal was to allow multiple agents to jointly improve their forecasting capabilities without disclosing private data or relying on centralized authorities. A key challenge addressed was how to motivate and sustain cooperation in competitive environments, where individual benefits may be unevenly distributed.

6.1 Summary and Contributions

The system developed in this thesis builds upon an existing privacy-preserving framework based on the LASSO-VAR model, which supports decentralized model training and forecasting. While the original system ensured confidentiality, it failed to address the problem of incentivizing participation, making its adoption more difficult in real-world competitive settings.

To address this gap, the main contributions of this thesis were the design of an incentive mechanism and the development of validation strategies to ensure honest behavior:

Incentive Mechanism A contribution-based payment model was introduced to allocate rewards according to each participant's impact on the final forecast. This ensures that agents are compensated fairly based on the value of their participation.

Validation Algorithms Two distinct validation strategies were proposed to prevent dishonest reporting of contributions. The first is based on perturbation analysis, detecting manipulation by observing its effect on forecast accuracy. The second relies on zero-knowledge proofs to verify the correctness of reported contributions without revealing private data.

To evaluate the proposed mechanisms, both a testing framework and a functional prototype were developed. The prototype extends the original privacy-preserving system with incentives, while the testing framework enables targeted assessments using both synthetic and real-world datasets. Together, these tools made it possible to validate the fairness and robustness of the proposed incentive mechanisms and associated validation techniques.

The results obtained from these evaluations provide direct answers to the research questions and hypothesis defined in Chapter 1. Regarding RQ1, results showed that the system is capable of delivering accurate forecasts while preserving privacy and fairly compensating participants, even under adversarial conditions. In response to RQ2, the distributed contribution metrics showed strong alignment with centralized Shapley value estimates, providing evidence for the fairness of the system.

These results support the hypothesis that a distributed forecasting system can preserve privacy, ensure fair compensation, and yet match the performance of centralized alternatives.

6.2 Future Work

Despite the promising results, several limitations remain unaddressed and point to directions for future work.

Firstly, the validation strategies proposed in this thesis may be extended. In particular, zero-knowledge proofs could be further explored by assessing their performance in broader settings and by investigating alternative constructions or implementation strategies. There is also potential to combine both validation strategies. Perturbations help detect the presence of manipulation, while zero-knowledge proofs ensure that submitted values remain consistent. Used together, these techniques could verify that the reported inputs are consistent and correspond to the estimated model that yields the best predictive performance.

Moreover, while dishonest behavior can be detected, this work does not address how such situations should be handled. Currently, there is no enforcement mechanism to determine what happens once manipulation is identified. Future work should consider how to define and implement enforcement policies, such as withholding rewards or invalidating forecasts, and evaluate how these choices affect the system's reliability and fairness.

Another area for improvement is the degree of decentralization. Although the developed work supports decentralized training and preserves data privacy, a central coordinator was still used for validation and payment. This role could be replaced by extending the use of blockchain, which would remove the need for the validator. Smart contracts can handle payment distribution, contribution aggregation, and the logic of the validation algorithms. Executing these components on-chain would make the system public, transparent, and auditable, allowing all participants to verify how rules are enforced and thereby improving fairness. This design fits naturally with ZKPs, which are already widely supported in blockchain platforms and can be embedded in smart contracts to enable decentralized, verifiable validation without relying on a trusted coordinator.

A further extension concerns the valuation of contributions, which in the current system is based on a fixed fee parameter. A more flexible approach would allow this value to be dynamic, reflecting context-specific factors. Implementing such a system would require mechanisms for agents to negotiate or agree on prices, possibly using market-based or auction-driven models.

Finally, blockchain transactions could be optimized by batching payments per agent instead of issuing one transaction per forecast. This would significantly reduce transaction fees and make the system more efficient.

References

[1] Amirhossein Ahmadi, Mohammad Talaei, Masod Sadipour, Ali Moradi Amani, and Mahdi Jalili. Deep Federated Learning-Based Privacy-Preserving Wind Power Forecasting. *IEEE Access*, 11:39521–39530, 2023. Conference Name: IEEE Access.

[2] Amal Alshardan, Sidra Tariq, Rab Nawaz Bashir, Oumaima Saidani, and Rashid Jahangir. Federated Learning (FL) Model of Wind Power Prediction. *IEEE Access*, 12:129575–129586, 2024. Conference Name: IEEE Access.

[3] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Succinct non-interactive zero knowledge for a von neumann architecture. In *Proceedings of the 23rd USENIX Security Symposium*, pages 781–796. USENIX Association, 2014.

[4] Ricardo Bessa, Carlos Moreira, Bernardo Silva, and Manuel Matos. Handling renewable energy variability and uncertainty in power systems operation. *WIREs Energy and Environment*, 3(2):156–178, March 2014.

[5] Ricardo J. Bessa, Artur Trindade, and Vladimiro Miranda. Spatial-Temporal Solar Power Forecasting for Smart Grids. *IEEE Transactions on Industrial Informatics*, 11(1):232–241, February 2015.

[6] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *Advances in Cryptology – CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 103–112. Springer, 1988.

[7] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[8] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, October 1988.

[9] Christopher Briggs, Zhong Fan, and Peter Andras. Federated Learning for Short-Term Residential Load Forecasting. *IEEE Open Access Journal of Power and Energy*, 9:573–583, 2022. Conference Name: IEEE Open Access Journal of Power and Energy.

[10] M. Cao, Q. Wang, and Q. Wang. Federated learning in smart home: A dynamic contract-based incentive approach with task preferences. *Computer Networks*, 249, 2024.

[11] R.V. Cherukuri, G. Lavanya Devi, and N. Ramesh. Federated learning with blockchain-based model aggregation and incentives. *International Journal of Computers and Applications*, 46(6):407–417, 2024.

[12] Fabio Coelho, Filipe Silva, Carla Goncalves, Ricardo Bessa, and Ana Alonso. A Blockchain-based Data Market for Renewable Energy Forecasts. In *2022 Fourth International Conference on Blockchain Computing and Applications (BCCA)*, pages 297–304, San Antonio, TX, USA, September 2022. IEEE.

[13] European Commission. Renewable energy statistics. https://ec.europa.eu/eurostat/databrowser/view/nrg_ind_ren/default/table?lang=en, 2025. Accessed: 2025-06-29.

[14] Sizheng Fan, Hongbo Zhang, Yuchen Zeng, and Wei Cai. Hybrid Blockchain-Based Resource Trading System for Federated Learning in Edge Computing. *IEEE Internet of Things Journal*, 8(4):2252–2264, February 2021. Conference Name: IEEE Internet of Things Journal.

[15] Muhammad Firdaus, Harashta Tatimma Larasati, and Kyung-Hyune Rhee. A Secure Federated Learning Framework using Blockchain and Differential Privacy. In *2022 IEEE 9th International Conference on Cyber Security and Cloud Computing (CSCloud)/2022 IEEE 8th International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, pages 18–23, Xi'an, China, June 2022. IEEE.

[16] Tianxing Fu, Jia Hu, Geyong Min, and Zi Wang. Zero-Knowledge Proof-Based Consensus for Blockchain-Secured Federated Learning, March 2025. arXiv:2503.13255 [cs].

[17] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(3):690–728, July 1991.

[18] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The Knowledge Complexity of Interactive Proof-Systems. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, pages 291–304. ACM, 1985.

[19] Carla Gonçalves, Ricardo J Bessa, and Pierre Pinson. A critical overview of privacy-preserving approaches for collaborative forecasting. *International journal of Forecasting*, 37(1):322–342, 2021.

[20] Carla Goncalves, Ricardo J. Bessa, and Pierre Pinson. Privacy-Preserving distributed learning for renewable energy forecasting. *IEEE Transactions on Sustainable Energy*, 12(3):1777–1787, 3 2021.

[21] Carla Goncalves, Pierre Pinson, and Ricardo J. Bessa. Towards data markets in renewable energy forecasting. *IEEE Transactions on Sustainable Energy*, 12(1):533–542, 7 2020.

[22] W. Guo, Y. Wang, and P. Jiang. Incentive mechanism design for Federated Learning with Stackelberg game perspective in the industrial scenario. *Computers and Industrial Engineering*, 184, 2023.

[23] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated Learning for Mobile Keyboard Prediction, February 2019. arXiv:1811.03604 [cs].

[24] Vikas Hassija, Vaibhav Chawla, Vinay Chamola, and Biplab Sikdar. Incentivization and Aggregation Schemes for Federated Learning Applications. *IEEE Transactions on Machine Learning in Communications and Networking*, 1:185–196, 2023. Conference Name: IEEE Transactions on Machine Learning in Communications and Networking.

[25] István Hegedűs, Gábor Danner, and Márk Jelasity. Gossip Learning as a Decentralized Alternative to Federated Learning. In José Pereira and Laura Ricci, editors, *Distributed Applications and Interoperable Systems*, volume 11534, pages 74–90. Springer International Publishing, Cham, 2019. Series Title: Lecture Notes in Computer Science.

[26] Neveen Mohammad Hijazi, Moayad Aloqaily, Mohsen Guizani, Bassem Ouni, and Fakhri Karray. Secure Federated Learning With Fully Homomorphic Encryption for IoT Communications. *IEEE Internet of Things Journal*, 11(3):4289–4300, February 2024. Conference Name: IEEE Internet of Things Journal.

[27] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 603–618, Dallas Texas USA, October 2017. ACM.

[28] Tao Hong, Pierre Pinson, Shu Fan, Hamidreza Zareipour, Alberto Troccoli, and Rob J. Hyndman. Probabilistic energy forecasting: Global Energy Forecasting Competition 2014 and beyond. *International Journal of Forecasting*, 32(3):896–913, July 2016.

[29] Gangqiang Hu, Jianmin Han, Jianfeng Lu, Juan Yu, Sheng Qiu, Hao Peng, Donglin Zhu, and Taiyong Li. Game-Theoretic Design of Quality-Aware Incentive Mechanisms for Hierarchical Federated Learning. *IEEE Internet of Things Journal*, 11(15):26033–26045, August 2024. Conference Name: IEEE Internet of Things Journal.

[30] Jiawen Kang, Zehui Xiong, Dusit Niyato, Shengli Xie, and Junshan Zhang. Incentive Mechanism for Reliable Federated Learning: A Joint Optimization Approach to Combining Reputation and Contract Theory. *IEEE Internet of Things Journal*, 6(6):10700–10714, December 2019. Conference Name: IEEE Internet of Things Journal.

[31] Jiawen Kang, Zehui Xiong, Dusit Niyato, Yuze Zou, Yang Zhang, and Mohsen Guizani. Reliable Federated Learning for Mobile Networks. *IEEE Wireless Communications*, 27(2):72–80, April 2020.

[32] Harsh Kasyap, Arpan Manna, and Somanath Tripathy. An Efficient Blockchain Assisted Reputation Aware Decentralized Federated Learning Framework. *IEEE Transactions on Network and Service Management*, 20(3):2771–2782, September 2023. Conference Name: IEEE Transactions on Network and Service Management.

[33] Sungwook Kim. Incentive Design and Differential Privacy Based Federated Learning: A Mechanism Design Perspective. *IEEE Access*, 8:187317–187325, 2020. Conference Name: IEEE Access.

[34] Ryan Lavin, Xuekai Liu, Hardhik Mohanty, Logan Norman, Giovanni Zaarour, and Bhaskar Krishnamachari. A Survey on the Applications of Zero-Knowledge Proofs, August 2024. arXiv:2408.00243 [cs].

[35] Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854, November 2020.

[36] Li Li, Xi Yu, Xuliang Cai, Xin He, and Yanhong Liu. Contract-Theory-Based Incentive Mechanism for Federated Learning in Health CrowdSensing. *IEEE Internet of Things Journal*, 10(5):4475–4489, March 2023. Conference Name: IEEE Internet of Things Journal.

[37] Wei Yang Bryan Lim, Jianqiang Huang, Zehui Xiong, Jiawen Kang, Dusit Niyato, Xian-Sheng Hua, Cyril Leung, and Chunyan Miao. Towards Federated Learning in UAV-Enabled Internet of Vehicles: A Multi-Dimensional Contract-Matching Approach. *IEEE Transactions on Intelligent Transportation Systems*, 22(8):5140–5154, 2021.

[38] Haizhou Liu, Xuan Zhang, Xinwei Shen, and Hongbin Sun. A Fair and Efficient Hybrid Federated Learning Framework based on XGBoost for Distributed Power Prediction, January 2022. arXiv:2201.02783 [cs].

[39] Jiawei Liu, Guopeng Zhang, Kezhi Wang, and Kun Yang. Task-Load-Aware Game-Theoretic Framework for Wireless Federated Learning. *IEEE Communications Letters*, 27(1):268–272, January 2023. Conference Name: IEEE Communications Letters.

[40] Jing Liu, Yang Xiao, Shuhui Li, Wei Liang, and C. L. Philip Chen. Cyber Security and Privacy Issues in Smart Grids. *IEEE Communications Surveys & Tutorials*, 14(4):981–997, 2012.

[41] Yuan Liu, Wangyuan Yu, Zhengpeng Ai, Guangxia Xu, Liang Zhao, and Zhihong Tian. A Blockchain-Empowered Federated Learning in Healthcare-Based Cyber Physical Systems. *IEEE Transactions on Network Science and Engineering*, 10(5):2685–2696, September 2023. Conference Name: IEEE Transactions on Network Science and Engineering.

[42] H. Lyu, Y. Zhang, C. Wang, S. Long, and S. Guo. Federated learning privacy incentives: Reverse auctions and negotiations. *CAAI Transactions on Intelligence Technology*, 8(4):1538–1557, 2023.

[43] Zeba Mahmood and Vacius Jusas. Implementation Framework for a Blockchain-Based Federated Learning Model for Classification Problems. *Symmetry*, 13(7):1116, June 2021.

[44] Wuxing Mao, Qian Ma, Guocheng Liao, and Xu Chen. Game Analysis and Incentive Mechanism Design for Differentially Private Cross-Silo Federated Learning. *IEEE Transactions on Mobile Computing*, 23(10):9337–9351, October 2024. Conference Name: IEEE Transactions on Mobile Computing.

[45] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data, January 2023. arXiv:1602.05629 [cs].

[46] Aristeidis Mystakidis, Paraskevas Koukaras, Nikolaos Tsalikidis, Dimosthenis Ioannidis, and Christos Tjortjis. Energy Forecasting: A Comprehensive Review of Techniques and Technologies. *Energies*, 17(7):1662, March 2024.

[47] Akarsh K Nair, Sinem Coleri, Jayakrushna Sahoo, Linga Reddy Cenkeramaddi, and Ebin Deni Raj. Incentivized Federated Learning: A Survey. *IEEE Transactions on Emerging Topics in Computational Intelligence*, pages 1–20, 2025.

[48] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Technical report, Bitcoin.org, May 2009. White paper.

[49] Dinh C. Nguyen, Ming Ding, Pubudu N. Pathirana, Aruna Seneviratne, Jun Li, and H. Vincent Poor. Federated Learning for Internet of Things: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*, 23(3):1622–1658, 2021.

[50] Dinh C. Nguyen, Quoc-Viet Pham, Pubudu N. Pathirana, Ming Ding, Aruna Seneviratne, Zihuai Lin, Octavia Dobre, and Won-Joo Hwang. Federated Learning for Smart Healthcare: A Survey. *ACM Computing Surveys*, 55(3):1–37, March 2023.

[51] K. Ouyang, J. Yu, X. Cao, and Z. Liao. Towards Reliable Federated Learning Using Blockchain-Based Reverse Auctions and Reputation Incentives. *Symmetry*, 15(12), 2023.

[52] Jinlong Pang, Jieling Yu, Ruiting Zhou, and John C.S. Lui. An Incentive Auction for Heterogeneous Client Selection in Federated Learning. *IEEE Transactions on Mobile Computing*, 22(10):5733–5750, October 2023. Conference Name: IEEE Transactions on Mobile Computing.

[53] Jaehyoung Park and Hyuk Lim. Privacy-Preserving Federated Learning Using Homomorphic Encryption. *Applied Sciences*, 12(2):734, January 2022.

[54] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. Privacy-Preserving Deep Learning via Additively Homomorphic Encryption. *IEEE Transactions on Information Forensics and Security*, 13(5):1333–1345, May 2018.

[55] Jiahao Qi, Feilong Lin, Zhongyu Chen, Changbing Tang, Riheng Jia, and Minglu Li. High-Quality Model Aggregation for Blockchain-Based Federated Learning via Reputation-Motivated Task Participation. *IEEE Internet of Things Journal*, 9(19):18378–18391, October 2022. Conference Name: IEEE Internet of Things Journal.

[56] Minfeng Qi, Ziyuan Wang, Shiping Chen, and Yang Xiang. A Hybrid Incentive Mechanism for Decentralized Federated Learning. *Distrib. Ledger Technol.*, 1(1):4:1–4:15, September 2022.

[57] Pian Qi, Diletta Chiaro, Antonella Guzzo, Michele Ianni, Giancarlo Fortino, and Francesco Piccialli. Model aggregation techniques in federated learning: A comprehensive survey. *Future Generation Computer Systems*, 150:272–293, January 2024.

[58] Mubashir Husain Rehmani, Martin Reisslein, Abderrezak Rachedi, Melike Erol-Kantarci, and Milena Radenkovic. Integrating Renewable Energy Resources Into the Smart Grid: Recent Developments in Information and Communication Technologies. *IEEE Transactions on Industrial Informatics*, 14(7):2814–2825, July 2018.

[59] Yuris Mulya Saputra, Diep N. Nguyen, Dinh Thai Hoang, Thang X. Vu, Eryk Dutkiewicz, and Symeon Chatzinotas. Federated Learning Meets Contract Theory: Economic-Efficiency Framework for Electric Vehicle Networks. *IEEE Transactions on Mobile Computing*, 21(8):2803–2817, 2022.

[60] Zahraa Khduair Taha, Chong Tak Yaw, Siaw Paw Koh, Sieh Kiong Tiong, Kumaran Kadirgama, Foo Benedict, Jian Ding Tan, and Yogendra AL Balasubramaniam. A Survey of Federated Learning From Data Perspective in the Healthcare Domain: Challenges, Methods, and Future Directions. *IEEE Access*, 11:45711–45735, 2023.

[61] Chenchen Tan, Xinghao Li, Longxiang Gao, Tom H. Luan, Youyang Qu, Yong Xiang, and Rongxing Lu. Digital Twin Enabled Remote Data Sharing for Internet of Vehicles: System and Incentive Design. *IEEE Transactions on Vehicular Technology*, 72(10):13474–13489, October 2023. Conference Name: IEEE Transactions on Vehicular Technology.

[62] Ming Tang, Fu Peng, and Vincent W.S. Wong. A Blockchain-Empowered Incentive Mechanism for Cross-Silo Federated Learning. *IEEE Transactions on Mobile Computing*, 23(10):9240–9253, October 2024. Conference Name: IEEE Transactions on Mobile Computing.

[63] Xiaoli Tang and Han Yu. Efficient Large-Scale Personalizable Bidding for Multiagent Auction-Based Federated Learning. *IEEE Internet of Things Journal*, 11(15):26518–26530, August 2024. Conference Name: IEEE Internet of Things Journal.

[64] Julija Tastu, Pierre Pinson, Pierre-Julien Trombe, and Henrik Madsen. Probabilistic Forecasts of wind power Generation Accounting for geographically dispersed information. *IEEE Transactions on Smart Grid*, 5(1):480–489, 1 2014.

[65] Tra Huong Thi Le, Nguyen H. Tran, Yan Kyaw Tun, Minh N. H. Nguyen, Shashi Raj Pandey, Zhu Han, and Choong Seon Hong. An Incentive Mechanism for Federated Learning in Wireless Cellular Networks: An Auction Approach. *IEEE Transactions on Wireless Communications*, 20(8):4874–4887, August 2021. Conference Name: IEEE Transactions on Wireless Communications.

[66] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. Data Poisoning Attacks Against Federated Learning Systems, August 2020. arXiv:2007.08432 [cs].

[67] Xuezhen Tu, Kun Zhu, Nguyen Cong Luong, Dusit Niyato, Yang Zhang, and Juan Li. Incentive Mechanisms for Federated Learning: From Economic and Game Theoretic Perspective. *IEEE Transactions on Cognitive Communications and Networking*, 8(3):1566–1593, September 2022. Conference Name: IEEE Transactions on Cognitive Communications and Networking.

[68] T. Wan, T. Jiang, W. Liao, and N. Jiang. Hierarchical Incentive Mechanism for Federated Learning: A Single Contract to Dual Contract Approach for Smart Industries. *International Journal of Intelligent Systems*, 2024, 2024.

[69] Quyuan Wang, Songtao Guo, Guiyan Liu, Li Yang, and Chengsheng Pan. MotiLearn: Contract-Based Incentive Mechanism for Heterogeneous Edge Collaborative Training. *IEEE Transactions on Network Science and Engineering*, 9(4):2895–2909, July 2022. Conference Name: IEEE Transactions on Network Science and Engineering.

[70] Siyang Wang, Haitao Zhao, Wanli Wen, Wenchao Xia, Bin Wang, and Hongbo Zhu. Contract Theory Based Incentive Mechanism for Clustered Vehicular Federated Learning. *IEEE Transactions on Intelligent Transportation Systems*, 25(7):8134–8147, July 2024. Conference Name: IEEE Transactions on Intelligent Transportation Systems.

[71] Tianhao Wang, Johannes Rausch, Ce Zhang, Ruoxi Jia, and Dawn Song. A Principled Approach to Data Valuation for Federated Learning. In Qiang Yang, Lixin Fan, and Han Yu, editors, *Federated Learning*, volume 12500, pages 153–167. Springer International Publishing, Cham, 2020. Series Title: Lecture Notes in Computer Science.

[72] Xiaofei Wang, Yunfeng Zhao, Chao Qiu, Zhicheng Liu, Jiangtian Nie, and Victor C. M. Leung. InFEDge: A Blockchain-Based Incentive Mechanism in Hierarchical Federated Learning for End-Edge-Cloud Communications. *IEEE Journal on Selected Areas in Communications*, 40(12):3325–3342, December 2022. Conference Name: IEEE Journal on Selected Areas in Communications.

[73] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. Beyond Inferring Class Representatives: User-Level Privacy Leakage From Federated Learning. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pages 2512–2520, Paris, France, April 2019. IEEE.

[74] Zhilin Wang, Qin Hu, Ruinian Li, Minghui Xu, and Zehui Xiong. Incentive Mechanism Design for Joint Resource Allocation in Blockchain-Based Federated Learning. *IEEE Transactions on Parallel and Distributed Systems*, 34(5):1536–1547, May 2023. Conference Name: IEEE Transactions on Parallel and Distributed Systems.

[75] T. Wen, H. Zhang, H. Zhang, H. Wu, D. Wang, X. Liu, W. Zhang, Y. Wang, and S. Cao. RTIFed: A Reputation based Triple-step Incentive mechanism for energy-aware Federated learning over battery-constricted devices. *Computer Networks*, 241, 2024.

[76] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. Technical report, Ethereum Foundation, 2014.

[77] Leijie Wu, Song Guo, Zicong Hong, Yi Liu, Wenchao Xu, and Yufeng Zhan. Long-Term Adaptive VCG Auction Mechanism for Sustainable Federated Learning With Periodical Client Shifting. *IEEE Transactions on Mobile Computing*, 23(5):6060–6073, May 2024. Conference Name: IEEE Transactions on Mobile Computing.

[78] Zhibo Xing, Zijian Zhang, Meng Li, Jiamou Liu, Liehuang Zhu, Giovanni Russello, and Muhammad Rizwan Asghar. Zero-Knowledge Proof-based Practical Federated Learning on Blockchain, April 2023. arXiv:2304.05590 [cs].

[79] H. Xu, P. Nanda, J. Liang, and X. He. FCH, an incentive framework for data-owner dominated federated learning. *Journal of Information Security and Applications*, 76, 2023.

[80] Yin Xu, Mingjun Xiao, Haisheng Tan, An Liu, Guoju Gao, and Zhaoyang Yan. Incentive Mechanism for Differentially Private Federated Learning in Industrial Internet of Things. *IEEE Transactions on Industrial Informatics*, 18(10):6927–6939, October 2022. Conference Name: IEEE Transactions on Industrial Informatics.

[81] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated Machine Learning: Concept and Applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2):1–19, March 2019.

[82] Ruizhe Yang, Tonghui Zhao, F. Richard Yu, Meng Li, Dajun Zhang, and Xuehui Zhao. Blockchain-Based Federated Learning With Enhanced Privacy and Security Using Homomorphic Encryption and Reputation. *IEEE Internet of Things Journal*, 11(12):21674–21688, June 2024. Conference Name: IEEE Internet of Things Journal.

[83] X. Yang and T. Li. A Blockchain-based federated learning framework for secure aggregation and fair incentives. *Connection Science*, 36(1), 2024.

[84] Minghao Yao, Saiyu Qi, Zhen Tian, Qian Li, Yong Han, Haihong Li, and Yong Qi. Quantifying Bytes: Understanding Practical Value of Data Assets in Federated Learning. *Tsinghua Science and Technology*, 30(1):135–147, February 2025. Conference Name: Tsinghua Science and Technology.

[85] Zhenning Yi, Yutao Jiao, Wenting Dai, Guoxin Li, Haichao Wang, and Yuhua Xu. A Stackelberg Incentive Mechanism for Wireless Federated Learning With Differential Privacy. *IEEE Wireless Communications Letters*, 11(9):1805–1809, September 2022. Conference Name: IEEE Wireless Communications Letters.

[86] Xuefei Yin, Yanming Zhu, and Jiankun Hu. A Comprehensive Survey of Privacy-preserving Federated Learning: A Taxonomy, Review, and Future Directions. *ACM Computing Surveys*, 54(6):1–36, July 2022.

[87] Yulan Yuan, Lei Jiao, Konglin Zhu, and Lin Zhang. Incentivizing Federated Learning Under Long-Term Energy Constraint via Online Randomized Auctions. *IEEE Transactions on Wireless Communications*, 21(7):5129–5144, July 2022. Conference Name: IEEE Transactions on Wireless Communications.

[88] Rongfei Zeng, Chao Zeng, Xingwei Wang, Bo Li, and Xiaowen Chu. A Comprehensive Survey of Incentive Mechanism for Federated Learning, June 2021. arXiv:2106.15406 [cs].

[89] Yufeng Zhan, Jie Zhang, Zicong Hong, Leijie Wu, Peng Li, and Song Guo. A Survey of Incentive Mechanism Design for Federated Learning. *IEEE Transactions on Emerging Topics in Computing*, 10(2):1035–1044, April 2022. Conference Name: IEEE Transactions on Emerging Topics in Computing.

[90] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, March 2021.

[91] Hangjian Zhang, Yanan Jin, Jianfeng Lu, Shuqin Cao, Qing Dai, and Shasha Yang. Contribution Matching-Based Hierarchical Incentive Mechanism Design for Crowd Federated Learning. *IEEE Access*, 12:24735–24750, 2024. Conference Name: IEEE Access.

[92] Ran Zheng, Andreas Sumper, Monica Aragüés-Peñalba, and Samuel Galceran-Arellano. Advancing Power System Services With Privacy-Preserving Federated Learning Techniques: A Review. *IEEE Access*, 12:76753–76780, 2024. Conference Name: IEEE Access.

[93] Shuyuan Zheng, Yang Cao, Masatoshi Yoshikawa, Huizhong Li, and Qiang Yan. FL-Market: Trading Private Models in Federated Learning. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 1525–1534, December 2022. arXiv:2106.04384 [cs].

[94] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang. Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services*, 14(4):352–375, 2018.

[95] Juncen Zhu, Jiannong Cao, Divya Saxena, Shan Jiang, and Houda Ferradi. Blockchain-empowered Federated Learning: Challenges, Solutions, and Future Directions. *ACM Comput. Surv.*, 55(11):240:1–240:31, February 2023.

[96] Y. Zhu, Z. Liu, P. Wang, and C. Du. A dynamic incentive and reputation mechanism for energy-efficient federated learning in 6G. *Digital Communications and Networks*, 9(4):817–826, 2023.

[97] Zhaowei Zhu, Jingxuan Zhu, Ji Liu, and Yang Liu. Federated Bandit: A Gossiping Approach. *Proc. ACM Meas. Anal. Comput. Syst.*, 5(1):02:1–02:29, February 2021.

Appendix A

Systematic Review

This section outlines the systematic approach used to identify and select papers for the State of the Art review. It details a reproducible methodology designed to ensure a comprehensive and unbiased collection of relevant studies.

Search queries were derived from keywords contained in the base references of the thesis proposal. Results obtained from each information source were then screened by removing duplicates and applying the eligibility criteria defined. This process resulted in 113 publications to be considered for content assessment.

A.1 Base References

The three references mentioned in the proposal [12, 20, 21] provide a starting point for the systematic review by introducing the key concepts and themes needed to perform the study. This will help in searching for articles by providing the necessary keywords for the queries used in the following sections. The most central concerns were split into different concepts, each of which is laid out in Table A.1 along with respective similar words.

Table A.1: Search Keywords and Concepts.

Federated Learning	**Incentives**	**RES**
Distributed Learning Collaborative Learning	Profit Allocation Fair compensation Data Valuation Reward	Renewable Energy RES Forecasting

Blockchain	**Privacy-Preserving**	**Others**
Ledger	Confidential Trustless	Forecasting Decentralized

A.2 Information Sources

The literature review was conducted using three major academic databases: IEEE Xplore, ACM Digital Library, and SCOPUS.

IEEE Xplore indexes peer-reviewed publications from leading journals and conferences in engineering and computer science. It is particularly relevant to this work due to its strong coverage of topics in Informatics Engineering and Sustainable Energy. Additionally, two of the three core references for this thesis were published in IEEE journals, further supporting the choice of this database.

ACM Digital Library offers high-quality publications in the fields of Computing and Information Technology, making it a valuable source for research on privacy-preserving systems and decentralized algorithms.

SCOPUS was included to ensure broader coverage of interdisciplinary literature, especially at the intersection of energy systems, economics, and computer science. Its comprehensive indexing supports a more complete and balanced review across domains.

A.3 Search Strategy

This section reviews the methodology applied to retrieve publications from the databases mentioned above.

A.3.1 Query Strings

Queries were first devised for IEEE Xplore using the keywords in Table A.1. According to the abundance and quality of the results, these were iteratively refined. Three different query strings were created to better cover the multidisciplinary aspects of the study. These are as follows:

1. Privacy-preserving Federated Learning forecasting for RES sector
2. Usage of blockchain to distribute data sharing incentives in a Federated Learning context
3. Privacy-preserving Federated Learning with data sharing incentives

Each of the queries focuses on a subset of the topics outlined above. All concepts were structured into separate clauses containing a disjunction of the gathered synonyms. These clauses were then combined as conjunctions, according to the scope of each query.

The ACM Digital Library query string for each is detailed in Listings A.1, A.2, and A.3. Equivalent translations for other systems have been omitted for the sake of simplicity.

Listing A.1: ACM Digital Library Query String 1.

```
(Title:(Forecast*) OR Abstract:(Forecast*))
AND
```

```
(Title:("Federated Learning") OR Title:("Distributed Learning") OR
 Title:("Collaborative Learning") OR Abstract:("Federated Learning") OR
 Abstract:("Distributed Learning") OR Abstract:("Collaborative Learning"))
AND
(AllField:(Decentralized) OR AllField:(Privacy) OR
 AllField:("Privacy-Preserving"))
AND
(AllField:(Energy) OR AllField:("Renewable Energy") OR
 AllField:("Renewable Energy Source") OR AllField:(RES))
```

Listing A.2: ACM Digital Library Query String 2.

```
(AllField:(Forecast*) OR AllField:("Federated Learning") OR
 AllField:("Distributed Learning") OR AllField:(Collaborati*) OR
 AllField:("Data Market"))
AND
(Title:("Federated Learning") OR Title:("Distributed Learning") OR
 Title:(Collaborati*))
AND
(AllField:(Blockchain) OR AllField:("Ledger"))
AND
(Title:(Monetization) OR Title:(Incentive))
AND
(AllField:(Monetization) OR AllField:(Incentive))
```

Listing A.3: ACM Digital Library Query String 3.

```
(Title:("Federated Learning") OR Title:("Distributed Learning") OR
 Title:("Collaborative Learning") OR Abstract:("Federated Learning") OR
 Abstract:("Distributed Learning") OR Abstract:("Collaborative Learning"))
AND
(AllField:(Decentralized) OR AllField:(Privacy) OR
 AllField:("Privacy-Preserving") OR AllField:(Trustless))
AND
(Title:(Incentive) OR Title:("Profit Allocation") OR
 Title:("Fair Compensation") OR Title:(Reward) OR
 Abstract:("Profit Allocation") OR Abstract:("Fair Compensation") OR
 Abstract:(Reward))
```

A.3.2 Eligibility Criteria

Results obtained from the previous queries were filtered using the eligibility criteria described in Table A.2. This process was performed using the filtering capabilities of each website to their full

extent. Further scanning was done manually by reading the titles and abstracts of the remaining records, with exclusions made according to criteria IC3, EC2, and EC3.

Table A.2: Inclusion and exclusion criteria.

	Inclusion Criteria
IC1	Published in peer-reviewed journals
IC2	Published in the last 10 years (after 2014)
IC3	Published in relevant areas to the study, that is, Federated Learning, Privacy-Preservation, Blockchain, and Profit Allocation.
	Exclusion Criteria
EC1	Not published in the English language
EC2	Addresses the problem of data sharing incentives outside the scope of distributed model training
EC3	Use of Privacy-Preserving Federated Learning without data sharing incentives and out of the scope of RES forecasting

A.3.3 Results

The initial database search identified a total of 811 records. Additionally, 3 references were identified from the proposal references, bringing the total to 814.

After applying the eligibility criteria, 485 records were excluded using each source's filtering options, leaving 326 publications for duplicate removal. After these were removed, only 157 unique records were considered for eligibility. A manual assessment was then conducted by applying the inclusion and exclusion criteria in Section 2.3.2. Ultimately, the search process resulted in 113 papers considered for content analysis.

Figure A.1 displays a diagram of the process and obtained results.

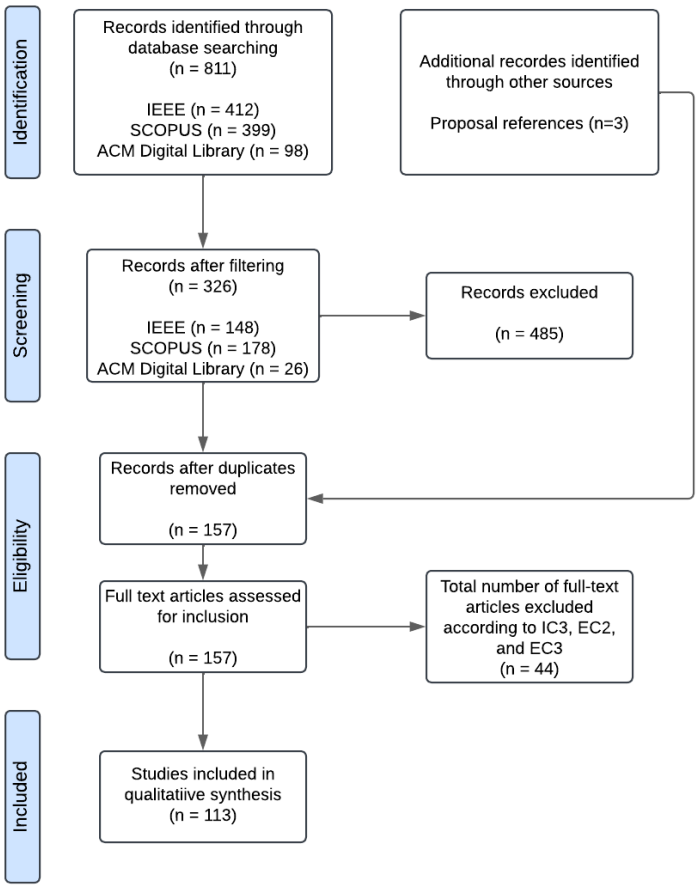


Figure A.1: Publication collections flow diagram.

Appendix B

Sequence Diagrams

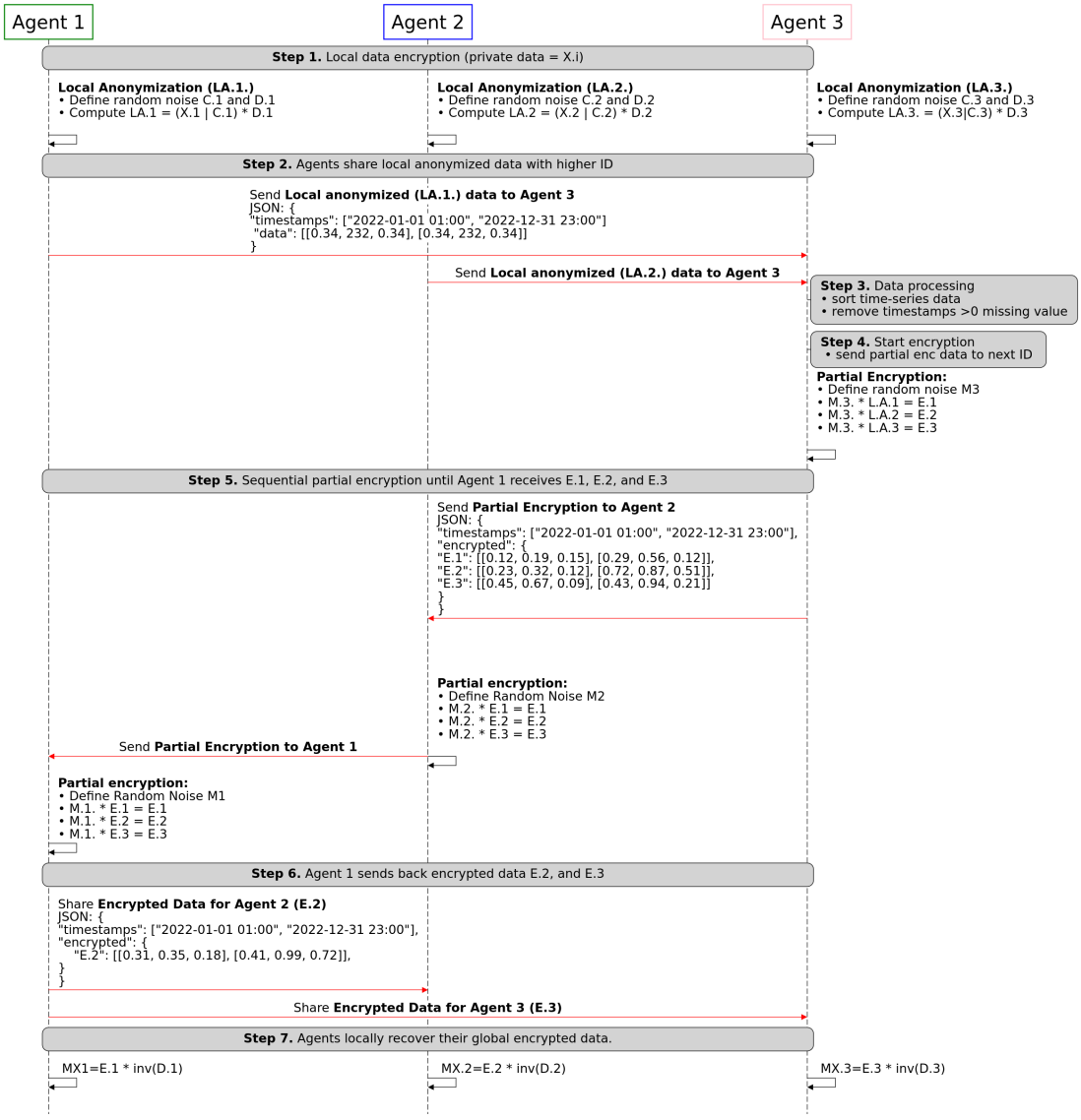


Figure B.1: Sequence diagram of the encryption protocol [20].

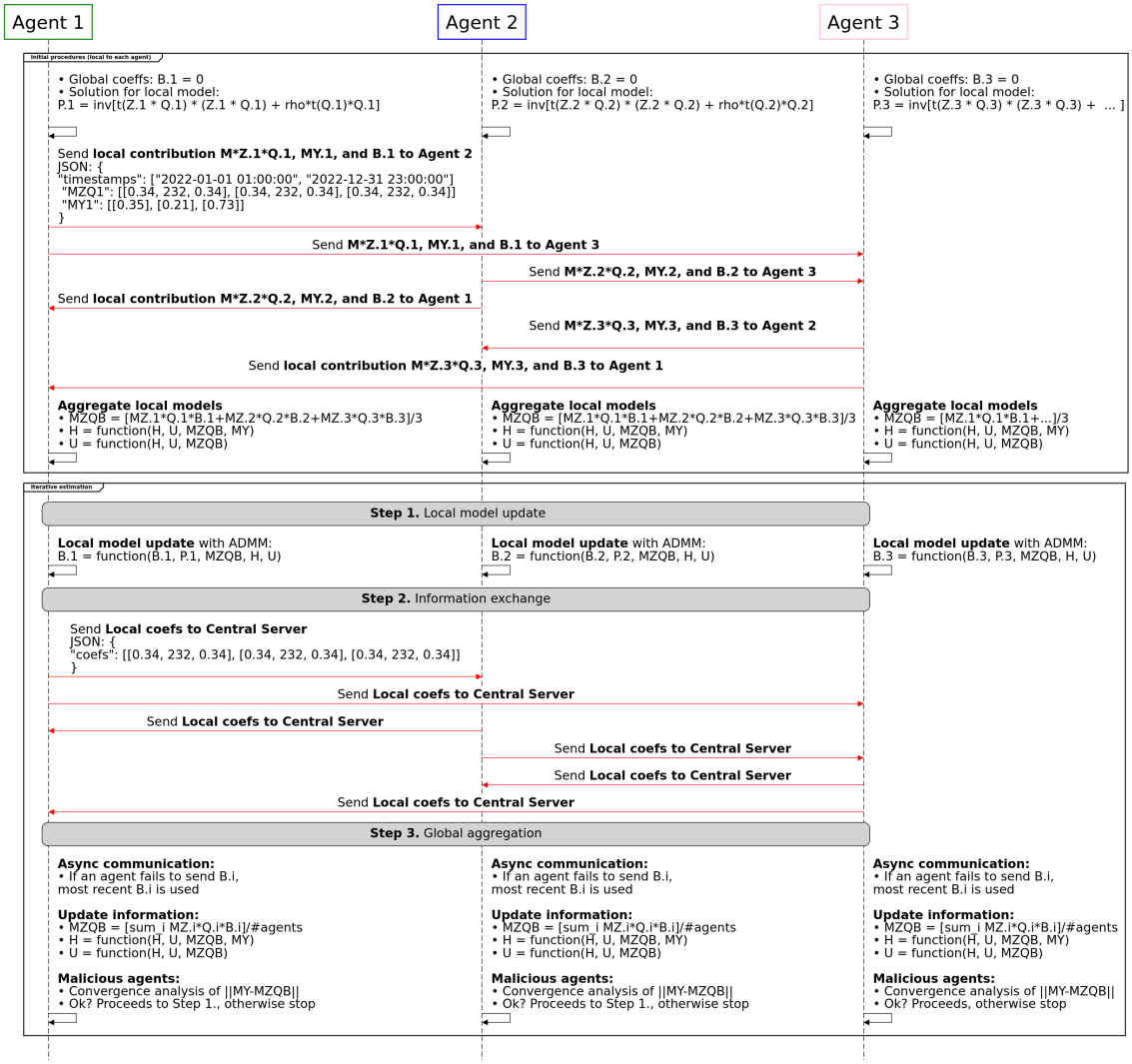


Figure B.2: Sequence diagram of the model fitting stage [20].

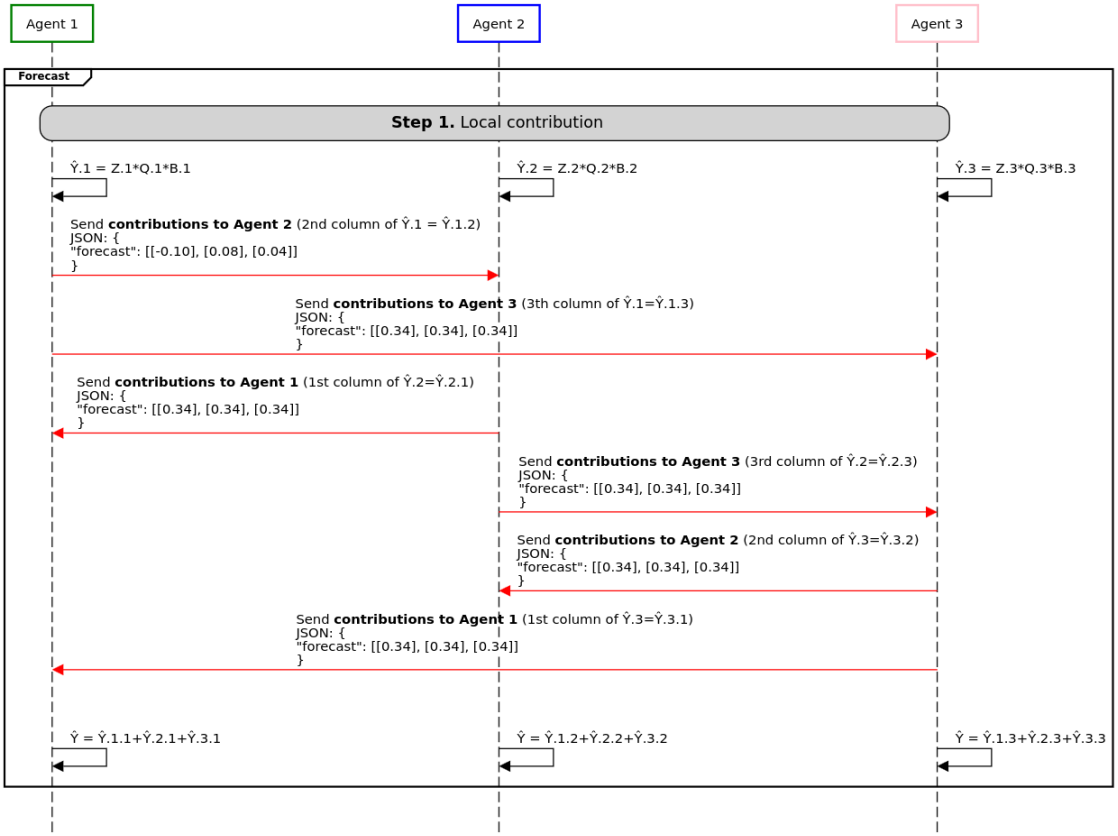
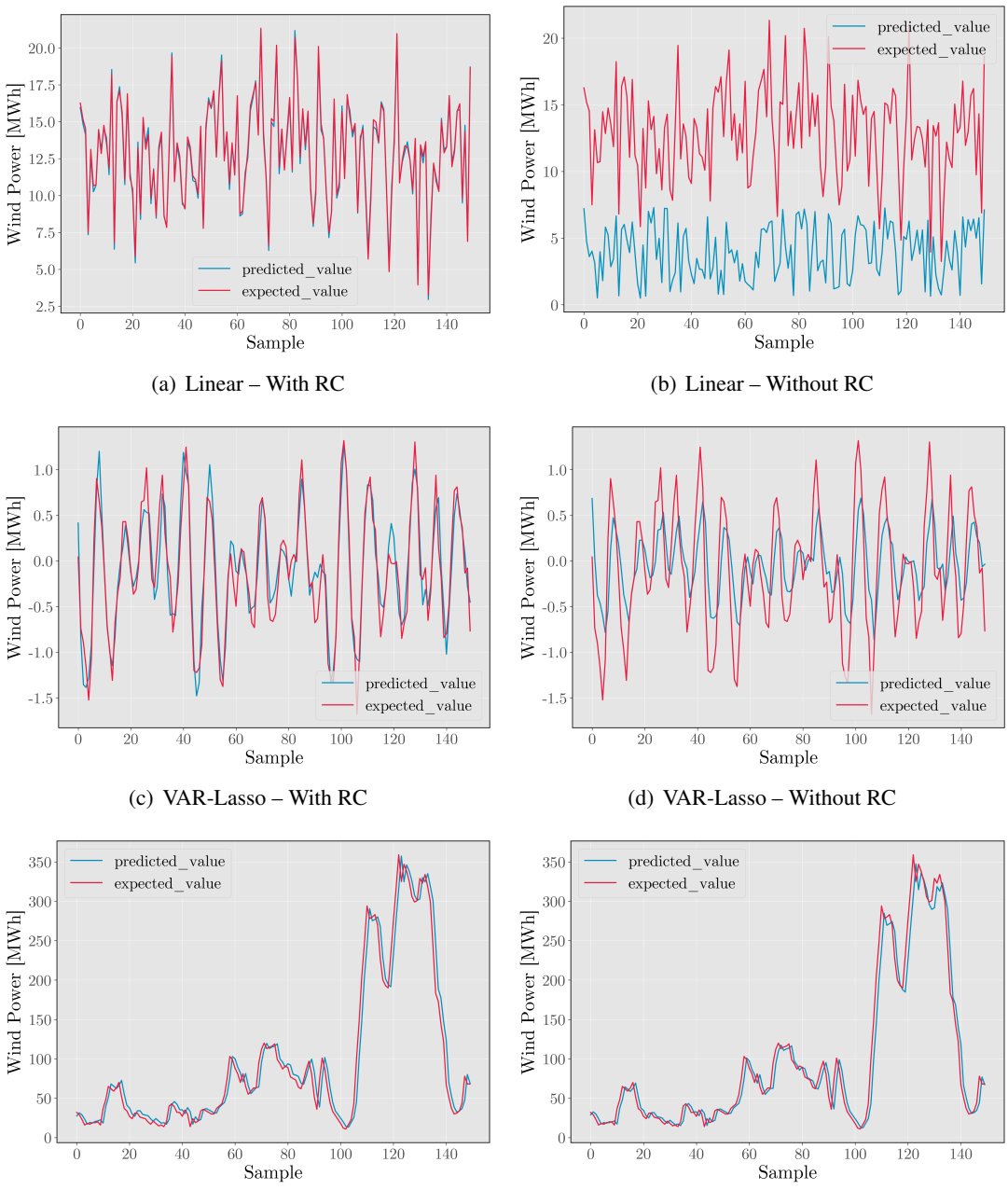


Figure B.3: Sequence diagram of the prediction process [20].

Appendix C

Results



(a) Linear – With RC

(b) Linear – Without RC

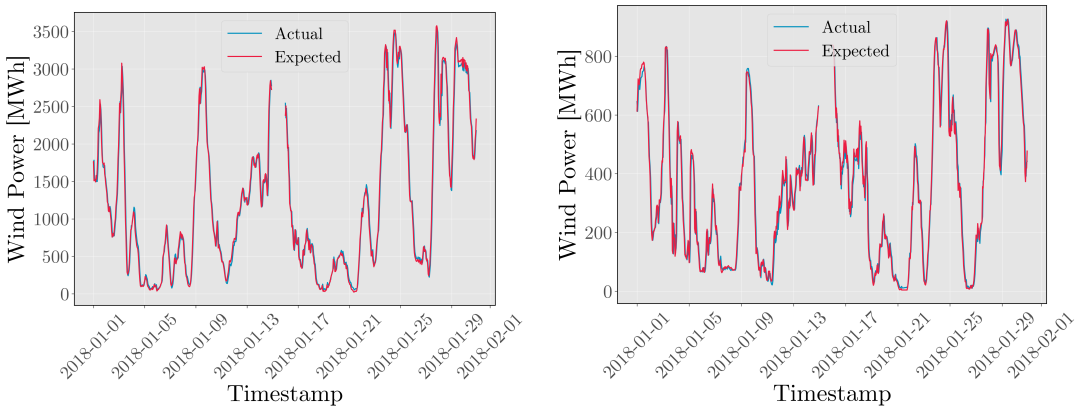
(c) VAR-Lasso – With RC

(d) VAR-Lasso – Without RC

(e) Nord Pool – With RC

(f) Nord Pool – Without RC

Figure C.1: Forecast results with and without remote contributions (RC) across all datasets. Each plot shows the first 150 samples of predicted values compared to the expected values. The results were obtained using the setup with 8,000 records, 6 agents, and one variable per agent, focusing on the predictions for agent zero (SE1).



(a) DK1 forecast

(b) DK2 forecast

(c) SE1 forecast

(d) SE2 forecast

(e) SE3 forecast

(f) SE4 forecast

Figure C.2: Forecasts generated by the prototype for each region during January 2018. Predicted values are plotted against the expected wind power measurements.