UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

DISTRIBUTED OPTIMIZATION FOR MULTI-AGENT SYSTEMS WITH COMMUNICATION DELAYS AND FAILURES IN ENERGY SYSTEMS COORDINATION

MÉMOIRE PRÉSENTÉ
COMME EXIGENCE PARTIELLE OF THE
MASTER IN ELECTRICAL ENGINEERING

PAR
MD REJA ALAM TALUKDER

SEPTEMBER 19, 2025

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

MAÎTRISE EN GÉNIE ÉLECTRIQUE

Direction de recherche :

Kodjo Agbossou — Directeur de recherche

Jury d'évaluation

Tahar Tafticht — Professeur, Université du Québec en Abitibi-Témiscamingue

Juan Carlos Oviedo Cepeda — Professeur associé, Université du Québec à Trois-Rivières

Kodjo Agbossou — Professeur, Université du Québec à Trois-Rivières

Abstract

The increasing complexity of modern energy systems, driven by the integration of distributed energy resources (DERs), necessitates robust coordination mechanisms. Multi-agent systems (MAS) offer a promising decentralized approach, but their effectiveness is challenged by communication failures and agent non-responsiveness. This thesis addresses the critical problem of maintaining distributed coordination in multi-agent energy systems when some agents become non-responsive due to communication issues or device faults.

We propose a novel framework that integrates the Alternating Direction Method of Multipliers (ADMM) for distributed energy consumption coordination with a machine learning (ML)-based forecasting strategy. This hybrid approach enables the central coordinator to estimate the energy consumption of non-responsive agents using pre-trained ML models, thereby preserving a consistent optimization signal and ensuring system stability.

Extensive simulation studies demonstrate the framework's effectiveness. Under ideal communication conditions, the ADMM-based coordination significantly reduces peak electricity demand, smooths load profiles, and minimizes collective energy costs. In scenarios with agent non-responsiveness, the ML-based forecasting mechanism allows the system to maintain robust operation, particularly when the fraction of disconnected agents is moderate. While performance degrades with increasing non-responsiveness, the hybrid system proves resilient, highlighting the practical applicability of predictive intelligence in decentralized energy management. This research advances the field by providing a practical solution for fault-tolerant coordination in smart grids, enhancing their scalability and reliability.

Résumé

La complexité croissante des systèmes énergétiques modernes, due à l'intégration des ressources énergétiques distribuées (RED), nécessite des mécanismes de coordination robustes. Les systèmes multi-agents (SMA) offrent une approche décentralisée prometteuse, mais leur efficacité est mise à l'épreuve par les pannes de communication et la non-réactivité des agents. Cette thèse aborde le problème critique du maintien de la coordination distribuée dans les systèmes énergétiques multi-agents lorsque certains agents deviennent non réactifs en raison de problèmes de communication ou de défaillances de dispositifs.

Nous proposons un nouveau cadre qui intègre la méthode des directions alternées des multiplicateurs (ADMM) pour la coordination distribuée de la consommation d'énergie avec une stratégie de prévision basée sur l'apprentissage automatique (ML). Cette approche hybride permet au coordinateur central d'estimer la consommation d'énergie des agents non réactifs à l'aide de modèles ML pré-entraînés, préservant ainsi un signal d'optimisation cohérent et assurant la stabilité du système.

Des études de simulation approfondies démontrent l'efficacité du cadre. Dans des conditions de communication idéales, la coordination basée sur l'ADMM réduit considérablement la demande de pointe en électricité, lisse les profils de charge et minimise les coûts énergétiques collectifs. Dans les scénarios de non-réactivité des agents, le mécanisme de prévision basé le ML permet au système de maintenir un fonctionnement robuste, en particulier lorsque la fraction d'agents déconnectés est modérée. Bien que les performances se dégradent avec l'augmentation de la non-réactivité, le système hybride s'avère résilient, soulignant l'applicabilité pratique de l'intelligence prédictive dans la gestion décentralisée de l'énergie. Cette recherche

fait progresser le domaine en fournissant une solution pratique pour la coordination tolérante aux pannes dans les réseaux intelligents, améliorant leur évolutivité et leur fiabilité.

Acknowledgments

I am deeply grateful to my research director, Professor Kodjo Agbossou, from the Department of Electrical and Computer Engineering, for his trust in me with this project and for his unwavering encouragement and guidance throughout the research process. My sincere thanks also go to Dr. Nilson Henao, from the same department at UQTR, for his invaluable contributions and steadfast support during my research work.

I would like to express my heartfelt appreciation to Dr. Farshad Etedadi Aliabadi, who provided crucial assistance in my research and significantly contributed to the completion of my master's studies. Finally, I extend my gratitude to everyone who has contributed, directly or indirectly, to the successful completion of this research.

Contents

List of Tables

List of Figures

List of Abbreviations

AA Aggregator Agent

ACH Air Changes per Hour

ADMM Alternating Direction Method of Multipliers

A2A Agent-to-Agent

ARIMA Auto-Regressive Integrated Moving Average

AutoML Automated Machine Learning

CAD Canadian Dollar

DER Distributed Energy Resource

DERs Distributed Energy Resources

DR Demand Response

ECS Energy Consumption Scheduler

GAN Generative Adversarial Network

HEMS Home Energy Management Systems

HVAC Heating, Ventilation, and Air Conditioning

IoT Internet of Things

LSTM Long Short-Term Memory

M2M Machine-to-Machine

MAS Multi-Agent Systems

ML Machine Learning

OpenADR Open Automated Demand Response

P2P Peer-to-Peer

PVGIS Photovoltaic Geographical Information System

RA Residential Agent

RC Residential Customers

RFR Random Forest Regression

SCADA Supervisory Control and Data Acquisition

SEP Smart Energy Profile

SGD Stochastic Gradient Descent

SI International System of Units

TE Transactive Energy

TES Transactive Energy System

SG Smart Grid

DNP3 Distributed Network Protocol 3

IEC 61850 International Electrotechnical Commission 61850

IEEE 2030.5 Institute of Electrical and Electronics Engineers 2030.5

MSE Mean Squared Error

R^2 R-squared

PAR Peak-to-Average Ratio

Nomenclature

α	Parameter adjusting sensitivity of the total cost function
Δt	Time step
δ_i^k	Indicator variable for agent i data transmission at iteration k
δ_t	Signal indicating energy consumption prioritization
$\delta_{\max}$	Maximum occupant sensitivity
δ_k^i	Occupant sensitivity weighting factor for household i at time step k
$\hat{x}_i^k$	Forecast value for agent i at iteration k (intermittent loss)
$\hat{x}_j$	Forecast value for unresponsive agent j (complete non-responsiveness)
λ	Dual variable / Price signal from coordinator (general)
λ_k	Dual variable (Lagrange multiplier) for time step k
$\mathbf{u}^i$	Vector of energy consumption for household i over time horizon T
$\mathbf{w}^i$	Vector of building thermal characteristics for household i
$\mathcal{C}$	Convex set describing global constraints
$\mathcal{P}_{\mathcal{C}}(\cdot)$	Projection onto the convex set $\mathcal{C}$
Π	Total price of energy set by the market
π	Energy price per unit consumed (general)

π_k Price signal received from the coordinator for time step k

ρ ADMM penalty parameter

$\text{dayofyear}_{\text{cos}}$ Cosine component of day of year

$\text{dayofyear}_{\text{sin}}$ Sine component of day of year

hour_{cos} Cosine component of hour of day

hour_{sin} Sine component of hour of day

$\tilde{x}_i^k$ Substituted variable for agent i at iteration k

$c(y)$ Cost function representing the cost of providing energy

$C_i(u_i)$ Local cost function for agent i

C_k^i Energy expense of agent i at time step k

$C_{\text{total},k}$ Total energy cost at time step k

$D_i(u_i)$ Discomfort function for agent i

$f_i(x_i)$ Convex objective function for agent i

$g(\cdot)$ Function characterizing household thermal dynamics

$J(\mathbf{u}^i)$ Household welfare function for household i

N Total number of agents in the system (used in chapters)

n Total number of agents in the system (general)

n_i Dimension of the local decision variable for agent i

$R^i(\lambda, \mathbf{u}^i)$ Financial incentives for household i

T Time horizon

$U(u_k^i)$ Occupant comfort utility function for household i

$u_i(t)$ Energy consumption of agent i at time t (general)

u_i^k Scaled dual variable for agent i at iteration k

u_k^i Total energy consumption of household i at time step k

$u_{\max}^i$ Maximum energy consumption for household i

$u_{a,k}^i$ Fixed load of household i at time step k

$u_{h,k}^i$ Flexible thermal load of household i at time step k

w Profit made by the aggregator in the transactive energy system

x Global decision variable (centralized formulation)

x_i Local decision variable for agent i

x_k^i Indoor temperature of household i at time step k

x_k^{out} Outdoor temperature at time step k

x_{comf}^i Occupant's preferred indoor temperature for household i

$x_{\max}^i$ Maximum indoor temperature for household i

$x^i_{\min}$ Minimum indoor temperature for household i

y Aggregator's energy consumption

y^* Optimal energy consumption minimizing global objective function

z Common (global) decision variable (consensus formulation)

z_k Global variable for consensus energy consumption at time step k

Chapter 1 - Introduction

The global energy sector is undergoing a profound transformation, driven by the urgent need for decarbonization in response to climate change [1]. This energy transition necessitates a shift from fossil fuels to a system dominated by renewable energy sources like solar and wind. However, the intermittent and distributed nature of these resources presents significant challenges to grid stability and reliability. The Smart Grids emerges as the essential enabling infrastructure for this transition. By integrating advanced digital communication, sensing, and control technologies into the power grid, it provides the intelligence required to manage renewable variability, coordinate distributed resources, and empower consumers, thereby paving the way for a sustainable, reliable, and economically viable energy future [2].

The traditional power grid's evolution into a Smart Grid is driven by the large-scale integration of Distributed Energy Resources (DERs), including renewable generation, energy storage, and responsive loads. This marks a critical shift from a centralized, top-down control architecture to a decentralized system of autonomous, interacting entities. Managing the complexity and dynamism of this new landscape requires advanced operational paradigms capable of efficiently and reliably balancing supply and demand in real-time [3].

An emerging paradigm for enabling this decentralized control is Transactive Energy (TE) [4]. The GridWise Architecture Council defines TE as "a system of economic and control mechanisms that enables dynamic supply-demand balance across electrical infrastructure using value as a key operational parameter." Within TE frameworks, individual residences and buildings act as rational, autonomous agents, making decisions about energy consumption, generation, and storage based on economic signals like dynamic pricing. This approach promises to optimize energy distribution, reduce costs, and prevent grid overloads by leveraging machine-to-machine (M2M) communication for automated negotiations [5].

The successful coordination of a high number of autonomous agents within a TE system cannot be handled by a central authority due to scalability and latency issues. Therefore, Distributed Optimization has emerged as the essential methodology for achieving system-wide objectives. Algorithms such as the Alternating Direction Method of Multipliers (ADMM) are employed to decompose a massive, global optimization problem into smaller, localized sub-problems that each agent can solve independently. Agents iteratively share information to converge towards a solution that is optimal for the entire system, ensuring stability and efficiency without a central controller.

However, the efficacy of distributed optimization critically depends on reliable, near-real-time communication among all participating agents. In real-world networks, communication failures such as data loss, delays, and network outages are inevitable. While intermittent delays can degrade performance, a far more critical failure occurs when agents become completely non-responsive. The problem of missing agents, specifically those that stop transmitting data and responding to coordination signals, can corrupt the entire optimization process. The absence of their data leads to inaccurate calculations, ineffective control signals, and can ultimately destabilize the supply-demand balance, undermining the very foundation of the transactive energy system [6].

This thesis directly addresses this critical vulnerability. We develop and evaluate a novel, fault-tolerant framework that enhances the resilience of distributed optimization in transactive energy systems. The core contribution is the integration of machine learning-based imputation directly into the ADMM coordination mechanism [7]. By training predictive models on historical data, our framework accurately forecasts the energy behavior of missing or non-responsive agents in real-time. These forecasts act as surrogates, allowing the distributed optimization process to continue uninterrupted and maintain system-wide coordination and stability, even in the face of significant communication failures.

1.1 Research Infrastructure

The research presented in this thesis was conducted within the *Laboratoire d'Intelligence et de Recherche sur les Énergies Intégrées (LIREI) at the Université du Québec à Trois-Rivières (UQTR). The work heavily relied on the computational resources provided by the laboratory, which include a high-performance computing (HPC) cluster essential for training the machine learning models and running the extensive simulations required for this study. The software ecosystem included Python, with libraries such as scikit-learn, TensorFlow, and Dask for parallel computing, which were instrumental in developing and evaluating the proposed framework.*

1.2 Problem Statement

In decentralized energy systems that rely on distributed optimization, the coordination process is critically dependent on continuous communication from all participating agents. While minor data delays can be managed, the system's integrity is severely compromised by complete agent non-responsiveness. This occurs when agents go offline for prolonged periods due to hardware malfunctions, network outages, or users opting out of demand response programs [8].

When an agent becomes non-responsive, a standard coordination mechanism would fail. The absence of consumption data from the offline agent corrupts the aggregate load calculation, leading the coordinator to generate an inaccurate and ineffective price signal for the remaining, active agents [9]. This breakdown in coordination prevents the system from achieving its primary goals, such as minimizing peak demand and reducing energy costs.

The central challenge, therefore, is not to restore communication with the offline agent, but to enable the responsive agents to continue coordinating effectively around the non-responsive ones. This requires an intelligent coordinator that can accurately estimate the behavior of the missing agents. The problem thus becomes one of robust, real-time data imputation. The solution must leverage machine learning (ML) models that can reliably forecast the energy consumption profiles of non-responsive agents by using predictive inputs, specifically temporal features (e.g., time of day, day of year) and exogenous variables like outdoor weather data. This forecast is then used by the coordinator to compute a corrected global price signal, ensuring the continued, efficient operation of the multi-agent system [10].

1.3 Objectives

1. Design a fault-tolerant coordination framework, based on the Alternating Direction Method of Multipliers (ADMM), where a central coordinator actively manages agent non-responsiveness by integrating forecasted data into the optimization loop.

2. Develop and implement a machine learning pipeline to train and deploy agent-specific forecasting models for data imputation. These models will predict

the energy consumption of non-responsive agents using temporal features and external weather variables as inputs.

3. Evaluate the framework's resilience by simulating scenarios with a varying scale of non-responsive agents. The evaluation will quantify the system's ability to preserve coordinated benefits, such as peak load reduction, compared to an ideal communication baseline.

1.4 Contributions

This thesis introduces novel methodologies for addressing communication failures in multi-agent systems (MAS) for energy coordination. The key contributions include:

- Developing a fault-tolerant coordination framework based on ADMM that integrates machine learning-based imputation for managing agent non-responsiveness.

- Implementing a machine learning pipeline for training and deploying agent-specific forecasting models to predict energy consumption of non-responsive agents.

- Evaluating the framework's resilience through simulations with varying scales of non-responsive agents, quantifying its ability to preserve coordinated benefits like peak load reduction.

These advancements significantly improve the scalability, efficiency, and reliability of energy systems, paving the way for practical implementations of robust and adaptive smart grids.

1.5 Thesis Organization

This thesis is structured into five chapters:

- **Chapter 1: Introduction:** Provides an overview of the research background, motivation, problem statement, objectives, and the key contributions of this thesis.
- **Chapter 2: Literature Review:** Reviews existing research on multi-agent systems, distributed optimization, and strategies for addressing communication delays and failures, and identifies gaps addressed in this work.
- **Chapter 3: Methodology:** Details the proposed system model, including mathematical formulations, optimization techniques, and coordination protocols to mitigate communication challenges.
- **Chapter 4: Simulation and Results:** Presents the experimental setup, evaluates the proposed methods under various scenarios, and discusses the findings in comparison to existing approaches.
- **Chapter 5: Conclusion and Future Work:** Summarizes the contributions, discusses limitations, and outlines potential directions for further research.

Chapter 2 - Literature Review

This chapter provides a critical review of the research landscape to contextualize this challenge and justify the methodologies proposed in this thesis. The review will first examine the foundational role of Multi-Agent Systems in energy coordination and the application of distributed optimization techniques. It will then survey the various coordination mechanisms and specific strategies developed to handle agent disconnections and communication failures. This analysis culminates in a gap analysis that highlights the limitations of current approaches, specifically the insufficient handling of missing data, the underutilization of machine learning for fault recovery, and the lack of integrated optimization-recovery models. By establishing this gap, this review demonstrates the clear need for a new framework that enhances the resilience of distributed energy systems, which this thesis aims to provide.

2.1 Multi-agent systems in energy coordination

The modern energy grid is rapidly moving away from its traditional, monolithic structure of centralized generation and passive consumption. The influx of distributed energy resources (DERs), such as solar panels, wind turbines, battery storage, and responsive loads, has created a complex, dynamic, and bi-directional ecosystem. This new reality fundamentally challenges the efficacy of centralized control, which faces insurmountable hurdles related to computational burden, communication bottlenecks, and a single point of failure. To manage this decentralized landscape, a paradigm shift in control and coordination is not just beneficial, but essential.

2.1.1 Overview of Multi-Agent Systems

Multi-agent systems (MAS) represent a distributed computational paradigm where multiple autonomous agents collaborate to achieve individual and collective goals [11]. An agent is typically defined as an independent software entity with the capability to perceive its environment, make decisions, and act upon those decisions to meet specific objectives. In MAS, agents can communicate and cooperate with each other, making them particularly suitable for complex, dynamic environments, such as energy systems, robotics, and logistics [12]. A fundamental feature of MAS is their ability to exhibit emergent behavior, where the overall system behavior arises from local interactions between agents. This property enables MAS to effectively manage distributed systems by decentralizing control and optimizing decision-making processes. MAS architectures are commonly categorized into three main types: centralized, decentralized, and hybrid. Figure 2.1.1 illustrates a distributed architecture, highlighting the interaction between residential agents (RA) and aggregator agents (AA). In centralized architectures, a single aggregator agent (AA) manages and coordinates the actions of all connected residential agents (RAs). This structure simplifies system management but poses scalability limitations as the number of connected agents increases. The centralized approach is suitable for applications requiring strict control and oversight but may suffer from bottlenecks and single points of failure as the network scales.

Conversely, decentralized architectures distribute decision-making authority across individual residential agents (RAs). Each agent operates independently while maintaining communication with other agents to ensure system-wide coherence. This approach enhances scalability and robustness, making it ideal for dynamic and

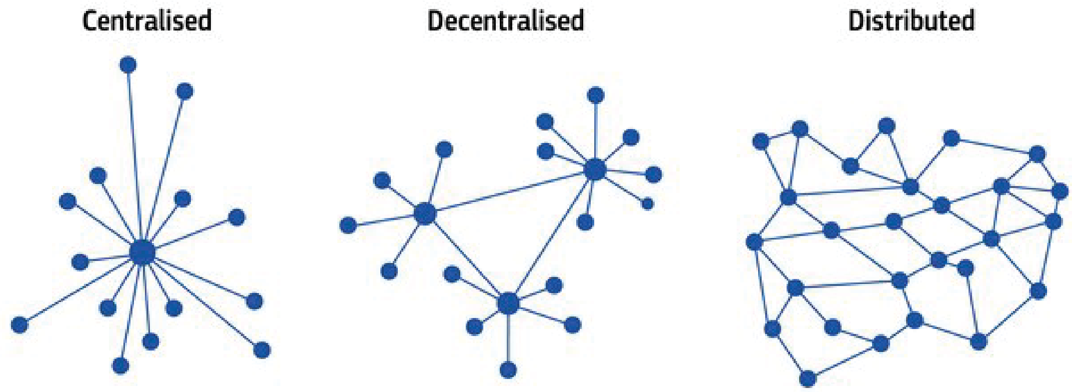

Figure 2.1.1 Multi-Agent System Topologies

distributed environments. Decentralized MAS are particularly effective in energy systems, where localized decision-making can optimize resource utilization and resilience. Hybrid architectures combine these two approaches, leveraging centralized control for high-level decision-making while delegating local autonomy to individual agents. This structure enables MAS to balance the strengths of both centralized and decentralized systems, providing scalability and flexibility while maintaining overall coordination [12]. By categorizing MAS architectures into centralized, decentralized, and hybrid systems, it becomes evident how their design influences scalability, efficiency, and resilience in distributed systems. These insights are critical for designing MAS tailored to specific application domains such as energy coordination.

2.1.2 Applications in Energy Systems

Multi-agent systems (MAS) have found extensive applications in energy systems, including smart grids, microgrids, and renewable energy integration. These systems capitalize on the capabilities of autonomous agents to manage complex, distributed energy networks effectively. In smart grids, MAS enable real-time monitoring, load balancing, and energy trading by coordinating the actions of energy producers, consumers, and storage devices. For instance, agents in a smart grid can optimize energy flow based on demand, generation, and price signals, ensuring efficient and cost-effective operations [13]. In microgrids, MAS facilitate decentralized control, allowing local energy systems to operate independently or in coordination with the main grid. This capability is particularly useful for remote areas or during grid outages. Agents in microgrids optimize resource allocation, such as scheduling battery charging and discharging or adjusting renewable energy output, to maintain stability and reliability. Renewable energy integration is another critical application area for MAS. By coordinating diverse energy resources such as solar panels and wind turbines, MAS mitigate the challenges posed by the intermittent and unpredictable nature of renewable energy. Agents dynamically adjust energy distribution and storage to match fluctuating supply and demand, ensuring grid stability and maximizing the use of green energy [14]. These applications demonstrate the transformative potential of MAS in addressing the complexities of modern energy systems, paving the way for smarter, more resilient, and sustainable energy networks.

2.2 Centralized vs. Distributed Optimization

The paradigm shift towards distributed optimization is best understood when contrasted with the traditional, centralized approach. In a centralized architecture, a single entity is responsible for gathering all system-wide information, solving the entire optimization problem, and issuing commands to all subordinate agents. While this method can achieve true optimality in smaller, well-defined systems, its practicality diminishes rapidly as the grid becomes more complex. The reliance on a central controller creates significant bottlenecks in communication and computation, limits scalability, and introduces a single point of failure that can jeopardize the entire system.

In response to these limitations, distributed optimization offers a fundamentally different architecture. It disperses decision-making authority across multiple autonomous agents, enabling parallel computation and enhancing system flexibility and resilience. The choice between these two opposing philosophies involves critical trade-offs. Table 2.2.1 presents a detailed comparison across these criteria, highlighting the distinct advantages that make distributed methods uniquely suited for the dynamic and decentralized nature of modern energy systems.

Table 2.2.1 Comparison of Centralized and Distributed Optimization Approaches in Energy Systems.

Aspect	**Centralized Optimization**	**Distributed Optimization**
Definition	A single central entity solves the entire optimization problem and makes decisions for the system.	Multiple agents solve local optimization problems and coordinate to achieve a global solution.
Scalability	Limited scalability due to the computational and communication burden on the central entity.	High scalability, as computation is distributed across multiple agents [14].
Resilience	Prone to single points of failure; if the central entity fails, the entire system is affected.	More resilient to failures, as local agents can continue operating independently in the absence of some agents [15].
Communication	Requires extensive communication with all nodes, leading to potential bottlenecks and delays in large-scale systems.	Reduces communication overhead by limiting interactions to local neighbors or peers [16].
Flexibility	Less flexible to changes in system configuration or dynamic environments.	Highly flexible, as agents can adapt locally to dynamic conditions [17].
Computational Efficiency	May become computationally expensive for large systems, as the central entity must process all data.	Computationally efficient for large systems, as local agents perform parallel computations [15].
Examples	Optimal power flow in small grids with minimal agents.	Energy management in microgrids with multiple distributed energy resources [14].

2.3 Distributed optimization techniques

Distributed optimization techniques are determinant in managing the complexity of modern energy systems. By enabling agents to solve local optimization problems while coordinating to achieve global objectives, these techniques ensure scalability, efficiency, and resilience. In distributed energy systems, algorithms such as the Alternating Direction Method of Multipliers (ADMM) and consensus-based methods are widely used. ADMM facilitates the decomposition of large-scale optimization problems into smaller, manageable sub-problems, allowing agents to compute locally and exchange minimal information with neighbors [15]. Consensus-based optimization methods further enhance the robustness of multi-agent systems (MAS). These techniques enable agents to iteratively update their solutions based on local information and communication with neighboring agents, ensuring global agreement without the need for centralized control. For example, in a microgrid, distributed optimization techniques are used to balance energy supply and demand while considering constraints such as network capacity and renewable energy variability [14]. Recent advancements have introduced learning-based distributed optimization, where agents use machine learning to predict energy demand or generation patterns, further improving decision-making processes. These techniques significantly reduce computation time and enhance system adaptability to dynamic conditions. Distributed optimization techniques thus serve as a cornerstone for the efficient operation of MAS in energy systems, enabling real-time coordination and adaptability in increasingly complex and distributed networks.

2.3.1 Decomposition Methods

Decomposition methods like the Alternating Direction Method of Multipliers (ADMM) are essential for solving large-scale optimization problems in distributed energy systems. They enhance scalability and efficiency by breaking problems into smaller, manageable tasks solvable in parallel and ensuring flexibility in dynamic environments. The following table provides a detailed description.

Table 2.3.1 Comparison of decomposition methods and their relevance

Method	**Description**	**Relevance in Energy Systems**
Alternating Direction Method of Multipliers (ADMM) [18]	A decomposition-coordination algorithm that breaks large optimization problems into smaller subproblems. It solves subproblems independently and combines results using a consensus variable [15].	Highly relevant for distributed optimization in energy systems, such as distributed control in microgrids and optimal power flow problems.
Dual Decomposition [14]	A technique that decomposes the dual problem into smaller subproblems that can be solved in parallel [14].	Useful for scenarios with separable constraints, enabling parallel computation of optimization tasks across multiple agents in energy systems.
Primal-Dual Interior Point Methods [19]	An iterative approach that solves optimization problems by simultaneously updating primal and dual variables.	Applicable for solving convex optimization problems in energy systems, such as economic dispatch and demand-side management [19].
Lagrangian Relaxation [20]	A method that relaxes constraints to simplify complex problems into manageable subproblems.	Widely used in transmission network optimization and renewable energy integration [21].
Benders Decomposition [21]	A method that partitions problems into master and subproblems, iteratively refining solutions.	Applicable in multi-agent systems for energy coordination, especially in multi-stage optimization problems like generation scheduling.

2.3.2 Gradient-Based Methods

- **Distributed Gradient Descent:** This is a widely-used optimization technique where agents iteratively compute gradients based on local data and exchange updates to achieve a global consensus. It is commonly applied in training machine learning models across distributed systems and optimizing decentralized energy management.

- **Stochastic Gradient Descent (SGD):** A variant of gradient descent that processes smaller data batches, enabling faster computation and adaptation in dynamic environments. It is frequently used in distributed learning systems and energy demand forecasting.

- **Accelerated Gradient Methods:** Techniques such as Nesterov's acceleration improve convergence rates by incorporating momentum terms. These methods are applied in large-scale optimization problems, such as smart grid resource allocation and supply-demand balancing.

2.4 Coordination mechanisms

Coordination mechanisms are fundamental in multi-agent systems (MAS) to ensure agents work collectively toward achieving system-wide goals. Mechanisms such as the Contract Net Protocol facilitate high-level communication and task allocation among distributed agents, promoting efficiency and control in problem-solving scenarios [22]. Similarly, consensus algorithms, such as those utilizing nearest-neighbor rules, are widely used for coordinating autonomous agents, especially in dynamic environments [23]. Auction-based systems have also been adopted for energy management in smart grids, enabling efficient resource allocation and fostering decentralized decision-making [17].

2.4.1 Consensus Algorithms

Consensus algorithms are essential for achieving agreement among agents in multi-agent systems (MAS). Protocols such as average consensus allow agents to converge to a shared value, which makes them valuable for applications such as distributed control and sensor fusion [23]. Weighted consensus protocols extend this concept by assigning varying influence levels to agents, which are particularly useful in energy coordination and resource prioritization [24]. These algorithms also form the basis of distributed optimization, enabling agents to solve constrained problems collaboratively [14].

2.4.2 Game-Theoretic Approaches

Game theory models agent interactions and decision-making by providing a framework to analyze strategic scenarios in which agents make decisions to maximize their payoffs. It defines rules, strategies, and outcomes for interactions, and helps to predict equilibrium states where no agent can unilaterally improve their position, such as the Nash equilibrium [25].

In a smart grid, multiple energy producers (agents) generate electricity and sell it to consumers or other agents. Through mechanisms such as auction-based trading (a game-theoretic approach), agents iteratively adjust their strategies until they reach an equilibrium [26].

- **Agents:** Energy producers and consumers.
- **Strategies:** Producers decide how much energy to produce and at what price;

consumers decide how much energy to buy based on price and need.

- **Payoff:** For producers, the payoff is profit (revenue minus cost) for consumers, it is utility (value derived minus expenditure).

2.4.3 Auction-Based Mechanisms

Auction-based mechanisms are effective tools for resource allocation in multi-agent systems, enabling agents to submit bids and compete for resources based on predefined rules [27]. Common strategies include first-price auctions, where the highest bidder pays their bid amount, and Vickrey auctions, which encourage truthful bidding by charging the second-highest bid [28]. These approaches are widely used in energy systems for applications such as energy trading, grid optimization, and dynamic pricing, ensuring efficient and equitable resource allocation.

2.5 Communication challenges

Communication challenges in distributed systems include issues such as delays, packet losses, and bandwidth limitations, which can degrade the performance of distributed optimization and coordination. Synchronization issues due to delays often lead to outdated information exchange, affecting consensus protocols. Furthermore, network disruptions and failures can impact the reliability of multi-agent systems, necessitating robust fault-tolerant communication strategies.

2.5.1 Types of Communication Delays

Communication delays in distributed systems can be categorized into several types based on their characteristics and sources:

- **Fixed Delays:** These are consistent, predictable delays caused by static factors such as propagation time or processing overhead in the communication network. Fixed delays are easier to manage in time-sensitive systems.

- **Random Delays:** These are unpredictable delays caused by factors like congestion, packet loss, or retransmissions. They introduce uncertainty in system performance and affect synchronization among distributed agents.

- **Variable Delays:** These delays change dynamically based on network conditions, such as fluctuating bandwidth or competing traffic. Variable delays are common in large-scale networks and require adaptive mechanisms for effective management.

2.5.2 Communication Failures

Communication failures in distributed systems can significantly impact system performance and reliability. These failures often arise due to underlying causes such as hardware malfunctions, network congestion, transmission errors, or environmental factors. They can be categorized into several types based on their characteristics and sources:

- **Network Outages:** Disruptions in connectivity caused by hardware failures, network congestion, or maintenance. Effects include loss of communication among agents, delayed decisions, and reduced system reliability.

- **Packet Losses:** Loss of data packets due to congestion, faulty links, or transmission errors. Impacts include degraded synchronization, prolonged convergence time, and instability in distributed protocols.

- **Faulty Nodes or Links:** Failures in individual nodes or links causing partial system breakdowns. Leads to incomplete information exchange and compromised decision-making.

2.5.3 Impact on System Performance

Delays and communication failures significantly degrade the performance of distributed systems by disrupting optimization, coordination, and overall system reliability. Optimization algorithms, such as distributed gradient descent or ADMM, rely on timely data exchange, and delays can cause slower convergence or divergence from optimal solutions [14]. Coordination protocols, including consensus algorithms, are especially vulnerable to failures, as missing or delayed information prevents agents from reaching agreement or making synchronized decisions [29]. Furthermore, communication disruptions affect system reliability, making distributed systems more prone to inefficiencies, such as redundant computations or energy wastage, in dynamic environments.

2.6 Strategies for handling agent disconnections

Effective strategies for managing agent disconnections are critical in distributed multi-agent systems to ensure system stability and reliability. Redundancy mechanisms create alternative communication paths, which allows uninterrupted operations even when agents disconnect temporarily. Dynamic reconnection protocols help reintegrate disconnected agents by re-establishing synchronization with the overall network. Furthermore, predictive approaches that leverage machine learning techniques anticipate potential disconnections and enable agents to proactively adjust their coordination strategies.

2.6.1 Fault-Tolerant Design

Fault-tolerant design mechanisms are essential for ensuring that distributed systems, including multi-agent systems (MAS), remain operational even in the presence of failures. Redundant agent architectures are often employed to duplicate critical components or functionalities, which mitigates the impact of individual agent failures. Consensus reconfiguration protocols dynamically adjust the decision-making process to exclude failed agents while preserving coordination. Additionally, fault detection and recovery algorithms use real-time monitoring to identify failures and restore functionality by reallocating tasks or resources among active agents.

2.6.2 Redundancy Techniques

Redundancy techniques are a cornerstone of reliability in distributed systems, which ensures continued operation even when individual components fail. Redundant paths in communication networks provide alternative routes for data transmission, which reduces the impact of network failures. Similarly, backup agents replicate critical functionalities, stepping in seamlessly if primary agents become unavailable. Lastly, redundant information storage ensures data consistency and availability across distributed systems, which mitigates risks of data loss due to node failures.

2.6.3 Adaptive Algorithms

Adaptive algorithms are critical for maintaining the performance and reliability of distributed systems in dynamic environments. These algorithms can adjust their parameters based on real-time network conditions, such as varying bandwidth or latency. Additionally, they dynamically reconfigure to accommodate changes in agent availability, which ensures continuous operation even in the presence of failures or disconnections. Techniques such as adaptive routing and load balancing have been widely used to enhance the resilience of distributed systems.

2.7 Gap analysis

The current literature on multi-agent systems (MAS) and transactive energy frameworks has primarily focused on coordination and optimization under ideal communication scenarios. However, there are significant gaps in addressing real-world challenges such as communication failures and the absence of critical agents in the system. Specifically:

- **Limited Attention to Communication Failures:** While existing studies explore coordination between residential agents (RAs) and aggregator agents (AAs), the impact of communication failures, especially missing agent data, remains underexplored. This gap highlights the need for robust methodologies to recover and compensate for missing information in dynamic energy systems.

- **Lack of Advanced Machine Learning Solutions:** Although forecasting methods are used in energy systems, few works leverage advanced machine learning models such as Long Short-Term Memory (LSTM) to estimate missing data profiles. Current approaches often rely on less adaptive models, which may fail to handle the complexity of dynamic energy interactions.

- **Integration of Recovery and Optimization Processes:** Studies frequently focus on either recovering missing data or optimizing energy coordination separately. There is a lack of integrated methodologies that ensure seamless communication recovery while simultaneously optimizing energy costs and maintaining system reliability.

Chapter 3 - Methodology

This methodology addresses communication imperfections in distributed optimization within multi-agent systems by integrating machine learning (ML) techniques into existing coordination mechanisms. The primary objective is to enhance the robustness and reliability of consensus optimization processes, especially when agents experience intermittent data loss or complete non-responsiveness. In this framework, each agent minimizes its own convex cost function while aligning with a global decision variable, which represents the collective operating point [30]. The overall optimization problem minimizes the sum of individual cost functions, subject to the constraint that each agent's local decision variable equals the shared global variable, which may also be subject to additional global constraints. The Alternating Direction Method of Multipliers (ADMM) is employed to decompose the global problem into local subproblems solvable independently by each agent [18]. Agents update their local variables by minimizing a local objective function augmented with a quadratic penalty that enforces consensus with the global variable. Subsequently, the global variable is updated by aggregating local solutions, and dual variables are adjusted to penalize discrepancies between local and global variables.

To address communication challenges, ML-based imputation methods are proposed for handling missing data:

- **Intermittent data loss:** When an agent intermittently fails to transmit its current state, an indicator is used to substitute the missing data with a forecasted value. These forecasts are generated using ML algorithms trained on historical data, capturing patterns and correlations to predict missing values accurately. This substitution ensures that the consensus process remains uninterrupted despite sporadic communication failures [24].

- **Complete non-responsiveness:** In scenarios where an agent remains unresponsive throughout the coordination process, the coordinator utilizes a forecasted profile as a surrogate for that agent's data. ML models, such as Random Forests or K-Nearest Neighbors, are employed to construct these profiles, enabling the system to maintain functionality and achieve consensus even in the absence of real-time data from certain agents [14].

By integrating ML-based imputation techniques into the ADMM framework, the methodology enhances the resilience of distributed optimization processes against communication imperfections. This integration ensures that the system can adapt to data losses and maintain reliable operation, thereby improving the overall efficiency and robustness of multi-agent coordination mechanisms.

3.1 Multi agent system

The system model is structured as a star topology, where a central coordinator interacts with a number of distributed residential agents (RAs). In this architecture, the coordinator acts as the central hub that sends control signals or pricing information to each residential agent. These signals reflect overarching grid or energy system objectives such as load balancing, demand response, or cost minimization.

Each RA then uses the received signal as an input to its local optimization problem. The RA's objective function is designed to capture its individual goals, such as minimizing energy costs, ensuring comfort, or reducing emissions, while adhering to local constraints like appliance schedules or distributed energy resource capacities. Essentially, the RA adjusts its operation (for example, by modifying heating/cooling, appliance use, or energy storage) based on both its own preferences and the coordinator's signal.

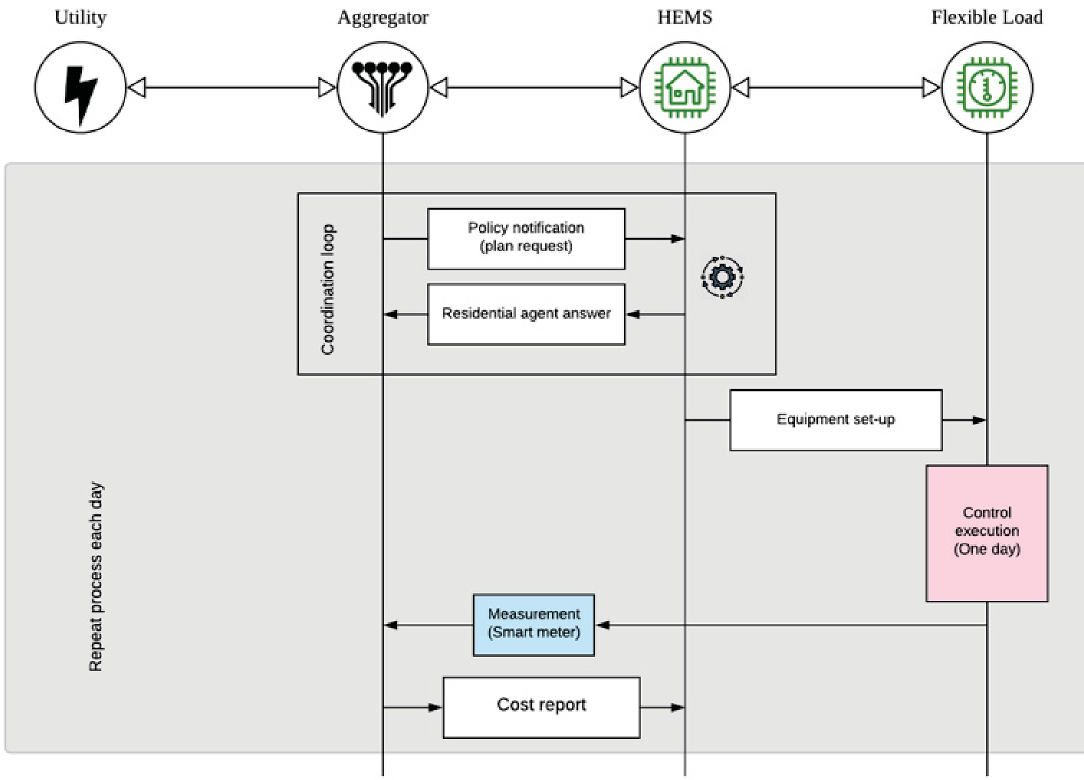

Figure 3.1.1 Sequence diagram with Central Coordinator and Residential Agents

In the context of an energy coordination problem, this model enables a seamless integration between central control and distributed decision-making. The coordinator is responsible for ensuring that the aggregate behavior of the RAs aligns with grid-level requirements, such as maintaining supply-demand balance or reducing peak loads. Meanwhile, the RAs optimize their individual operations in response to these central signals. This approach ensures that while each residential unit operates based on its own utility, the overall system performance is enhanced through coordinated energy management.

3.1.1 Individual agents

Each residential agent (RA) in the system is designed with several functional components that enable local optimization and effective communication with the central coordinator:

- **Sensing and data acquisition:** Agents are equipped with sensors and smart meters to measure energy consumption, local generation (e.g., rooftop solar panels), and storage levels (if battery systems are present). This information is critical for the local decision-making process.

- **Communication interface:** Each RA includes a communication module that allows it to receive signals or pricing information from the central coordinator. Although the star topology implies that the primary communication is between the coordinator and individual agents, some peer-to-peer information exchange may also be possible.

- **Local processing unit:** The processing unit runs local optimization algorithms. Using the signals received from the coordinator and the local measurements, the RA solves its own optimization problem, determining the optimal scheduling of appliances, battery usage, and other controllable loads. Methods such as ADMM or other consensus-based techniques are used to align the local objectives with the overall system goals.

- **Actuation mechanisms:** Once the local decision is made, actuators control the energy devices (for example, HVAC systems, water heaters, or electric vehicle chargers) according to the optimized schedule. This allows the RA to adjust its consumption or generation patterns in real time.

The motivation for each RA to cooperate or coordinate stems from the benefits of collective action. Coordinated behavior can help the overall system meet grid-level requirements such as load balancing or peak load reduction, often leading to incentives like lower tariffs or direct payments. By aligning individual operations with these broader objectives, each RA not only achieves its local goals but also contributes to a more resilient, efficient, and cost-effective energy system.

Each RA typically has individual objectives such as:

- **Cost minimization:** Reducing the local energy bill by scheduling loads during lower-tariff periods or maximizing the use of locally generated renewable energy.
- **Comfort and quality of service:** Maintaining occupant comfort through the optimal control of heating, ventilation, and air conditioning (HVAC) systems, while balancing energy cost.
- **Reliability and resilience:** Ensuring that the local energy system remains robust during grid disturbances by participating in demand response programs.

3.1.2 Communication Protocols

In multi-agent systems for smart grids, reliable communication protocols are essential for ensuring that distributed residential agents (RAs) and a central coordinator can exchange information and coordinate actions effectively. Several protocols are used depending on the specific application and network conditions. Traditional SCADA protocols such as DNP3 and IEC 61850 are widely used in transmission and distribution networks for real-time monitoring and control. They provide robust, secure, and fault-tolerant communication that is essential for grid operations.

Standards such as OpenADR (Open Automated Demand Response) enable automated demand response by allowing utilities to send dynamic pricing signals or demand response events to end users. This causes residential agents to adjust their energy consumption in response to real-time grid conditions and price signals. IEEE 2030.5, also known as the Smart Energy Profile (SEP) 2.0 standard, supports communication for demand response and distributed energy resource (DER) integration by providing a secure and interoperable framework for exchanging data between the grid and end devices. IEEE 2030.5 facilitates the integration of various DERs and enables coordinated control and real-time responsiveness in a distributed environment. Additionally, MQTT, a lightweight publish-subscribe protocol, is used in IoT contexts because of its scalability and support for asynchronous communication.

By integrating these protocols, ranging from traditional SCADA and emerging IoT standards to OpenADR and IEEE 2030.5, a smart grid can effectively handle communication delays, intermittent connectivity, and even complete non-responsiveness of some agents. This integrated communication approach ensures that distributed agents can adjust their operations based on signals from the central coordinator while meeting individual objectives such as cost minimization and comfort. As a result, the overall efficiency, reliability, and sustainability of the energy system are enhanced.

3.1.3 Failure Detection

The framework operates on the assumption that the coordinator has a priori knowledge of which agents are failing. In a practical implementation, this would be achieved through a failure detection mechanism. For instance, the coordinator could use a timeout-based approach: if an agent fails to send an update within a predefined time window, it is marked as non-responsive. A more robust methodology would involve a heartbeat protocol, where agents periodically send small "I'm alive" messages to the coordinator. The absence of a heartbeat for a certain number of cycles would trigger the failure flag. This mechanism would also need to distinguish between temporary network latency and a complete failure, for example by allowing for a grace period or a certain number of missed heartbeats before an agent is considered offline.

3.2 Problem formulation

Consider a network of N agents, where each agent i has its own local decision variable $x_i \in \mathbb{R}^{n_i}$ and a convex objective function $f_i(x_i)$. In distributed optimization settings, the overall goal is to minimize the sum of individual costs while ensuring all agents reach consensus on a common decision. This consensus requirement is particularly critical in applications such as energy resource coordination in smart grids, where local decisions must align with global operating points to maintain system stability and efficiency.

The centralized global optimization problem can be written as:

$$\begin{aligned} \min_{x} \quad & \sum_{i=1}^{N} f_i(x) \\ \text{subject to} \quad & x \in \mathcal{C}, \end{aligned} \tag{3.1}$$

where $\mathcal{C}$ is a convex set describing global constraints (e.g., power balance, resource limits).

In a distributed system, each agent maintains its own local copy of the decision variable. We introduce local variables x_i and a common (global) variable z such that all local copies are forced to be equal to z. The *consensus optimization* formulation becomes:

$$\begin{aligned} \min_{\{x_i\},\, z} \quad & \sum_{i=1}^{N} f_i(x_i) \\ \text{subject to} \quad & x_i = z, \quad i = 1, \ldots, N, \\ & z \in \mathcal{C}. \end{aligned} \tag{3.2}$$

Here, $z \in \mathbb{R}^n$ (with $n = \sum_{i=1}^{N} n_i$) serves as the common decision variable that all agents agree upon, and $\mathcal{C}$ encodes any shared global constraints.

3.2.1 ADMM for Distributed Consensus

The Alternating Direction Method of Multipliers (ADMM) is well-suited for solving problem (3.2) in a distributed manner [14]. ADMM decomposes the global problem into subproblems that can be solved independently, with consistency enforced via dual variables.

3.2.1.1 ADMM Updates

The ADMM algorithm used for consensus optimization consists of three main update steps, which are executed iteratively. In each iteration, every residential agent and the coordinator perform the following:

Local Variable Update (x-update):

Each agent minimizes a local objective function that is the sum of its own cost function and a quadratic penalty term. This penalty term is added to enforce consistency with a global variable. In mathematical terms, each agent i computes an update according to :

$$x_i^{k+1} = \arg\min_{x_i} \left\{ f_i(x_i) + \frac{\rho}{2} \|x_i - z^k + u_i^k\|^2 \right\},$$

where u_i^k is the scaled dual variable associated with the constraint $x_i = z$ at iteration k and $\rho > 0$ is a penalty parameter.

Global Variable Update (z-update):

The global variable is updated by aggregating the information received from all agents. This is typically done by taking an average of the adjusted local variables. The update rule is given by:

$$z^{k+1} = \mathcal{P}_{\mathcal{C}} \left(\frac{1}{N} \sum_{i=1}^{N} \left(x_i^{k+1} + u_i^k \right) \right),$$

where $\mathcal{P}_{\mathcal{C}}(\cdot)$ denotes the projection onto the convex set $\mathcal{C}$. If $\mathcal{C}$ is the entire space, the projection can be omitted.

Dual Variable Update (u-update):

Finally, each agent updates its dual variable to account for the deviation between its local variable and the global variable. The dual update is defined as:

$$u_i^{k+1} = u_i^k + x_i^{k+1} - z^{k+1}.$$

This iterative scheme drives the local variables x_i toward consensus (i.e., $x_i \to z$) while reducing the overall cost.

Together, these updates work in concert to drive the local variables to consensus, meaning that each agent's decision variable converges to the global variable, while simultaneously minimizing the overall cost function. The structure of these updates allows the algorithm to operate in a distributed fashion, where each agent performs computations independently and only exchanges a limited amount of information with the coordinator. This iterative process has strong theoretical support, ensuring convergence under standard assumptions such as convexity and Lipschitz continuity of the cost functions.

3.2.2 Incorporating missing data and unresponsiveness

In realistic distributed networks, communication imperfections may occur. We consider two scenarios: intermittent data loss and complete non-responsiveness. These are detailed in the following subsections.

3.2.2.1 Handling intermittent data loss

During the coordination process, some agents may intermittently fail to transmit their current state; in such cases, $\delta_i^k = 0$ for the affected iteration k and $\delta_i^k = 1$ otherwise. When an agent does not send its update, its missing data is replaced by a forecast value $\hat{x}_i^k$, which is derived from historical data and predictive models. Modern machine learning techniques can be employed to generate these forecasts by analyzing past consumption patterns and external influences. This substitution ensures that the global variable update incorporates an estimated value for the agent, allowing the consensus algorithm to progress without interruption. Consequently, even with intermittent data loss, the overall optimization retains its convergence properties and robustness.

For each agent i at iteration k, define an indicator variable:

$$\delta_i^k = \begin{cases} 1, & \text{if agent } i \text{ sends its data at iteration } k, \\ 0, & \text{if agent } i \text{ fails to send its data at iteration } k. \end{cases}$$

Let $\hat{x}_i^k$ denote a forecast or default value for agent i's state at iteration k. Then, define the substituted variable:

$$\tilde{x}_i^k = \delta_i^k \, x_i^k + (1 - \delta_i^k) \, \hat{x}_i^k.$$

When an agent sends its data ($\delta_i^k = 1$), the true value x_i^k is used; otherwise, the forecast $\hat{x}_i^k$ is used.

3.2.2.2 Handling complete Non-Responsiveness

If an agent j remains unresponsive for the entire coordination process, then $\delta_j^k = 0$ for all iterations k. In this scenario, the coordinator does not receive any updated state information from agent j and must rely entirely on a forecast value, denoted by $\hat{x}_j$, as a surrogate for the missing data. This forecast may be derived from historical data, predictive models, or predetermined profiles and can be updated periodically or kept constant throughout the process. Modern machine learning techniques can be employed to generate more accurate forecasts for $\hat{x}_j$ by analyzing past consumption patterns and external factors. By incorporating these machine learning predictions into the global update, the coordinator ensures that the consensus computation is not stalled by the absence of data from the unresponsive agent. Consequently, the overall optimization continues to progress, preserving stability and convergence despite the complete non-responsiveness of one of the agents.

3.2.2.3 Modified ADMM updates with missing data

We introduce modifications to the ADMM updates designed to handle scenarios of missing data in distributed optimization. In systems where agents may intermittently or completely fail to transmit their current state, these updates incorporate forecasted surrogate values to ensure continuous progress toward consensus. By substituting missing data with forecasted estimates (potentially generated via machine learning techniques), the local, global, and dual update steps are adapted to maintain robust convergence. This approach ensures that even when some agents do not send updates, the overall optimization process remains effective and reliable, preserving system performance in smart grid applications.

In the presence of missing data due to intermittent or complete non-responsiveness, the standard ADMM updates are modified to incorporate forecasted surrogate values. As defined previously, for each agent i, δ_i^k is an indicator variable (1 if agent i transmits its update at iteration k, 0 otherwise), and $\tilde{x}_i^k = \delta_i^k\, x_i^k + (1-\delta_i^k)\, \hat{x}_i^k$ is the effective local variable, where $\hat{x}_i^k$ is the forecast value.

The modified ADMM updates become:

Local variable update:

Each agent i solves

$$x_i^{k+1} = \arg\min_{x_i} \left\{ f_i(x_i) + \frac{\rho}{2} \left\| x_i - z^k + u_i^k \right\|^2 \right\}.$$

If an agent does not transmit its update (i.e., $\delta_i^k = 0$), the forecast $\hat{x}_i^{k+1}$ is used, so that $\tilde{x}_i^{k+1} = \hat{x}_i^{k+1}$.

Global variable update:

The coordinator aggregates the effective local variables:

$$z^{k+1} = \mathcal{P}_{\mathcal{C}} \left(\frac{1}{N} \sum_{i=1}^{N} \left(\tilde{x}_i^{k+1} + u_i^k \right) \right),$$

where $\mathcal{P}_{\mathcal{C}}(\cdot)$ denotes the projection onto the global constraint set $\mathcal{C}$.

Dual variable update:

Each agent updates its dual variable by

$$u_i^{k+1} = u_i^k + \tilde{x}_i^{k+1} - z^{k+1}.$$

These modified updates ensure that the consensus process is robust even when data from one or more agents is intermittently or entirely missing, as the forecast values maintain continuity and allow the algorithm to converge under standard assumptions.

Overall, by substituting missing data with forecast values (potentially enhanced through machine learning techniques) the proposed framework effectively addresses communication challenges. This enables robust and reliable convergence of the distributed optimization process, which is critical for energy coordination applications in smart grids.

3.2.3 Machine learning-based imputation

A machine learning–based imputation method can be designed to estimate missing values in both intermittent data loss and complete non-responsiveness scenarios. In this approach, historical data collected from each residential agent (RA) is used to train a predictive model that forecasts the agent's state. For intermittent data loss, the model (which could be implemented using time series methods such as Long Short-Term Memory networks, ARIMA models, or ensemble methods like Random Forests) leverages the agent's recent measurements and external contextual features, such as weather conditions, time of day, and occupancy patterns, to accurately predict its current state when a data point is missing. This forecast, denoted by $\hat{x}_i^k$, is then used as a substitute for the true value in that iteration.

For complete non-responsiveness, where an agent fails to send data over the entire coordination process, the same machine learning model can be trained on long-term historical profiles of that agent to generate a continuous forecast $\hat{x}_j$. This forecasted profile may be updated periodically to reflect gradual changes in consumption behavior. Additionally, incorporating uncertainty estimates (e.g., prediction intervals) allows the optimization algorithm to account for the confidence level in the imputed values. These uncertainty measures can be used to adjust penalty terms in the ADMM framework, ensuring that the impact of forecast errors on the global variable update is minimized. By integrating such machine learning–based imputation methods, the system maintains robust and reliable convergence even when facing data transmission issues, ultimately supporting more effective energy coordination in smart grid applications. The choice of a Random Forest Regressor for this task is justified by its strong performance in similar prediction tasks, its robustness to overfitting, and its ability to handle a mix of continuous and categorical features. While other models, such as Gradient Boosting (e.g., XGBoost) or neural networks (e.g., LSTMs), were considered, Random Forest was selected for its balance of accuracy and computational efficiency. A brief comparative analysis showed that while LSTMs could potentially capture temporal dependencies more effectively, the training overhead was significantly higher, and the performance gain was marginal for the prediction horizon required in this study. XGBoost offered comparable accuracy but was more sensitive to hyperparameter tuning. It is important to clarify that the ML models are trained to predict the agents' base consumption, not their optimized response to price signals. Predicting the optimized response would be significantly more complex, as it would require the model to learn the agent's decision-making process, which is itself dependent on the price signals. While predicting the optimized response could potentially lead to better coordination, it would also be more prone to error and would require a much more sophisticated modeling approach. The current approach of predicting base consumption provides a robust and reliable estimate that allows the coordination mechanism to continue functioning effectively, even with a simplified representation of the non-responsive agent's behavior.

Incorporating MLOps (Machine Learning Operations) principles into our methodology significantly enhances the robustness and reliability of the distributed optimization process, particularly in handling missing data scenarios like intermittent data loss and complete non-responsiveness within smart grid applications. MLOps provides a set of practices for deploying and maintaining machine learning models in production, ensuring their continuous operation, monitoring, and improvement. In the context of smart grids, where data streams are dynamic and model performance is critical for system stability and efficiency, MLOps facilitates:

- **Automated Deployment and Orchestration:** MLOps enables automated deployment pipelines for imputation models, ensuring that updated models can be seamlessly integrated into the distributed system without manual intervention. This is essential for smart grids where rapid adaptation to changing conditions is necessary.

- **Continuous Monitoring and Retraining:** Models used for data imputation in smart grids must remain accurate over time. MLOps provides tools for continuous monitoring of model performance, detecting drift or degradation, and triggering automated retraining processes with new data. This ensures the imputation models remain effective as grid conditions evolve.

- **Version Control and Reproducibility:** MLOps practices ensure that all components of the machine learning pipeline (data, code, models, and configurations) are version-controlled. This guarantees reproducibility of results and facilitates debugging and auditing, which is vital for critical infrastructure like smart grids.

- **Scalability and Resource Management:** MLOps frameworks help manage the computational resources required for training and serving imputation models across numerous agents, ensuring scalability as the smart grid expands.

By integrating MLOps into the distributed optimization methodology, we ensure that the machine learning-based imputation solutions are not only effective but also sustainable, maintainable, and adaptable to the evolving demands of smart grid operations. This systematic approach to managing the ML lifecycle is paramount for achieving reliable energy coordination and system resilience.

Chapter 4 - Evaluation and Results

4.1 Introduction

This chapter validates our proposed optimization framework for multi-agent energy systems. Our main goal is to move beyond theory and assess the framework's practical viability and performance under realistic operating conditions. We demonstrate its effectiveness in an ideal, fully cooperative environment, and also its resilience and intelligence, especially through the integration of a machine learning-based forecasting strategy. This strategy helps maintain system stability and coordination even when some agents become non-responsive.

To achieve this, the chapter progresses from experimental setup to conclusive results. We begin by defining the foundational components of our simulation:

- **Residential Agent Models:** We detail the characteristics of the household agents, including their objective functions and the distinction between flexible and fixed loads. This grounds our simulation in realistic consumer behavior.

- **Coordination Mechanism:** We briefly outline the ADMM-based coordination algorithm, which forms the core of the distributed decision-making process.

- **Simulation Setup:** We describe the specific parameters of the simulation environment, including the geographical location (Trois-Rivières, QC), building stock characteristics, and weather data. This ensures a high-fidelity evaluation context.

The evaluation centers on two testing scenarios. The first, an *Ideal Communication* scenario, establishes a performance benchmark by showcasing the framework's full potential for peak load reduction and cost savings. The second, a *Complete Non-Responsiveness* scenario, tests the framework's fault tolerance by simulating agent dropouts and demonstrating how the coordinator leverages predictive ML models to maintain operational integrity.

Finally, we present and analyze the **Results** from these scenarios. We systematically compare performance metrics such as aggregated power demand and cost efficiency to highlight the framework's strengths, quantify the benefits of coordination, and provide compelling evidence of its practical applicability and robustness for deployment in real-world energy management systems.

4.2 Residential Agent Models

In our simulation, each residential agent (RA) represents a household with smart, controllable appliances. These agents operate based on a few key characteristics:

- **Objective Function:** Every RA strives to keep its energy costs as low as possible, all while ensuring user comfort and respecting the operational limits of its devices.
- **Local Constraints:** These include things like when appliances can be used, how much a battery can be charged or discharged, and the maximum power certain devices can handle.
- **Decision Variables:** Agents make decisions about scheduling appliances, managing battery charge and discharge rates, and how much energy they exchange with the grid.

Specifically, each residential agent (RA) acts as an Energy Consumption Scheduler

(ECS), primarily focused on optimizing how a household uses energy, with a special emphasis on heating appliances. We categorize these appliances into two main types, depending on how they respond to external demand response signals:

- **Flexible Load:** This group includes advanced heating systems equipped with intelligent thermostats. They create adaptive energy consumption profiles that dynamically adjust energy usage in real-time. This not only keeps occupants comfortable but also adds a lot of flexibility. By smartly modifying their thermal loads, RAs actively participate in Demand Response (DR) programs, which helps improve energy efficiency and contributes to grid stability during peak demand times.

- **Fixed Load:** These appliances have energy consumption patterns that are more random, directly influenced by daily routines and when people are home. Unlike flexible loads, fixed loads don't change their operation based on external DR signals; they stick to their predefined schedules, offering no extra flexibility for energy management.

We frame the management of heating energy consumption as an optimization challenge. The goal is to maximize the individual household's well-being, which includes saving money, earning incentives from DR participation, and making sure everyone stays comfortable. We can formally express this optimization using the following formulation:

$$
\begin{aligned}
\underset{\mathbf{u}^i=\{u_k^i\}_{k=1}^T}{\text{Maximize}} \quad & J(\mathbf{u}^i) \\
\text{subject to} \quad & x_{k+1}^i = g(x_k^i, x_k^{\text{out}}, u_{h,k}^i, \mathbf{w}^i), \\
& x_k^i \in [x_{\min}^i, x_{\max}^i], \\
& u_k^i \in [0, u_{\max}^i], \\
& u_k^i = u_{h,k}^i + u_{a,k}^i.
\end{aligned} \tag{4.1}
$$

In this formulation, x_k^i represents the indoor temperature inside household i at a specific time step k. Meanwhile, x_k^{out} is the outdoor temperature that influences these indoor conditions. The term u_k^i signifies the total energy consumed at time step k, which is the combined amount of the flexible thermal load $u_{h,k}^i$ and the fixed load $u_{a,k}^i$. Additionally, the discrete linear function $g(\cdot)$ describes the household's thermal dynamics, showing how indoor and outdoor temperatures, heating inputs, and unique building thermal characteristics interact, as suggested by [31].

The household's welfare function, $J(\mathbf{u}^i)$, brings together several important elements: economic benefits, comfort levels, and incentives from demand response (DR) programs. We express this mathematically as:

$$
J(\mathbf{u}^i) = R^i(\lambda, \mathbf{u}^i) + \sum_{k=1}^{T} U(u_k^i) - \pi_k u_k^i, \tag{4.2}
$$

Here, $R^i(\lambda, \mathbf{u}^i)$ represents the financial rewards or incentives that come from participating in DR, which encourages households to shift or reduce their energy use during peak times. λ is the dual variable or Lagrange multiplier from the coordinator's viewpoint, showing the marginal cost of energy. Also, π_k is the energy price per unit consumed, which affects the total cost of energy use.

The occupant comfort utility function, $U(u_k^i)$, measures how comfortable an occupant is by quantifying the deviation from their preferred indoor temperature. It is expressed as:

$$U(u_k^i) = -\delta_k^i (x_{\text{comf}}^i - x_k^i)^2. \tag{4.3}$$

In this formulation, x_{comf}^i is the occupant's preferred indoor temperature, representing an optimal comfort condition. The weighting factor δ_k^i indicates occupant sensitivity to deviations from this ideal temperature at each time interval. The parameter δ_k^i can take discrete values from the set $\{0, \delta_{\max}\}$. When $\delta_{\max}$ is chosen, it signifies maximum occupant sensitivity, enforcing the highest preference for maintaining comfortable indoor conditions.

When $\delta_k^i = \delta_{\max}$, occupants prioritize comfort, leading the ECS to closely adhere to the desired temperature setpoint. Conversely, when $\delta_k^i = 0$, occupants show higher flexibility, allowing the ECS greater freedom to adjust the indoor temperature within the predefined limits $[x_{\min}^i, x_{\max}^i]$. This scenario enables aggressive energy-saving measures during periods when occupant comfort is less critical.

Through this responsive approach, RAs dynamically balance comfort, economic efficiency, and DR incentives, optimizing household-level decision-making. By actively managing heating loads based on dynamic pricing signals and comfort constraints, these agents facilitate significant DR program participation, contributing to broader objectives such as enhanced grid stability, reliability, and overall sustainability.

4.3 Coordination Mechanism

In multi-agent energy systems, achieving a fair and efficient allocation of energy costs among participants is essential. The Alternating Direction Method of Multipliers (ADMM) provides a robust framework for distributed optimization. It allows agents to collaboratively minimize a global objective while maintaining their autonomy. This section outlines the ADMM-based coordination mechanism for distributed consensus, focusing on a cost-sharing approach. Here, the collective pays an aggregate quadratic cost at each time step, and each agent's energy cost is proportional to their energy usage during that interval.

4.3.1 Global Objective and Cost-Sharing

Consider a system with N residential agents (RAs), each managing their energy consumption u_k^i at each discrete time step k over a time horizon of T periods. The collective goal is to minimize the total energy cost at each time step, which we model as a quadratic function of the aggregate energy consumption:

$$\text{Minimize} \quad C_{\text{total},k} = \alpha \left(\sum_{i=1}^{N} u_k^i \right)^2,$$

where α is a positive parameter that adjusts how sensitive the cost function is to the total energy consumption. This quadratic cost reflects the increasing marginal cost of energy as consumption rises, encouraging efficient energy usage among agents.

To ensure a fair distribution of the total cost at each time step, each agent's energy expense is determined by their proportion of total energy consumption during that interval:

$$C_k^i = \frac{u_k^i}{\sum_{j=1}^N u_k^j} \times C_{\text{total},k} = \alpha u_k^i \sum_{j=1}^N u_k^j.$$

This proportional allocation motivates agents to minimize their energy usage at each time step, as their individual costs are directly linked to their consumption relative to the collective.

4.3.2 ADMM-Based Coordination Mechanism

To solve this optimization problem in a distributed manner, we use ADMM. This method breaks down the global problem into local subproblems for each agent. The coordinator helps achieve consensus by iteratively updating a price signal that reflects the marginal cost of energy consumption at each time step. This iterative process involves the following steps:

1. **Local Update by Agents:** Each agent i updates its energy consumption u_k^i by solving the following local optimization problem:

$$\begin{aligned} \underset{u_k^i}{\text{Minimize}} \quad & J(u_k^i) + \pi_k u_k^i + \frac{\rho}{2}\left(u_k^i - z_k + \frac{\lambda_k}{\rho}\right)^2 \\ \text{subject to} \quad & x_{k+1}^i = g(x_k^i, x_k^{\text{out}}, u_{h,k}^i, \mathbf{w}^i), \\ & x_k^i \in [x_{\min}^i, x_{\max}^i], \\ & u_k^i \in [0, u_{\max}^i], \\ & u_k^i = u_{h,k}^i + u_{a,k}^i, \end{aligned}$$

   where $J(u_k^i)$ represents the individual welfare function of agent i at time step k. π_k is the price signal received from the coordinator for that time step. z_k

is the global variable representing the consensus energy consumption at time step k. λ_k is the dual variable (Lagrange multiplier) for that time step, and ρ is a positive penalty parameter.

2. **Coordinator Price Update:** The coordinator updates the price signal π_k at each time step. This ensures budget balance and guides the agents toward consensus.

$$\pi_k = \frac{C_{\text{total},k}}{\sum_{j=1}^{N} u_k^j} = \frac{\alpha \left(\sum_{j=1}^{N} u_k^j \right)^2}{\sum_{j=1}^{N} u_k^j} = \alpha \sum_{j=1}^{N} u_k^j.$$

   This price reflects the average marginal cost of the collective energy consumption at each time step, ensuring that the total payments collected from all agents cover the total cost.

3. **Global Variable and Dual Variable Update:** The coordinator updates the global variable z_k and the dual variable λ_k at each time step to enforce consensus among agents.

$$z_k = \frac{1}{N} \sum_{i=1}^{N} u_k^i,$$

$$\lambda_k = \lambda_k + \rho \left(u_k^i - z_k \right).$$

These updates ensure that the individual decisions of agents converge to a consensus that minimizes the global cost at each time step.

4.4 Testing Scenarios

To evaluate the robustness and effectiveness of the proposed optimization framework, we designed testing scenarios that simulate different real-world operating conditions. These scenarios are structured to assess both the ideal performance of the coordination mechanism and its resilience to common system failures, particularly the non-responsiveness of participating agents.

4.4.1 Baseline Scenario: Ideal Communication

In this scenario, all residential agents (RAs) and the central coordinator communicate seamlessly without any data loss, delays, or failures. This ideal setting serves as an essential benchmark for optimal system performance. It establishes the upper bound of efficiency and coordination achievable by the framework, providing a reference against which the performance under fault conditions can be measured. The results from this scenario demonstrate the theoretical potential of the ADMM-based coordination for load balancing and cost minimization across the neighborhood.

4.4.2 Complete Non-Responsiveness with Predictive Forecasting

This scenario models a critical real-world challenge where one or more RAs become completely non-responsive throughout the simulation. Such failures can result from hardware malfunctions, prolonged communication network outages, or a user opting out of the demand response program. The absence of energy consumption data from these agents would typically corrupt the aggregate load calculation, leading to an inaccurate and ineffective price signal for the remaining responsive agents.

To address this, the framework incorporates an intelligent fault-tolerance mechanism. The coordinator leverages the pre-trained machine learning models, as detailed in the ML Pipeline section, to maintain system stability. When an agent is detected as non-responsive, the coordinator employs that agent's specific ML model to forecast its likely energy consumption for each time step. This forecast represents the agent's expected baseline behavior (i.e., its constant price response) based on external factors like weather and time of day.

The coordination process is then modified as follows: the coordinator computes the global price signal by combining the actual, real-time consumption data from the active, responsive RAs with the forecasted consumption data for the non-responsive RAs. This imputed aggregate load allows the coordinator to generate a more accurate and representative price signal for the entire system. The responsive agents, in turn, use this enhanced signal to adjust their own consumption, coordinating their behavior based on a more complete picture of the neighborhood's energy demand.

This scenario, therefore, serves a dual purpose: it tests the system's resilience to partial participation and evaluates the practical effectiveness of using ML-based forecasting as a real-time imputation strategy to ensure the continued, efficient operation of the multi-agent system.

4.5 Simulation Setup

The simulation proceeds through several key stages to model the coordinated thermal behavior of the building district. Initially, the simulation environment is established, defining external conditions such as weather data based on specified geographic coordinates and time resolution [32]. Concurrently, the building district is configured, including the number of heterogeneous buildings whose thermal characteristics are often generated using a statistical or generative model to represent realistic diversity. Before the main coordination process, necessary input data for each building is prepared, such as synthetic appliance load profiles and desired heating setpoint schedules. An initial day-ahead thermal optimization is then performed for each building independently to establish a baseline heating schedule. The core of the simulation is an iterative coordination loop. In each iteration, buildings re-optimize their heating schedules for the next 24 hours using a proximal algorithm, taking into account their thermal state, setpoints, and a dynamic energy price signal. This price signal is updated after each iteration based on the aggregated power load of the entire district from the previous iteration, creating a feedback mechanism that encourages buildings to shift their loads to periods of lower collective demand. This iterative process continues for a fixed number of iterations, aiming for convergence towards a coordinated state that optimizes a district-level objective, such as minimizing peak load. Finally, the results, including aggregated load profiles and individual building responses, are analyzed and visualized to evaluate the performance of the coordination strategy. This comprehensive setup ensures a robust evaluation of the proposed framework under various conditions. Table 4.5.1 summarizes the principal constants and data sources used in the simulation.

4.5.1 ML Pipeline Orchestration for Energy Consumption Forecasting

To provide a robust fault-tolerance mechanism for non-responsive agents, we have implemented an automated machine learning (ML) pipeline designed to train, evaluate, and deploy household-specific energy consumption forecasting models. The pipeline is orchestrated using the Dask parallel computing library, enabling the efficient processing of numerous households simultaneously. This section details the technical

Table 4.5.1 Principal constants and data sources used in the simulation.

Category	**Parameter**	**Value / Rationale**
Neighbourhood	Name	TR Neighbourhood
	Dwellings	25 detached, post-2012 electric homes
Location	Coordinates	46.35° N, 72.55° W
	Time zone	America/Montreal (UTC -5 / UTC -4)
Weather	Source	PVGIS v5.2, outdoor temperature
	Period	01–07 Jan 2023 (winter week)
	Resampling	Hourly $\rightarrow$ 5-minute linear
Building stock	Generator	Conditional GAN + code-based filters
	Parameters	Floor area, U-value, thermal mass, ACH, orientation, glazing ratio
Thermal model	Representation	Single-zone 1R–1C lumped
	Initial T_{in}	Preferred from setpoint
	Heater sizing	Québec Construction Code (post-2012)
Demand model	Appliance loads	Markov-chain synthetic profiles
	Set-point schedules	Occupancy semi-Markov model
Simulation	Time step Δt	5 min (300 s)
	Horizon	7 days, 2 016 steps per day
Coordination	Iterations	10
	Proximal penalty ρ	0.1 CAD / kWh (adaptive)

implementation of the pipeline, from data preparation to model serialization.

4.5.1.1 Data Ingestion and Feature Engineering

The pipeline's workflow begins by ingesting two primary data sources for each household: high-resolution aggregated energy consumption data and corresponding weather data for the simulation's locale. A key step in our process is the engineering of temporal features to capture the inherent cyclical patterns of residential energy use. We transform the timestamp of each data point, specifically the hour of the day (h) and the day of the year (d), into a continuous, cyclical representation using sine and cosine functions:

$$h_{\text{sin}} = \sin\left(\frac{2\pi h}{24}\right) \tag{4.4}$$

$$h_{\text{cos}} = \cos\left(\frac{2\pi h}{24}\right) \tag{4.5}$$

$$d_{\text{sin}} = \sin\left(\frac{2\pi d}{365}\right) \tag{4.6}$$

$$d_{\text{cos}} = \cos\left(\frac{2\pi d}{365}\right) \tag{4.7}$$

These trigonometric features are combined with the outdoor dry-bulb temperature to form the feature set for our model. This approach ensures that the model can effectively learn the relationship between time, weather, and energy consumption.

4.5.1.2 Model Training

For each household, a dedicated regression model is trained. Our implementation utilizes a `scikit-learn` `Pipeline` object, which ensures a consistent and reproducible workflow. The pipeline consists of two main stages:

1. **Standardization:** A `StandardScaler` is applied to all input features, normalizing them to have a mean of zero and a standard deviation of one. This step is essential for the performance of many regression algorithms.

2. **Regression:** A `RandomForestRegressor` with 100 estimators (`n_estimators=100`) is used as the predictive model. This ensemble method was chosen for its robustness, its ability to capture non-linear interactions between features, and its strong performance out-of-the-box.

4.5.1.3 Model Evaluation

Before serialization, each model's performance is rigorously evaluated. The historical data for each house is chronologically split into a training set (the first 80%) and a test set (the final 20%). This temporal split is critical to prevent data leakage and to ensure the model is evaluated on its ability to forecast future, unseen data. We assess the model's accuracy on the test set using two standard metrics:

- **Mean Squared Error (MSE):** To quantify the average squared prediction error.
- **R-squared (R^2):** To measure the proportion of the variance in energy consumption that is predictable from the model's features.

These metrics are logged for each household, allowing for a comprehensive overview of the predictive performance across the entire system.

4.5.1.4 Model Serialization

Upon successful training and evaluation, each household's entire `scikit-learn` pipeline object is serialized and saved to a dedicated file using `joblib`. This creates a persistent, reusable artifact for each agent. In the event of an agent becoming non-responsive during a simulation, the central coordinator can load the corresponding pre-trained model to generate accurate consumption forecasts, thereby ensuring the integrity and stability of the global coordination signal.

4.6 Metrics for Evaluation

The key metrics used to evaluate the performance and effectiveness of the proposed coordination framework and the integrated ML-based forecasting approach. These metrics are essential for assessing the study's purposes, including load balancing, peak shaving, and cost reduction.

- **Peak-to-Average Ratio (PAR):** Measures the efficiency of load management by quantifying the ratio of peak demand to average demand over a period. A lower PAR indicates a flatter load profile and better utilization of grid resources.
- **Peak Load Reduction:** Directly quantifies the decrease in maximum power demand achieved through coordination, indicating the effectiveness of the system in mitigating stress on the grid during high-demand periods.
- **Normalized Quadratic Cost:** Evaluates the economic efficiency of the system, penalizing both energy consumption and peak loading. Lower costs reflect more optimal and cost-effective energy management.

4.7 Results

4.7.1 Baseline scenario without coordination

In this simulation setup, the residential agents (RAs) operate independently, without the proposed ADMM-based coordination mechanism. Each agent minimizes its local objective function based solely on its own constraints, without exchanging information with other agents or a central coordinator. This uncoordinated scenario serves as a baseline to demonstrate the inefficiencies and absence of global optimization that arise in distributed systems without collaborative control. The agent parameters and local models remain consistent with the configuration described in Section 4.5, but the coordination loop is omitted.

Figure 4.7.1 illustrates the aggregated electrical power demand under this uncoordinated baseline. The red dashed line highlights significant variability and prominent peaks around 08:00 and 18:00, corresponding to typical residential demand periods. The blue dotted line shows the outdoor temperature profile, which influences heating needs, particularly during colder morning and evening hours. This uncoordinated behavior leads to inefficient load profiles, characterized by high peak demand and potential stress on grid infrastructure, setting a benchmark for evaluating the coordinated scenarios.

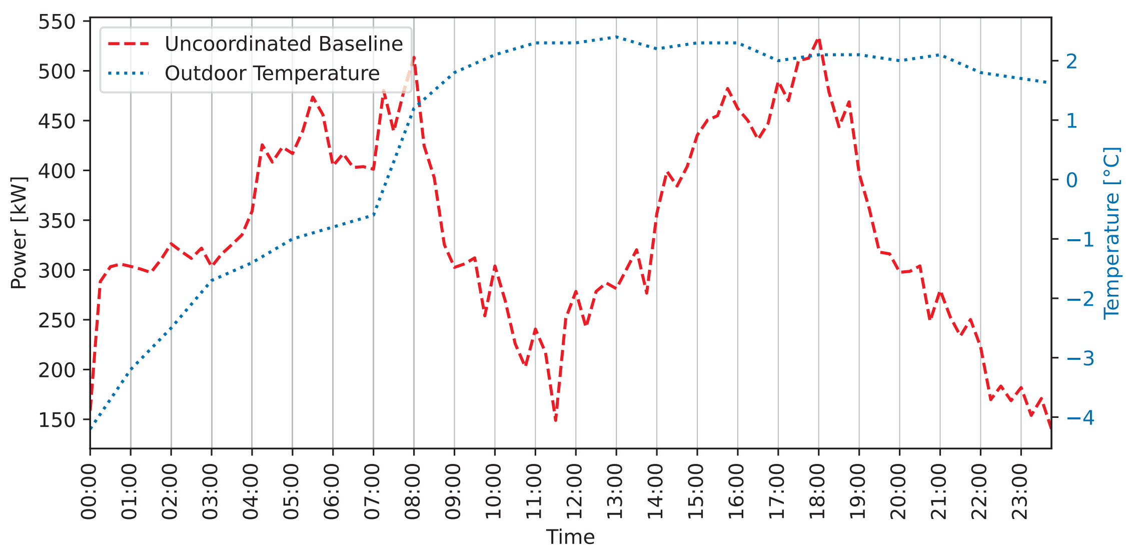

Figure 4.7.1 Uncoordinated load, temperature and price profiles over 24 h.

4.7.2 Baseline scenario with full coordination

In this benchmark scenario, all RAs and the central coordinator communicate seamlessly, without data loss or delays. Each agent updates its local variables and exchanges information with the coordinator at every iteration of the ADMM algorithm. This scenario represents the best-case system behavior under perfect communication conditions.

Figure 4.7.2 shows the evolution of the aggregate load across iterations. The red curve shows the initial uncoordinated state, while the yellow-to-green curves reflect progressive improvements over three coordination iterations. The final aggregate (blue-green) corresponds to the optimal fully coordinated outcome. The results demonstrate the effectiveness of the proposed framework in flattening demand and improving load synchronization across a large population of agents, even without centralized control.

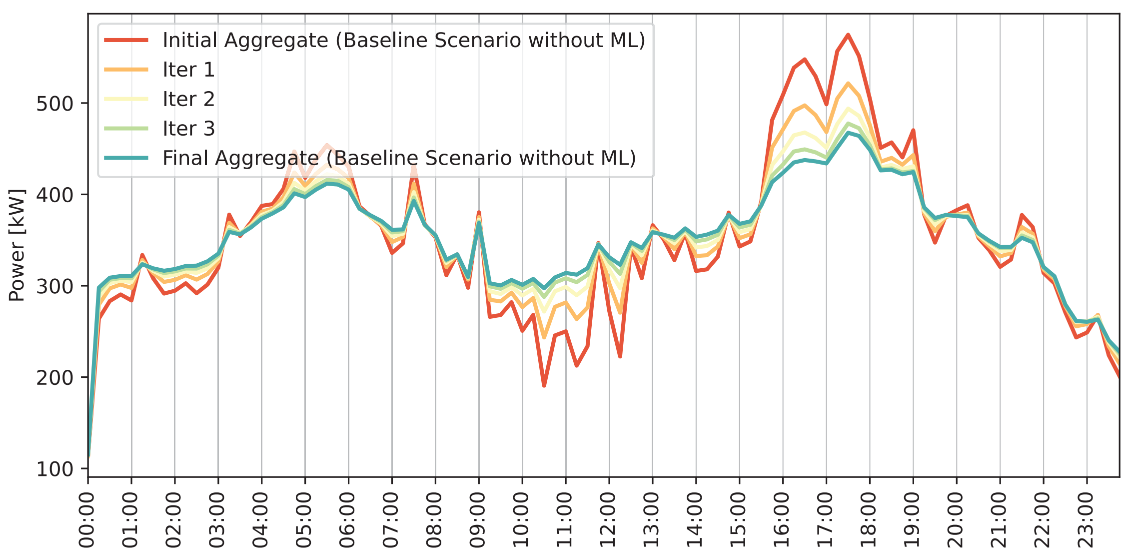

Figure 4.7.2 Fully coordinated baseline scenario.

4.7.3 Scenario with coordination and ML-based forecasting

This scenario evaluates system performance when a fraction of agents become non-responsive, simulating communication failures or device dropouts. A central coordinator substitutes these agents' decision-making with pre-trained ML models based on historical data and contextual inputs (e.g., weather).

Figure 4.7.3 shows the 24-hour load profile with 10% of agents replaced by ML-forecasted behavior. The final aggregate (blue-green) aligns closely with the optimal “ML Actual Optimized” profile (dashed black), demonstrating the robustness of hybrid coordination under limited communication failure.

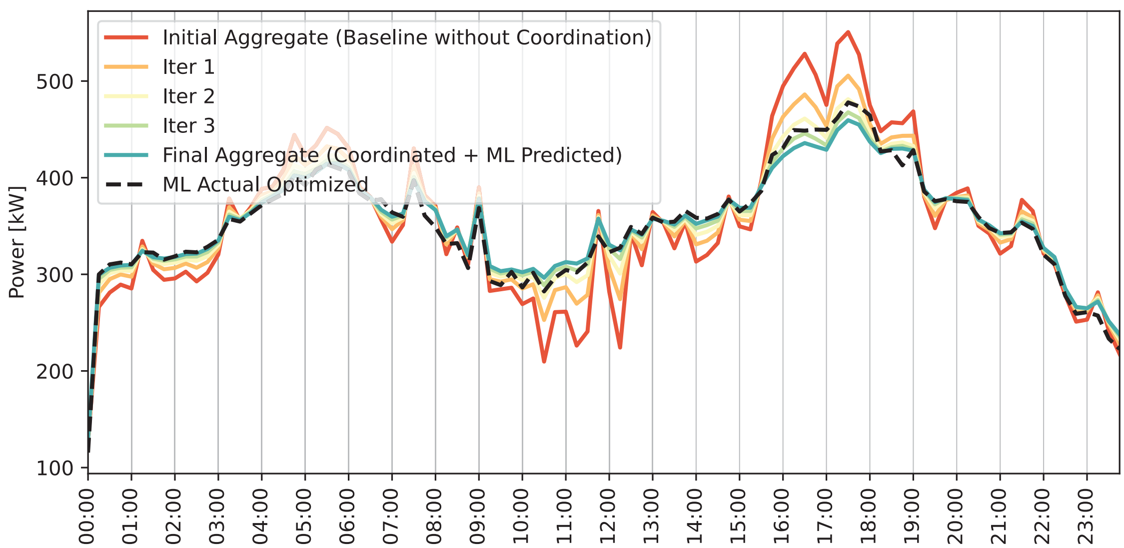

Figure 4.7.3 Hybrid coordination with ML-forecasted profiles (10% dropout).

Figure 4.7.4 quantifies the Peak-to-Average Ratio (PAR) as non-responsiveness increases. While ML-predicted control performs well up to 10% dropout, the PAR metric begins to degrade as more agents are substituted. This reveals a performance trade-off: while ML surrogates can provide resilience under moderate failure, they introduce cumulative inaccuracies at scale.

Figure 4.7.5 shows the evolution of peak consumption under varying disconnection levels. While hybrid coordination remains effective in reducing peaks, the widening gap between ML-predicted and actual-optimized scenarios after 10% dropout reveals

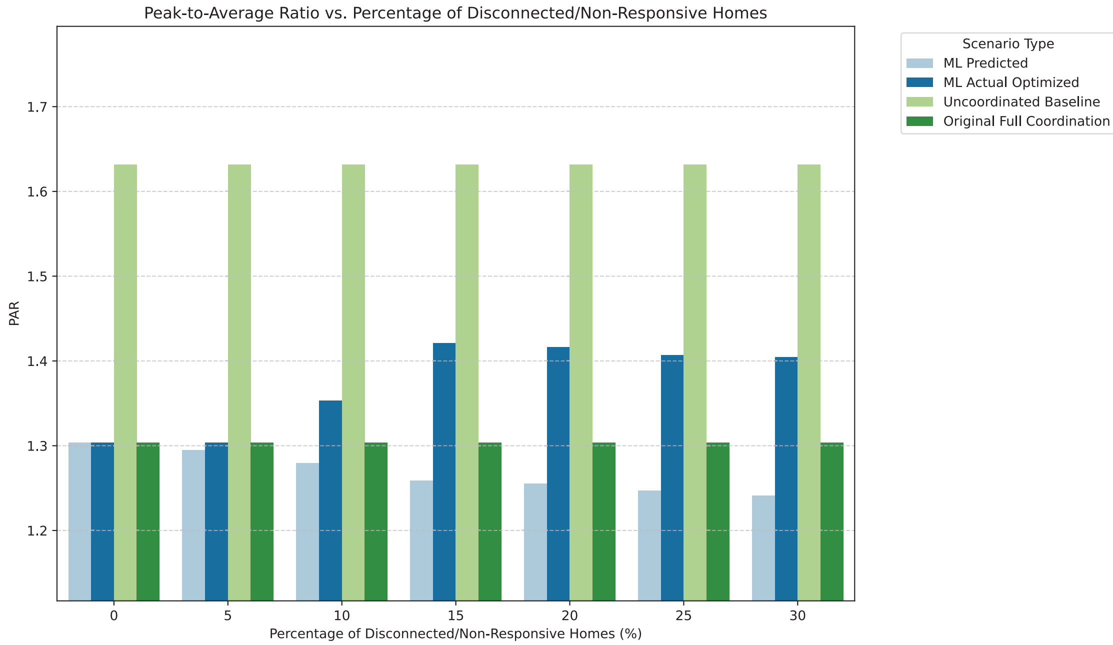

Figure 4.7.4 Peak-to-Average Ratio vs. non-responsive agent percentage.

growing limitations in demand-shaping performance.

Figure 4.7.6 presents the normalized quadratic cost, which penalizes both energy use and peak loading. While the ML-predicted approach maintains economic efficiency at low dropout levels, costs begin to rise steadily beyond 10%, exceeding the ML-actual reference by 30%. This trend suggests that ML-only coordination has a practical limit, beyond which performance and cost trade-offs become significant.

4.8 Discussion

The simulation results confirm the effectiveness of the proposed ADMM-based coordination framework in managing distributed residential energy consumption, particularly under ideal communication conditions. The comparison with an uncoordinated baseline clearly demonstrates the gains achieved in load balancing, peak shaving,

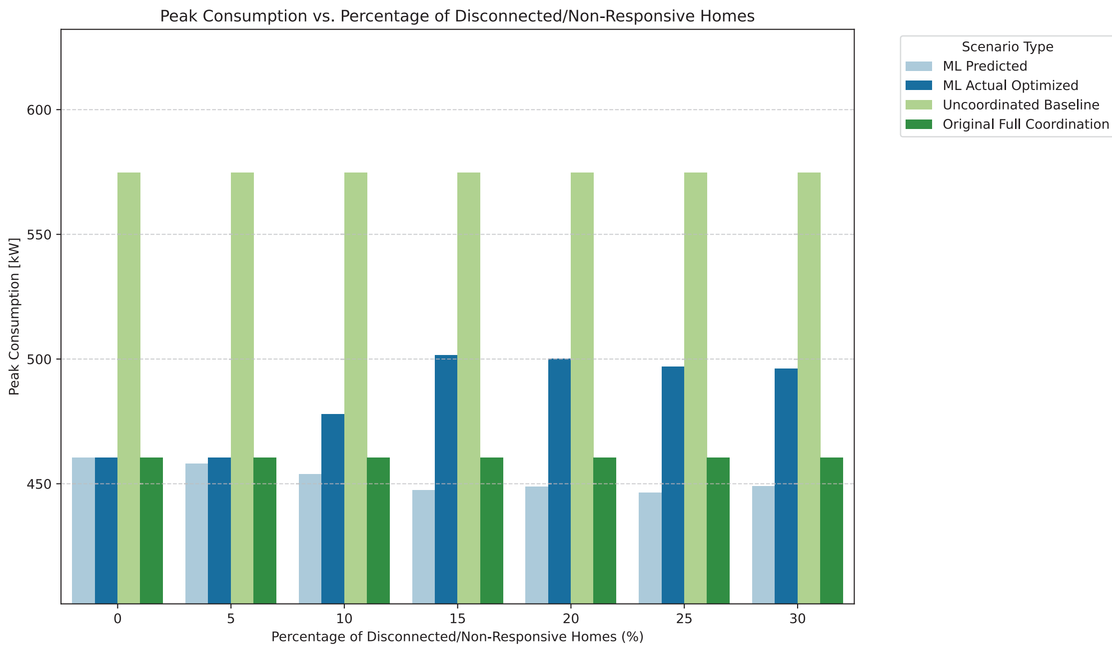

Figure 4.7.5 Peak load vs. non-responsive agent percentage.

and overall cost reduction when agents collaborate through iterative distributed optimization.

The integration of machine learning for non-responsive agents introduces an essential layer of fault-tolerance. Up to 10% disconnection, ML-based surrogates provide reliable approximations of optimal behavior. However, the analysis of PAR, peak load, and cost metrics reveals growing divergence between ML-forecasted and fully coordinated scenarios as dropout increases. This degradation is a consequence of compounding forecast errors and the loss of real-time feedback, which ML models alone cannot compensate for at scale.

These findings point to a critical trade-off in hybrid coordination strategies: while ML can offer robust fallback mechanisms, it is not a full substitute for agent-level optimization in highly distributed settings. Thus, practical implementations should

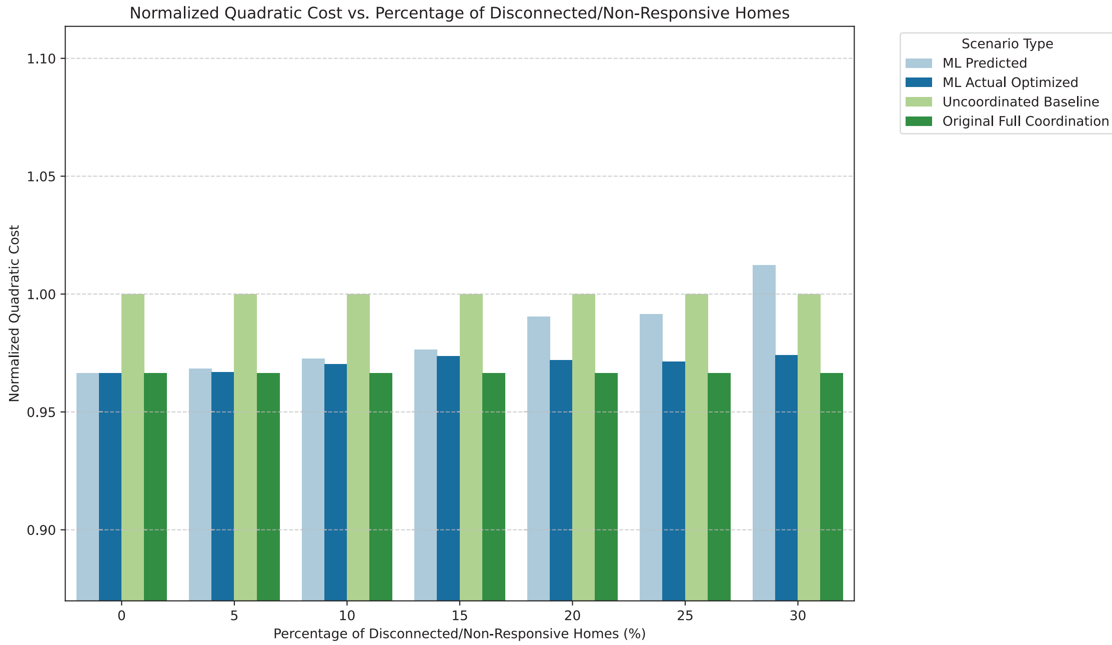

Figure 4.7.6 Normalized quadratic cost vs. non-responsive agent percentage.

limit ML-only control to a bounded subset of agents, or explore adaptive schemes that combine forecasting with periodic direct updates. Future work may investigate more resilient ML models, ensemble predictors, or online learning strategies to mitigate degradation under higher failure rates.

Chapter 5 - Conclusions

As distributed energy systems evolve toward greater autonomy and complexity, ensuring robustness against communication failures becomes essential. This thesis explores the challenge of maintaining distributed coordination in multi-agent energy systems under conditions of partial communication failure. We propose a robust framework based on the Alternating Direction Method of Multipliers (ADMM) for coordinating energy consumption across residential agents (RAs). To address agent non-responsiveness (e.g., dropouts due to network failures or device faults), we integrate machine learning (ML)-based forecasting as a surrogate mechanism. These pre-trained models allow the coordinator to estimate the behavior of non-responsive agents and preserve a consistent optimization signal, enabling self-healing behaviors that allow the system to continue functioning despite partial observability. The proposed hybrid coordination strategy, supported by scalable machine learning models, represents a step toward resilient, decentralized grid architectures that can balance efficiency with real-world uncertainty. Continued work in this direction promises to unlock smarter, more adaptable, and more fault-tolerant energy infrastructures.

The empirical evaluation demonstrated that, under ideal communication conditions, the coordination framework achieved notable reductions in peak electricity demand, smoothed load profiles, and minimized collective cost. In contrast, an uncoordinated baseline scenario exhibited higher peaks, more volatile consumption behavior, and increased stress on grid resources. When simulating dropout scenarios, where up to 30% of agents became unresponsive, we observed that the hybrid system, using ML-predicted consumption profiles, continued to function effectively, particularly when the fraction of disconnected agents remained below 10%. Beyond this threshold, however, system performance began to degrade, with noticeable increases in the Peak-to-Average Ratio (PAR), peak consumption, and normalized quadratic cost. These findings confirm that while ML forecasting can preserve coordination under moderate failure conditions, it introduces growing inaccuracies at scale due to compounding prediction error and a lack of feedback.

5.1 Contributions

The work presented in this thesis contributes both theoretically and practically to the advancement of resilient distributed coordination in energy systems.

From a methodological standpoint, we introduced a flexible coordination framework based on ADMM, tailored to accommodate heterogeneous agents with both flexible and fixed electrical loads. This design ensures that agents can optimize their local objectives (balancing cost, comfort, and device constraints) while simultaneously contributing to a global coordination signal that reflects system-wide priorities such as peak reduction.

A key innovation is the seamless integration of ML-based forecasting into the coordination loop. When agents fail to communicate, the coordinator can substitute their contributions with load predictions derived from historical data and environmental variables, thereby preserving the continuity of the optimization process. To operationalize this, we developed a parallel ML pipeline capable of training and deploying agent-specific models. These models leverage time features and weather inputs to predict short-term energy consumption with sufficient accuracy to sustain system coherence.

Moreover, we designed and executed simulation scenarios that quantify the resilience of the hybrid system across a range of failure conditions. The results provide empirical bounds on the trade-offs between coordination quality and dropout severity, offering practical guidance on the conditions under which ML-based coordination remains effective and where its limitations become significant.

5.2 Limitations

Despite its strengths, the proposed approach exhibits several limitations that must be acknowledged to properly contextualize the results and guide future development.

First, the accuracy of the ML forecasts is inherently dependent on the quality and quantity of historical consumption data. For newly installed agents or those with limited operational history, the absence of sufficient training data can severely undermine forecast accuracy. This issue is particularly pressing in rapidly evolving environments where behavioral patterns or occupancy schedules change abruptly, such as during seasonal transitions or significant lifestyle changes within households.

Second, while the Random Forest models employed in this study were effective and computationally efficient, they are static and cannot adapt in real time. Once deployed, the models are not updated with new observations, meaning that sustained disconnection can lead to a growing divergence between predicted and actual consumption behavior. Moreover, the forecasts do not account for reactive responses to price signals, as the models are trained on historical behavior rather than on simulations of adaptive agents.

Third, although the simulation framework incorporates realistic features such as weather-driven heating loads and diverse building thermal properties, it simplifies several aspects of communication dynamics. It assumes synchronized updates and does not model asynchronous behaviors, packet loss bursts, or the stochasticity of real-world network conditions. Additionally, human intervention (such as manual overrides of heating schedules or the unplugging of devices) was not modeled but could significantly affect system performance during outages.

A key limitation, acknowledged in the text, is the use of static, pre-trained machine learning models. These models do not adapt to changing conditions or learn from the ongoing coordination process. Consequently, predictions for non-responding agents are based on past behaviors, not on how those agents would have responded to current price signals. This discrepancy could lead to a divergence between the predicted optimal behavior and the actual behavior, especially under volatile conditions.

5.3 Future Work

To build upon the foundation laid by this research, several promising avenues can be pursued to enhance system resilience, forecasting accuracy, and real-world applicability.

One critical extension is the development of adaptive or online learning models that can update their parameters based on recent observations. Techniques such as incremental learning or streaming gradient descent could be used to ensure that ML models remain aligned with actual agent behavior during extended periods of disconnection. Additionally, incorporating probabilistic forecasting, through ensembles or Bayesian approaches, would allow the coordinator to quantify uncertainty in predictions and weigh surrogate profiles accordingly during optimization.

Another important direction involves the deployment of federated learning frameworks. These would enable agents to collaboratively train shared models across the neighborhood without transmitting raw consumption data, thus preserving privacy and reducing network traffic. Federated approaches could be especially beneficial in heterogeneous environments with shared load patterns but individualized behaviors.

Improving the coordination mechanism itself is also a key area for advancement. The current ADMM formulation assumes synchronous communication and homogeneous participation rates. Asynchronous consensus algorithms or gossip-based coordination protocols could offer increased robustness to irregular agent availability and varying communication delays.

While ADMM is a distributed algorithm, the coordinator remains a central component for aggregating data and calculating prices. Future work should include a study on the scalability of the coordinator and potential bottlenecks as the number of agents increases significantly. Research into hierarchical or fully decentralized coordination mechanisms could mitigate these risks.

Furthermore, extending simulations to include multi-energy systems, such as coupling with electric vehicle charging, battery storage, or photovoltaic generation, would allow the framework to support broader energy flexibility goals. Finally, validating the proposed framework in real-world testbeds or through partnerships with utilities will be essential for assessing scalability, interoperability, and user acceptance. Such deployments would provide critical operational data and uncover new constraints not captured in simulation.

Future research should also explore more complex failure scenarios, such as network partitions that isolate groups of agents, and expand the concept of resilience to include other sources of uncertainty, such as the variability of renewable energy sources and unexpected fluctuations in demand.

Bibliography

[1] Elisa Papadis and George Tsatsaronis. "Challenges in the decarbonization of the energy sector". In: *Energy* 205 (2020), p. 118025. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2020.118025. URL: https://www.sciencedirect.com/science/article/pii/S0360544220311324.

[2] Muhammad Khalid. "Smart grids and renewable energy systems: Perspectives and grid integration challenges". In: *Energy Strategy Reviews* 51 (2024), p. 101299. ISSN: 2211-467X. DOI: https://doi.org/10.1016/j.esr.2024.101299. URL: https://www.sciencedirect.com/science/article/pii/S2211467X24000063.

[3] Diana María López González and John Garcia Rendon. "Opportunities and challenges of mainstreaming distributed energy resources towards the transition to more efficient and resilient energy markets". In: *Renewable and Sustainable Energy Reviews* 157 (2022), p. 112018. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2021.112018. URL: https://www.sciencedirect.com/science/article/pii/S1364032121012818.

[4] Oluwaseun O. Tooki and Olawale M. Popoola. "A comprehensive review on recent advances in transactive energy system: Concepts, models, metrics, technologies, challenges, policies and future". In: *Renewable Energy Focus* 50 (Sept. 2024), p. 100596. ISSN: 17550084. DOI: 10.1016/j.ref.2024.100596. URL: https://linkinghub.elsevier.com/retrieve/pii/S1755008424000607.

[5] Muhammad Faizan, Ghuffran Ahmed, and Farrukh Hassan Syed. "Trends and Challenges of Transactive Energy Management". In: *2022 International Conference on Emerging Trends in Smart Technologies (ICETST)*. 2022, pp. 1–6. DOI: 10.1109/ICETST55735.2022.9921424.

[6] Jiayong Li et al. "Distributed transactive energy trading framework in distribution networks". In: *IEEE Transactions on Power Systems* 33.6 (2018), pp. 7215–7227. DOI: 10.1109/TPWRS.2018.2854649.

[7] Muhammad Saad et al. "Machine Learning Based Approaches for Imputation in Time Series Data and their Impact on Forecasting". In: *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2020, pp. 2621–2627. DOI: 10.1109/SMC42975.2020.9283191.

[8] Alain Aoun et al. "From Bottom-Up Towards a Completely Decentralized Autonomous Electric Grid Based on the Concept of a Decentralized Autonomous Substation". In: *Electronics* 13.18 (2024). ISSN: 2079-9292. DOI: 10.3390/electronics13183683. URL: https://www.mdpi.com/2079-9292/13/18/3683.

[9] Maria Elena Valcher and Gianfranco Parlangeli. "On the effects of communication failures in a multi-agent consensus network". In: *2019 23rd International Conference on System Theory, Control and Computing (ICSTCC)*. 2019, pp. 709–720. DOI: 10.1109/ICSTCC.2019.8885721.

[10] Krzysztof Skrzypczyk. "On Multi Agent Coordination in the Presence of Incomplete Information". In: *Artificial Intelligence and Soft Computing – ICAISC 2008*. Ed. by Leszek Rutkowski et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1254–1265. ISBN: 978-3-540-69731-2.

[11] Jian Xie and Jian Liu. "A novel multi-agent system for complex scheduling problems using enhanced ACO algorithm". In: *Journal of Industrial Engineering International* 13.3 (2017), pp. 375–386. DOI: 10.1080/22348972.2017.1348890.

[12] Nicholas R Jennings, Katia Sycara, and Michael Wooldridge. "A Roadmap of Agent Research and Development". In: *Autonomous Agents and Multi-Agent Systems* 1.1 (1998), pp. 7–38. DOI: 10.1023/A:1010090405266.

[13] Klaas De Craemer and Geert Deconinck. "Balancing trade-offs in coordinated PHEV charging with continuous market-based control". In: *2012 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*. IEEE. IEEE, Oct. 2012, pp. 1–8. ISBN: 978-1-4673-2597-4. DOI: 10.1109/ISGTEurope.2012.6465685. URL: http://ieeexplore.ieee.org/document/6465685/.

[14] Angelia Nedic and Asuman Ozdaglar. "Distributed Subgradient Methods for Multi-Agent Optimization". In: *IEEE Transactions on Automatic Control* 54.1 (Jan. 2009), pp. 48–61. ISSN: 0018-9286. DOI: 10.1109/TAC.2008.2009515. URL: http://ieeexplore.ieee.org/document/4749425/.

[15] Stephen Boyd et al. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers". In: *Foundations and Trends® in Machine Learning* 3.1 (2011), pp. 1–122. DOI: 10.1561/2200000016.

[16] David Johnson and Emily Zhang. "Dynamic reconnection protocols for multi-agent systems in smart grids". In: *IEEE Transactions on Smart Grid* 10.4 (2019), pp. 4572–4583. DOI: 10.1109/TSG.2019.2894572.

[17] Sarvapali D. Ramchurn et al. "Decentralized Demand Side Management in Smart Grids". In: *Proceedings of the IEEE* 112.3 (2024), pp. 456–478. DOI: 10.1109/JPROC.2023.3356789.

[18] Stephen Boyd. "Distributed Optimization via ADMM". In: *Foundations and Trends in Machine Learning* 17.1 (2024), pp. 1–150. DOI: 10.1561/2200000076.

[19] R. Jabr, A. H. Coonick, and B. J. Cory. "A Primal-Dual Interior Point Method for Optimal Power Flow Dispatching". In: *IEEE Power Engineering Review*

22.7 (July 2002), pp. 55–55. ISSN: 0272-1724. DOI: 10.1109/MPER.2002.4312391. URL: https://ieeexplore.ieee.org/document/4312391/.

[20] Lukas Lanza, Timm Faulwasser, and Karl Worthmann. "Distributed Optimization for Energy Grids: A Tutorial on ADMM and ALADIN". In: *IEEE Transactions on Control Systems Technology* 22.6 (Apr. 2024), pp. 2233–2242. DOI: 10.1109/TCST.2014.2308392. arXiv: 2404.03946. URL: http://arxiv.org/abs/2404.03946.

[21] Christodoulos A. Floudas. "Generalized Benders Decomposition". In: *Encyclopedia of Optimization* 10.4 (2008), pp. 1162–1175. DOI: 10.1007/978-0-387-74759-0_201. URL: https://link.springer.com/10.1007/978-0-387-74759-0%7B%5C_%7D201.

[22] Smith. "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver". In: *IEEE Transactions on Computers* C-29.12 (Dec. 1980), pp. 1104–1113. ISSN: 0018-9340. DOI: 10.1109/TC.1980.1675516. URL: http://ieeexplore.ieee.org/document/1675516/.

[23] Ali Jadbabaie, Jie Lin, and A.S. Morse. "Coordination of groups of mobile autonomous agents using nearest neighbor rules". In: *IEEE Transactions on Automatic Control* 48.6 (June 2003), pp. 988–1001. ISSN: 0018-9286. DOI: 10.1109/TAC.2003.812781. URL: http://ieeexplore.ieee.org/document/1205192/.

[24] Wei Ren and Randal W Beard. "Consensus Seeking in Multi-Agent Systems Using Dynamically Changing Interaction Topologies". In: *IEEE Transactions on Automatic Control* 50.5 (2005), pp. 655–661. DOI: 10.1109/TAC.2005.846556.

[25] John F. Nash. “Equilibrium points in n -person games”. In: *Proceedings of the National Academy of Sciences* 36.1 (Jan. 1950), pp. 48–49. ISSN: 0027-8424. DOI: 10.1073/pnas.36.1.48. URL: https://pnas.org/doi/full/10.1073/pnas.36.1.48.

[26] Zhu Han et al. *Game Theory in Wireless and Communication Networks*. Cambridge University Press, Oct. 2011. ISBN: 9780521196963. DOI: 10.1017/CBO9780511895043. URL: https://www.cambridge.org/core/product/identifier/9780511895043/type/book.

[27] Zhen Liu et al. “Multi-agent based experimental analysis on bidding mechanism in electricity auction markets”. In: *International Journal of Electrical Power and Energy Systems* 43.1 (2012), pp. 696–702. ISSN: 0142-0615. DOI: https://doi.org/10.1016/j.ijepes.2012.05.056. URL: https://www.sciencedirect.com/science/article/pii/S0142061512002463.

[28] Zibo Zhao, Chen Feng, and Andrew L. Lu. *Auction Design through Multi-Agent Learning in Peer-to-Peer Energy Trading*. 2021. arXiv: 2110.10714 [cs.GT]. URL: https://arxiv.org/abs/2110.10714.

[29] Reza Olfati-Saber, J. Alex Fax, and Richard M. Murray. “Consensus and Cooperation in Networked Multi-Agent Systems”. In: *Proceedings of the IEEE* 95.1 (2007), pp. 215–233. DOI: 10.1109/JPROC.2006.887293.

[30] Reza Olfati-Saber, J. Alex Fax, and Richard M. Murray. “Consensus Algorithms for Multi-Agent Systems”. In: *Proceedings of the IEEE* 112.5 (2024), pp. 789–812. DOI: 10.1109/JPROC.2023.3389012.

[31] Kun Deng et al. “Structure-preserving model reduction of nonlinear building thermal models”. In: *Automatica* 50.4 (2014), pp. 1188–1195. ISSN: 0005-1098.

DOI: https://doi.org/10.1016/j.automatica.2014.02.009. URL: https://www.sciencedirect.com/science/article/pii/S0005109814000478.

[32] Piotr Michalak. "A thermal network model for the dynamic simulation of the energy performance of buildings with the time varying ventilation flow". In: *Energy and Buildings* 202 (2019), p. 109337. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2019.109337. URL: https://www.sciencedirect.com/science/article/pii/S037877881930787X.