SynergyGrids: Blockchain Assisted Peer-to-Peer Energy Trading and Prosumer Management Framework

by

Faizan Safdar Ali

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Computer Science and Engineering

KOÇ ÜNİVERSİTESİ

September 21, 2020

SynergyGrids: Blockchain Assisted Peer-to-Peer Energy Trading and Prosumer Management Framework

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Faizan Safdar Ali

and have found that it is complete and satisfactory in all respects,

and that any and all revisions required by the final

examining committee have been made.

Committee Members:

Prof. Öznur Özkasap (Advisor)

Prof. Attila Gürsoy

Prof. Arda Yurdakul

Date:

To my family and specially my late beloved father. He fought 5 hard and long years against cancer. I attribute all my successes to him.

ABSTRACT

SynergyGrids: Blockchain Assisted Peer-to-Peer Energy Trading and Prosumer Management Framework
Faizan Safdar Ali
Master of Science in Computer Science and Engineering
September 21, 2020

With depletion in fossil energy and increasing environmental problems, energy markets have shifted towards renewable energy resources, such as wind and solar. The ultimate aim is to bridge the gap between generation and demand. As these sources can be harvested on a small scale, increasing number of producers are becoming capable of selling surplus energy. This fostered the concept of peer-to-peer (P2P) energy trading and sharing, by encouraging consumers to become prosumers (i.e. capable of both producing and selling surplus energy). Nowadays, the concept of energy trading and sharing is no longer a dream due to the emerge of the smart city concept. However, due to the heterogeneity of these energy resources (i.e. difference in energy generation and dependency) and a low degree of reliability, the task of efficiently managing these energy resources becomes difficult. Various energy trading systems and management tools have been developed, however, they mostly use centralized cloud systems and face inherent problems of high latency, low scalability, and security problems. Hence, a strong tool that can protect the prosumer data using emerging technology, log the changes for audit purposes, and eventually improve the performance of the system is needed. In this context, blockchain technology as a distributed ledger has emerged as an alternative to centralized systems.

In this thesis, state-of-the-art P2P energy trading and sharing solutions are investigated, highlighting their benefits, drawbacks and open issues by classifying them into sub-categories of *Infrastructure-based Energy Trading, Large Scale Energy Storage Based Trading, and Ad hoc P2P Energy Trading.* Considering the investigated open issues, a blockchain assisted P2P energy trading and prosumer management framework, namely *SynergyGrids*, that consists of two main modules is developed. As the first module of the framework, a decentralized grouping mechanism, named *SynergyChain*, is proposed for improving scalability and decentralization of the pro-

sumer grouping mechanism in the context of P2P energy trading. Smart contracts are used for storing transaction information and for the creation of the prosumer groups. *SynergyChain* integrates a reinforcement learning module to create a self-adaptive grouping technique for improving the overall system performance and profitability. The proposed *SynergyChain* model improves the time complexity for each energy request as it reduces the search space of available energy from n (number of prosumers) to k (number of groups formed). As the second module, a hybrid trading system, named *FederatedGrids*, that manages both the prosumers from local microgrids and across the microgrids to allow energy trading and sharing is proposed. This allows consumers to buy energy from local resources and reduce reliance on utility grids. *FederatedGrids* proposes a decentralized blockchain-based federated learning model that is used to predict the energy demand of the consumers.

The proposed *SynergyGrids* modules are developed using Python and Solidity and have been analyzed using Ethereum test nets. The comprehensive experimental analysis using the Hourly Energy Consumption data-set has been conducted. *SynergyChain* shows a 39.7% improvement in the performance and scalability of the system as compared to the centralized system, and 18.3% improvement in the profitability of the prosumers. The blockchain-based federated learning to predict the demand load proposed in *FederatedGrids* shows comparable accuracy with an average 97.8% improved network load. The energy trading and sharing results show a 17.8% decrease in energy cost for consumers and a 76.4% decrease in load over utility grids when compared to other models from the literature.

ÖZETÇE

SynergyGrids: Blokzincir Destekli Görevdeş Enerji Ticareti ve Tüketici Yönetim Sistemi
Faizan Safdar Ali
Bilgisayar Bilimi ve Mühendisliği, Yüksek Lisans
21 Eylül, 2020

Fosil enerjinin tükenmesi ve artan çevre sorunları ile enerji piyasaları, rüzgar ve güneş gibi yenilenebilir enerji kaynaklarına yönelmiştir. Burada amaç, üretim ve talep arasındaki boşluğu kapatmaktır. Bu kaynaklar küçük ölçekte toplanabildiğinden, artan sayıda üretici artık enerjiyi satabilir hale gelmektedir. Bu, tüketicilerin iki rol üstlenmesini teşvik ederek (hem üretip hem de fazla enerji satabilen) görevdeş (P2P) enerji ticareti ve paylaşımı kavramını ortaya çıkarmıştır. Günümüzde akıllı şehir teknolojilerinin gelişmesiyle birlikte enerji ticareti ve paylaşımı kavramları hayal olmaktan çıkmaktadır. Bununla birlikte, bu enerji kaynaklarının heterojenliği (enerji üretimi ve bağımlılıktaki farklılık) ve düşük derecede güvenilirlik nedeniyle, bu enerji kaynaklarını verimli bir şekilde yönetme görevi zorlaşmaktadır. Geliştirilen enerji ticaret sistemleri ve yönetim araçları çoğunlukla merkezi bulut sistemlerini kullanır ve yüksek gecikme, düşük ölçeklenebilirlik ve güvenlik sorunları gibi doğal sorunlarla karşı karşıya kalırlar. Dolayısıyla, gelişen teknolojiyi kullanarak tüketici verilerini koruyabilen, değişiklikleri denetim amacıyla günlüğe kaydedebilen ve sistemin başarımını iyileştiren güçlü bir araca ihtiyaç vardır. Bu bağlamda dağıtık bir defter olarak blokzincir teknolojisi, merkezi sistemlere alternatif olarak ortaya çıkmıştır.

Bu tezde, son teknoloji P2P enerji ticareti ve paylaşım çözümleri, avantaj, dezavantaj ve açık problemleri araştırılarak *Altyapı Tabanlı Enerji Ticareti, Büyük Ölçekli Enerji Depolamasına Dayalı Ticaret ve Tasarsız P2P Enerji Ticareti* grupları önerilmiştir. İncelenen açık konular göz önünde bulundurularak, iki ana modülden oluşan, blokzincir destekli P2P enerji ticareti ve tüketici yönetimi çerçevesi *SynergyGrids* geliştirilmiştir. Sistemin ilk modülü olarak, P2P enerji ticareti bağlamında prosumer gruplama mekanizmasının ölçeklenebilirliğini ve ademi merkeziyetini geliştirmek için *SynergyChain* adlı dağıtık bir gruplama mekanizması önerilmiştir. Akıllı sözleşmeler,

işlem bilgilerini depolamak ve müşteri gruplarının oluşturulması için kullanılmıştır. *SynergyChain* , genel sistem başarımını ve karlılığını arttırmak için uyarlanabilir bir gruplama tekniği oluşturmak için bir pekiştirmeli öğrenme modülünü içermektedir. Önerilen *SynergyChain* modeli, kullanılabilir enerjinin arama alanını n'den (üretici sayısı) k'ye (oluşturulan grup sayısı) düşürdüğü için her enerji talebi için zaman karmaşıklığını iyileştirmektedir. İkinci modül olarak, enerji ticaretine ve paylaşımına izin vermek için mikro şebekelerden üretim yapanları yöneten *FederatedGrids* adlı karma bir sistem önerilmektedir. Bu sistem, tüketicilerin yerel kaynaklardan enerji satın almasına ve ana şebekelere olan bağımlılığı azaltmasına olanak tanır. *FederatedGrids* tüketicilerin enerji talebini tahmin etmek için kullanılan dağıtık blokzincir-tabanlı birleşik bir öğrenme modeli önermektedir.

Önerilen *SynergyGrids* sistem modülleri Python ve Solidity kullanılarak geliştirilmiş ve Ethereum test ağları kullanılarak analiz edilmiştir. Saatlik Enerji Tüketimi veri setini kullanan kapsamlı deneyler gerçekleştirilmiştir. *SynergyChain* , merkezi sisteme kıyasla sistemin başarımı ve ölçeklenebilirliğinde %39,7 ve tüketicilerin karlılığında % 18,3 iyileşme göstermektedir. Talep yükünü tahmin etmek için FederatedGrids kapsamında önerilen blokzincir-tabanlı birleşik öğrenme modeli, ortalama %97,8 iyileştirilmiş ağ yükü ile karşılaştırılabilir doğruluk sağlamakatadır. Literatürdeki diğer modellerle karşılaştırıldığında tüketiciler için enerji maliyetinde %17,8 ve ana şebekelere göre yükte %76,4 oranında azalma olduğunu göstermektedir.

ACKNOWLEDGMENTS

I would like to state my gratitude to my advisor Prof. Öznur Özkasap for her support and guidance that kept me motivated throughout my masters. During my stay at Koç University as a part of the Distributed Systems and Reliable Networks (DISNET) Research Laboratory, I have gained immense knowledge about research methods, organization, and project management. I am grateful to my thesis committee members Prof. Attila Gürsoy and Prof. Arda Yurdakul for their time and consideration on evaluating this thesis. I am also grateful to Dr. Moayad Aloqaily and Dr. Ouns Bouachir for supporting me through my masters and providing invaluable suggestions for the research problems. I also want to thank Assoc Prof. Alptekin Kupcu for giving me my first publication experience in Turkey. I am forever in the debt of all the professors who taught me courses here at Koç University for everything I learned from them.

I would like to thank my family for their support and love they provided me during this journey. There were some hard times when I fell in the pitch of despair, but my brothers, sister, mother, and especially my father was always there to cherish me and make me happy. Thought of going back home and making them proud kept me going towards my goal. I want to thank Kamran Ali and Seema Momin for helping me through one of the toughest times and encouraging me to continue my studies when I had lost all hopes.

My experience at the DISNET lab is invaluable. In the likes of great Ph.D. students like Sanaz, Yahya, and Beakal, I learned the art of faith and determination. Their presence in the lab was a great motivation for me to thrive towards my goals. I was able to learn many research-related skills just by observing them and talking to them on and off the campus. I like to thank all of my other lab mates including Waris, Karim, Umuralp for the good times we had, here is to all the Yemekhane and Nero

Coffee sessions we had. I am grateful to all the Pakistani communities especially Zaheer, Armaghan, Munam, Hammad, Zeeshan, Adil, Palwisha, and Najeeb for making me feel at home far from home. We had a great time playing cricket, exploring Istanbul, and especially having each other during the pandemic that hit the world hard. I want to thank all my Turkish friends including Osman, Buse, and Gokce for making my life easier in Turkey, especially helping with situations where I required a Turkish translator. Finally, I would like to thank my special roommates Ehsan, Jerom, and Rustam for being like my family throughout my stay at Bogazici Park Evleri.

I acknowledge that this thesis research was partially supported by KUTEM (Koç University Tüpraş Energy Center) and Grant ERF-20, UAE. The development and experiments are conducted using Remix and blockchain testbeds (Rinksby, Ropsten).

TABLE OF CONTENTS

List of Tables **xiv**

List of Figures **xv**

Chapter 1: Introduction **1**

1.1 Motivation 1

1.2 Contributions 3

1.3 Thesis Organization 5

Chapter 2: Background, Preliminaries and Related Work **6**

2.1 Blockchain 6

2.1.1 Architecture 6

2.1.2 Types of Blockchains and Consensus Protocols 7

2.1.3 Applications and Benefits of Blockchain 8

2.2 Ethereum Blockchain 9

2.2.1 Solidity 10

2.2.2 Ethereum Virtual Machine (EVM) 10

2.2.3 Smart Contracts 10

2.2.4 Gas and Decentralized Applications (Dapps) 11

2.3 Energy Trading 11

2.4 Related Work 12

2.4.1 Smart Grids 13

2.4.2 Grouping and Clustering 14

2.5 Blockchain and P2P Energy Trading 16

2.5.1 Blockchain in P2P Energy Trade 16

2.5.2 Blockchain Based Energy Trading Initiatives 17

2.6 Discussion 18

Chapter 3: Blockchain Assisted Models For Energy Trading 19

3.1 Blockchain Energy Sector: Common Principles 19

3.1.1 Design Goals 19

3.1.2 Entities 21

3.1.3 Payments, Rewards and Demand Response 22

3.2 Infrastructure-based Energy Trading 22

3.3 Ad hoc P2P Energy Trading 24

3.4 Large Scale Energy Storage Based Trading 26

3.5 Summary and Comparison 27

3.6 Challenges and Limitations 27

3.6.1 Inherent Limitations 27

3.6.2 Energy Sector 28

Chapter 4: SynergyChain: System Architecture and Algorithms 30

4.1 Problem Statement 30

4.1.1 Proposed Solution 32

4.2 Prosumer Grouping Concept 33

4.2.1 Problem Definition 33

4.2.2 Role of Blockchain 35

4.2.3 Heuristic Approach 37

4.3 Reinforcement Learning Module (LM) 38

4.3.1 Problem Statement 38

4.3.2 Mapping RL to Grouping Problem 41

4.3.3 Model Representation 42

Chapter 5: SynergyChain: Implementation and Results 45

5.1 Implementation 45

5.1.1 Smart Contract Implementation 45

5.1.2 Independent Transactions 46

5.1.3 Learning Module . . . 49
5.2 Experimental Setup and Results . . . 49
5.2.1 Profitability . . . 50
5.2.2 Performance . . . 51
5.2.3 Effect of Reinforcement Learning Module . . . 53
5.2.4 Comparison . . . 54
5.3 Gas Cost Evaluation . . . 55
5.4 Discussion and Future Work . . . 55

Chapter 6: FederatedGrids: Architecture and Algorithms 57
6.1 Problem Statement . . . 57
6.1.1 Proposed Solution . . . 58
6.2 System Architecture . . . 59
6.2.1 Model Components . . . 59
6.2.2 Microgrid Cluster Definition . . . 60
6.2.3 Use of the Blockchain . . . 61
6.2.4 Interaction Between System Components and Participants . . . 62
6.3 Energy Trading . . . 63
6.3.1 Energy Pricing for Energy Trading . . . 64
6.4 Energy Sharing . . . 68
6.4.1 Limitations in Energy Sharing . . . 69
6.4.2 Energy Sharing Workflow . . . 70
6.5 Federated Learning . . . 70
6.5.1 Mapping FL to Energy Trading/Sharing . . . 71
6.5.2 Improvement in Network Load . . . 71
6.5.3 Federated learning and Blockchain . . . 72
6.5.4 Other Applications . . . 73

Chapter 7: FederatedGrids: Implementation and Results 76
7.1 Smart Contracts . . . 76

7.2 Experiments and Results 78
7.2.1 Federated Learning Simulation Setup 82
7.2.2 Federated Learning Results 83
7.2.3 Cost and Profit Analysis in Energy Trading 84
7.2.4 Energy Sharing Results 85
7.2.5 Network Performance 86
7.3 Gas Cost Evaluation 87

Chapter 8: Conclusion **89**
8.1 Remarks 89
8.2 Future Directions 90

Bibliography **92**

LIST OF TABLES

2.1 Energy trading literature review summary. 15

3.1 Comparison of the energy trade models. 28

4.1 Related work comparison of prosumer management solutions and summary. 32
4.2 Notation used in the article 36
4.3 The Performance Metrics 40

5.1 Group request time, throughput, request time, failed requests and grouping time. 51
5.2 Gas cost and price in Euro for different transactions. 55

6.1 Federated learning related literature review summary. 72
6.2 Resulting RMSE and MAPE for global models in the considered scenario with 20 nodes and 5 rounds. 73

7.1 Gas cost and price in Euro for different transactions. 88

LIST OF FIGURES

2.1 Sample blockchain blocks and transactions 7
2.2 Sample blockchain blocks and transactions 8
2.3 P2P energy trading system models 15
2.4 Overview of energy trading 15

3.1 Infrastructure-based P2P Energy Trading 23
3.2 Ad hoc P2P Energy Trading 23
3.3 Large Scale Energy Storage Based Trading 23

4.1 Individual and group P2P energy trading 32
4.2 Example: individual and group P2P energy trading 33
4.3 Blockchain-enabled vs centralized grouping 35
4.4 Architecture overview of SynergyChain 41

5.1 Average profit comparison with [Mengelkamp et al., 2018a,Sabounchi and Wei, 2017, Da Silva et al., 2013]. 47
5.2 Profit increase with LEP. 47
5.3 Execution time comparison with [Mengelkamp et al., 2018a,Sabounchi and Wei, 2017] for 1000 energy requests. 47
5.4 Profit increase of LEP and decrease of HEP after grouping. 52
5.5 Difference of profit decrease 52
5.6 Transactions happening per second. 52
5.7 Transaction time 52
5.8 Average response times 52
5.9 Average time for group info. 52

6.1 System architecture and interaction. 59

6.2 Interaction Between System Components and Participants. 74

7.1 Predicted energy load for per hour. 79
7.2 Total network gain achieved. 79
7.3 Training vs Validation loss 79
7.4 Energy Load of each microgrid. 79
7.5 Energy generation of each microgrid. 79
7.6 Average energy price with 5 microgrids. 79
7.7 Total energy sale with 5 microgrids. 80
7.8 Load over utility with 5 Microgrids. 80
7.9 Total consumer cost with 5 microgrids. 80
7.10 Average energy price with 3 microgrids. 80
7.11 Load over utility with 3 Microgrids. 80
7.12 Total energy sale with 3 microgrids. 80
7.13 Total consumer cost with 3 microgrids. 80
7.14 Total consumer cost with 5 microgrids and varying utility energy price. 80
7.15 Average energy price with 5 microgrids and varying utility price. . . . 80
7.16 Average request completion time varying number of energy requests. . 81
7.17 Average request completion time varying prosumers. 81
7.18 Average request completion time varying number of grids. 81
7.19 Fulfilled share requests with small and large energy requests. 81
7.20 Energy shared within and across microgrid. 81
7.21 Energy shared and traded by each migrogrid. 81
7.22 Prosumer sharing energy within grid and across grids. 85
7.23 Total Energy shared and traded per prosumer for grid 3. 85

Chapter 1

INTRODUCTION

1.1 Motivation

The global economy is experiencing significant growth of businesses with increasing demands for energy. Global energy consumption increased by 2.1% in 2017, and according to 2019 US Energy Information Administration (EIA) projects the expected growth in Asia by 2050 will be 50%. Since it is unlikely these demands will be met by conventional energy systems, the world is moving toward renewable energy systems (RES) supported by cyberphysical technologies [Muller, 2017]. In addition, climate changes increase the need for reduced dependency on conventional resources, several nations are aiming at increasing the dependency on renewable energies. Owing to this, there has been an increase in the share of renewable energy towards the total energy of the world overall. The shift from conventional energy sources to renewables is clear, as revealed in 2018 when 30% of the UK gross electricity consumption was generated by RES (According to Green Match report on Renewable Energy in the United Kingdom). This transformation has led to the emergence of prosumers (producers and consumers) which are capable of both generating and consuming energy. Many households and buildings are now equipped with solar panels that can contribute to the overall energy needs. Power systems need to adapt to renewable energy so they can operate efficiently and sustainably, and promote the concept of smart grids to support power systems.

The emergence of prosumers and smart grids creates opportunities for energy trading transactions that can be done between entities such as prosumers, grids, and energy storage. As energy is the most critical component for growing the economy, this paradigm shift in energy trading requires a system that is secure, efficient, and fosters energy economics. Moreover, the trading systems should become more decentralized, which will provide the most secure market for additional participants. Blockchain is a promising technology that offers a distributed, robust, secure, and private framework for P2P energy trading [Dorri et al., 2019]. In a recent report, Navigant Research reported that *Energy Blockchain Applications* are expected to have a compound annual growth rate of 67% over the next decade. The energy sector can deploy blockchain for quick payments, secure energy transactions, and privacy [Knirsch et al., 2018]. Looking at these advantages, the governments around the world are now turning towards P2P energy trading and the use of blockchain in management. Recently, Australia has run a small scale trial in Western Australia for low-cost trading [Colthorpe, 2020]. On the other hand, Japan has started building Blockchain backed systems to enhance "the safety of communication." [Contributor, 2020]. India has also successfully deployed blockchain for minimizing investment risks [Veerapaneni, 2020] and to trade solar energy [Xavier, 2020]. Other than households and buildings, the UK has also started to provide blockchain solutions for charging electrical vehicles through energy trading [Engineer, 2019].

Industrial investments into distributed energy resource technologies are increasing and playing a pivotal role in the global power mix, as part of a wider drive to provide a clean and stable source of energy. The management of prosumers that consume as well and generate energy, with heterogeneous energy sources is critical for sustainable and efficient energy sharing procedures. Thus, *Energy as a Service* is gaining abundant attention for its endless benefits to network and service provision. The idea of grouping the prosumers for better profits is a promising approach for prosumer management. Currently, the data collection and grouping is done in a centralized manner; that face trust, security, and scalability issues. Hence, a strong tool that can protect the prosumer data; log the changes for audit purposes, and eventually improve the performance of the system is necessary. SynergyChain (details in chapter 4) is proposing a blockchain-assisted adaptive model, namely SynergyChain, for improving scalability and decentralization of the prosumer grouping mechanism in the context of Peer-to-Peer (P2P) energy trading. Smart contracts are used for storing transaction information and for the creation of the prosumer groups.

Growing intelligent cities have seen an increasing amount of local energy generation through reusable energy resources. The trade of energy between these local energy generators and consumers can reduce the energy consumption cost and also reduce the dependency on conventional energy resources. However, these local energies might not be enough to fulfill energy consumption demands. A hybrid approach, where consumers can buy energy from both local prosumers (that generate energy) and also from prosumer of other locations, is required. Moreover, the energy needs of resources lacking consumers can be fulfilled with energy sharing. A centralized system can be used to manage this energy trading that faces several security issues and increase centralized development cost. In this work (details in chapter 6), a smart contract-based hybrid energy trading and sharing system have been proposed as a solution that reduces the average cost of energy and load over the utility grids. The system autonomously switches between trading and sharing depending on the load of the system that is calculated through federated learning implemented over the smart contract.

1.2 Contributions

In this thesis, we propose a decentralized blockchain based framework, SynergyGrids, for P2P energy trading and prosumer management. The modules of the framework; SynergyChain is used to create prosumer groups and the FederatedGrids provides platform to trade energy within and across a microgrid. Both of them aim towards improved profitability and consumer satisfaction in decentralized manner. The main contributions of the SynergyGrids framework are:

- A novel management system (SynergyChain) for creating prosumer groups using grouping has been proposed. The grouping mechanism is defined over the smart contract that can be executed by any participant in the network. A heuristic-based grouping algorithm is proposed considering grouping-goal for group creation. This grouping technique can be used to both create a coalition between the prosumers and the microgrids. A generic approach, that takes in the information of the participants and creates grouping, is proposed.

- A reinforcement learning (RL) module is proposed to create a self-adaptive grouping mechanism. The experiments were done by embedding this module within the grouping algorithm. The module takes profitability, performance and consumer count as an input and makes a decision if the group would be created in the next time step.

- A blockchain-based trading system (FederatedGrids) is proposed involving prosumers, consumers, and microgrids hence the proposed model utilizes all available energy to minimize the total energy cost of the consumer. A pricing mechanism that ensures the increase in utility of the participants is utilized. Most of the current energy trading models consider the profit maximization of an individual hence does not consider the effect of the price of energy on the overall system. This work calculates the energy price considering different factors that reduce the price in comparison to utility grid energy prices.

- An autonomous system that works on prediction values, through federated learning, is presented. The provisioned solution helps the utility grid by decreasing its load. The saved energy can then be utilized in low energy areas for better management. A matching mechanism is developed that takes into account energy requirements and match buyers with the sellers. To the best of our knowledge, FederatedGrids is the first attempt to create hybrid energy trading and sharing platform and to use federated learning over the smart contract for energy demand prediction.

- The effect of hybrid trading is discussed making this work reference or state-of-the-art for future works. This system eliminates the main grid monopoly by decentralizing the energy trading markets and creates an environment where local consumers can get energy at lower prices. The decentralization is achieved by designing self-enforcing smart contracts for creating grouping mechanism and devising transactions between entities hence reducing third party dependency and improving decentralization.

- The proposed mechanisms have been developed using Python and Solidity and analyzed using Ethereum test nets. Extensive experiments and evaluations using real-time data were performed. An implementation framework of prosumer grouping and energy trading using centralized and blockchain architecture has been developed and experimented using *Ropsten*, *Rinkeby* and *Python Web3* for performance comparison.

- The comprehensive analysis using the Hourly Energy Consumption data-set shows a 39.7% improvement in the performance and scalability of the system as compared to the centralized system, and 18.3% improvement in the profitability of the prosumers. The results show comparable accuracy with average *97.8%* improved network load. Also, the trading/sharing system shows *17.8%* decrease in energy cost for consumers and *76.4%* decrease in load over utility grids.

1.3 Thesis Organization

The rest of the thesis is organized as follows. Chapter 2 explains blockchain concepts, limitations, and decentralized applications that can be used for different use cases, especially in energy trading. Chapter 3 provides a review of energy trading and existing energy trading solutions organized by their use case. Chapter 4 explains the motivation, system design and architecture of SynergyChain, and Chapter 5 discusses results explaining each module of the system. In Chapter 6, FederatedGrids details and the efficacy of local and among grid energy trading and sharing are explained. Chapter 7 includes performance analysis and results of FederatedGrids components. Finally, Chapter 8 concludes with a summary of the contributions and future directions.

Chapter 2

BACKGROUND, PRELIMINARIES AND RELATED WORK

2.1 Blockchain

The blockchain concept involves development of an immutable and distributed chain-like data structure to securely store data that can be verified as required. Blockchain also offers strong consensus features that allow the ledger to be distributed among all participants. The ledger consists of blocks and each block has a number of transactions; this is known as the block size.

2.1.1 Architecture

Blockchain is set of transactions on distributed ledger (figure 2.1) that are split into blocks and are distributed over the network for validation and verification. The important parts of blockchain are discussed in this section.

Hash Function: Blockchain is connected using a collision resistant hash function that takes as an arbitrary input and gives an output with same length. The hash function is collision resistant because it is a necessary trait for integrity and immutability of the blockchain

Transactions and Blocks: The transaction is a date or information that can be of any type. The 'transaction' term was first used by Bitcoin [Nakamoto, 2008]. the transaction of Bitcoin contains the amounts transferred from sender to receiver and their respective addresses. The miners in the blockchain network are responsible for creating blocks. A block is a set of valid transactions with a timestamp and a hash that refers to the previous block. The very first block in every blockchain is Genesis Block containing the universal fact that can be verified. Figure 2.2 presents a sample example of connected blocks. The transactions are kept in a Merkle tree

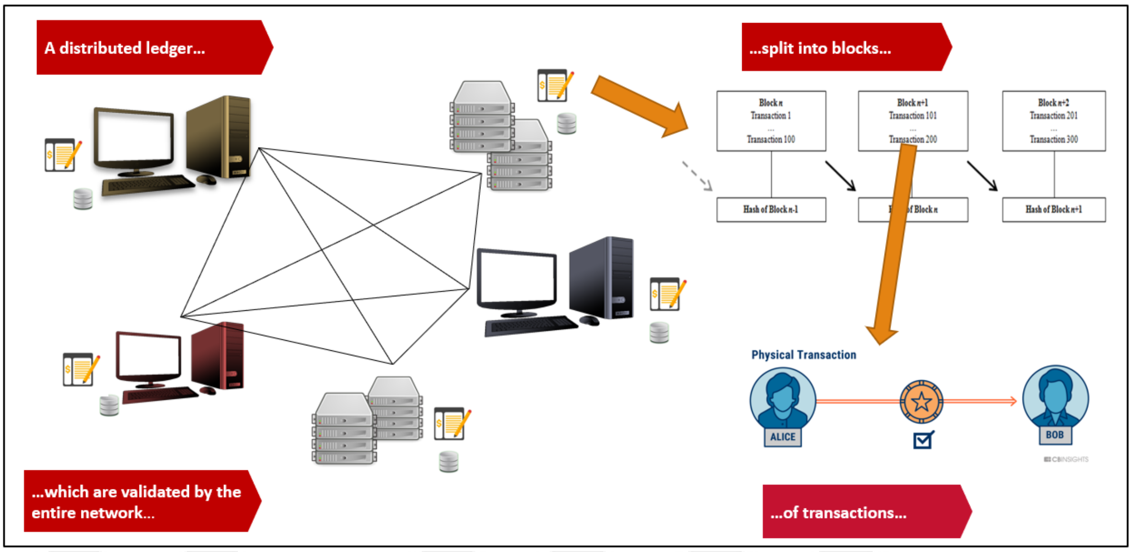

Figure 2.1: Sample blockchain blocks and transactions

[Merkle, 1980]. As can be seen, every block contains the hash of the preceding block (*prev_hash*) making blockchain computationally hard to be modified. Changing one block would require to change all the succeeding blocks to make blockchain valid.

2.1.2 Types of Blockchains and Consensus Protocols

The blockchain constitutes several types of entities. These entities can be reader, writer or both. The readers access the blockchain but do not participate in creation of the blocks whereas the writers can also create the blocks. Depending upon the activities an entity can perform, the blockchain can be divided into two types, permissioned (like Zerocash [Sasson et al., 2014], where anyone can read blockhchain and can update it as well) and permissionless blockhchain where each entitiy has set of permissions ((e.g., Hyperledger [Androulaki et al., 2018])).

Both types have their advantages and disadvantages depending upon the application. Permissionless blockchains allow more entities to be part of the network but are require a validator to authorize the transaction. This process is time taking and expensive. Also, it is prone to Sybil attacks [Douceur, 2002], where a network tries to appear as different nodes by creating fake identities. In contrast, permissioned systems use a participant list that speeds up the validation process and decreases

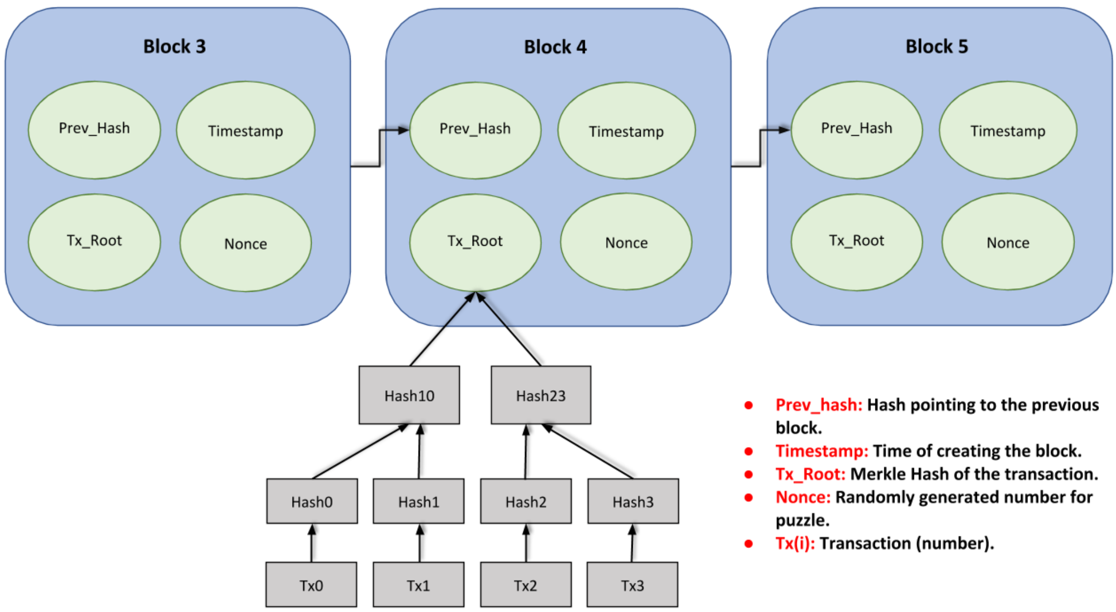

Figure 2.2: Sample blockchain blocks and transactions

the cost of maintenance, but it limits the number of participants in the blockchain. SynergyGrids utilizes permissionless blockchain, Ethereum, as it has larger network and regularizations that decreases the possibility of potential attacks.

Blockchain provides trust in untrustful enviornments. It replaces third party and provides the consensus protocols to maintain reliable and consistent state throughout the network. There are several consensus protocols, but proof-of-work is the most acceptable out of them. In PoW, a node can get its block accepted if it can solve a cryptographic puzzle (hash) and spend some computational resources in the process. It was first implemented by Bitcoin [Nakamoto, 2008]. The consensus protocol is important when it comes to the speed and scalability of the system. For this thesis, different Ethereum test nets are used that deploy distinct consunsus protocols.

2.1.3 Applications and Benefits of Blockchain

Blockchain technology can be applied in many fields and use cases. Cryptocurrencies are by far getting the most benefit from the concept. A lot of cryptographic research has been done and is going on to find the application of blockchain in different areas. By far blockchain is adopted by P2P communication and storage, IoT, and for improvement of security of different applications.

Based on the disruptive capabilities of blockchain, it has found its application in a wide range of applications including fields of e-finance, healthcare, smart governance, supply chain, and energy sector. In financial markets, blockchain is being used as payment methods and smart contracts while the healthcare department uses blockchain as an alternative of databases for record-keeping for better security and autonomy. The supply chain uses the advantage of traceability and accountability characteristics of blockchain to improve supply chain management solutions. E-voting and passport services are two of many applications that are being considered in the development of smart governance systems using blockchain. The application of blockchain in the energy sector is discussed in chapter 3.

Blockchain offers various advantages in the development of secure and trustless infrastructures. In general, blockchain provides the following benefits:

1. Credibility and Trustfulness: The chain can not be altered hence the transactions are immutable and verifiable.

2. Inherent Consistency: The data available on the blockchain is consistent throughout the network.

3. Authentication: Each participant in the blockchain has specific public/private key pair that helps in the authentication of the participants.

2.2 Ethereum Blockchain

Ethereum blockchain was first introduced in 2015 when Vitalik Buterin came up with an innovative idea to also execute the programs over the blockchain ledger, rather than just storing the transactions. The idea is to create a distributed mechanism where different machines can execute programs in a distributed manner. Hence, the Ethereum platform was created, following the concept of Proof of Work consensus introduced by Bitcoin blockchain. The platform could be used for the creation of functions, rules or clauses, terms, and conditions to carry out operations [Buterin et al., 2014].

2.2.1 Solidity

Ethereum platform provided a Turing complete programming language *Solidity* for the creation of the programs that can be executed over Ethereum blockchain. It is a javascript alike language over which variables, functions, and global states can be defined. The contracts consist of a combination of several such functions, each of which can be either public or private variable. The main advantage of Solidity over bitcoin is the execution of loops. Although *not* recommended, solidity language has the potential of running loops that creates easiness for developers. Solidity comes with its own limitation, such as no definition of floating-point numbers, but can facilitate the development and execution of any logic.

2.2.2 Ethereum Virtual Machine (EVM)

All the Ethereum programs are executed by Turing complete EVM that runs over every node in the network. In a distributed environment, a virus that makes its way to one computer can be replicated in other nodes in the system, hence bringing down the whole network. EVM handles this problem by running itself in a virtual sandbox that does not interact with the rest of the machine's functions. This boundary secludes the machine from other nodes and saves other nodes from getting effected.

2.2.3 Smart Contracts

Smart contracts are protocols that run on top of the blockchain to facilitate, verify, and implement the contexts of a normal contract with minimal human intervention [Szabo, 1997]. The very first idea of a smart contract was introduced in 1997. The concept behind smart contracts is like traditional contracts which "is executed when a specific conditioned is fulfilled." Building a house can be taken as an example of a contract. A house owner signs a contract with the builder that specifies when the house will be ready. Upon completion of the house, the builder receives its payment from the owner.

Traditional contracts are mediated by trusted third parties such as banks or governments. Ethereum smart contracts aim at removing this involvement of trusted parties and make the system more peer to peer (P2P) in nature.

2.2.4 Gas and Decentralized Applications (Dapps)

Usually, Ethereum transactions use Ether as a payment method, but to separate logical execution from finance, Ethereum has provided the concept of Gas the price of which does not fluctuate like the price of Ether. The developer now has to pay Gas for every execution over Ethereum blockchain.

The idea of Gas is also motivated by the cyberattacks that can happen over Ethereum. For example, an adversary can run an infinite loop over the user's account hence spending its Ether. When using Gas, the user can set a limit on how much gas can be spent in each transaction. Also, the idea of Gas encourages developers to create efficient and shortcodes.

The invention of smart contracts gave birth to a new field of development, Dapps. Dapps are similar to the web application in a way how their functions are executed. They have a user interface that can be used to interact with an application. The difference is that its back end is developed over Ethereum (or any other) blockchain-based smart contracts. An API is used in JavaScipt for the development of the front end. There are several applications that now provide services in decentralized fashion for better scalability and security [1].

2.3 Energy Trading

Energy trading is the sale and purchase of specific amounts of energy from the producer to the buyer [Ali et al., 2020d]. The overview of categories of energy trading is given in figure 2.3. Energy is conventionally traded wholesale, and most of it is transferred from big energy producers (e.g. coal mines, fuel energy generators) to distributors, who then forward the power to commodities such as houses, office build-

[1]https://www.stateofthedapps.com/whats-a-dapp

ings and farms. This concept is summarized in figure 2.4. With depletion in fossil energy and increasing environmental problems, the world started making progress towards sustainable and renewable energy such as wind and solar energy. Added advantage of these sources is that they can be harvested on a small scale. Today, many households and office buildings have solar panels installed on their premises, and this allows them to be producers capable of selling surplus energy to others. This fostered the P2P concept of energy trading, by encouraging consumers to become prosumers (i.e. capable of both producing and selling surplus energy). This means that more energy is available, and consequently reduces overall energy costs. This chapter provides background and concepts associated to energy trading. This chapter also includes brief description of energy trading techniques and models [Ali et al., 2020d] and showcases the difference between them.

2.4 Related Work

Different models for conducting P2P energy trade have emerged in the past few decades. Most of these models handle energy trade, prosumer management, and price optimization. Most of the proposed systems utilize technologies such as smart meters, cloud storage, and edge computing for development and implementation. Smart microgrids [Mengelkamp et al., 2018a] [Tsikalakis and Hatziargyriou, 2011] are early examples, where the prosumers are connected locally and can trade the energy with each other and with the other energy grids. [Zhang et al., 2018] propose peer to peer energy trade between the prosumers and consumers within the microgrid to create smart microgrids. A centralized system is used where the prosumers can sell energy to consumers through a bidding system. A similar approach is proposed by [Long et al., 2018] where local prosumer management is done within the residential prosumers in the locality. Another work [Lüth et al., 2018] creates a system with the objective of this community is to minimize the costs of electricity consumption from the transmission grid by prioritizing self-sufficiency. [Petri et al., 2020] proposes a rather different system to provide a trading platform. The proposed model introduces the concept of a federation for better price optimization.

The first hybrid approach was introduced by [Liu et al., 2017] where the trading is possible between both prosumers within microgrids and from the utility grids. Afterward, different game theoretical models for equilibrium creation were proposed. [Paudel et al., 2018] proposes a novel game-theoretical model for P2P energy trade where the buyer can adjust energy consumption behavior based on energy price. A price and seller competition is done in addition to the use of an M-leader and N-follower Stackelberg game approach to model the interaction between buyers and sellers. The results also show that P2P energy trading provides significant financial and technical benefits to the community, and it is emerging as an alternative to cost-intensive energy storage systems. Similarly, [Sabounchi and Wei, 2018] aims to create a Nash equilibrium using Game theory, hence reduce dependency on central grids. Most of these works have drawbacks that are faced due to their centralized nature.

2.4.1 Smart Grids

Distributed energy trading and prosumer management have been discussed in the literature. Smart microgrids [Mengelkamp et al., 2018a] [Tsikalakis and Hatziargyriou, 2011] are examples, where the prosumers are connected locally and can trade the energy with each other and with the other energy grids. Attempts to implement them over the blockchain have also been made. Grid+ [gri,] is one of the examples, that provides accounting and management systems to power trading with the help of the smart contract. However, These models are structurally limited to the local participants. In another work [Menniti et al., 2013], Energy district (ED)-based prosumer management has been presented. It is a center of energy consumption that contains many interconnected prosumers. The authors have proposed aggregator based demand-side management. In [Lukovic and Kovac, 2013], Multi-Agent System is introduced that can deal with different issues in the power system. The system consists of lower and higher system levels. A central Energy Management Systems (EMS) that enable prosumer to monitor and automate energy consumption and production has been presented in [Luna and et. al., 2016]. Prosumers in this community cooperate to improve the performance of the neighborhood. It consists of two EMSs; local and global. The local one is attached to each prosumer and EMS of each prosumer is attached to the central EMS. In this way, the prosumers in a community cooperate to improve the performance of the neighborhood. The drawback is the dependence on centralized authorities.

2.4.2 Grouping and Clustering

A set of prosumers contributes to a virtual power plant (VPP) is yet another attempt to manage prosumer [Rathnayaka et al., 2015] [Rathnayaka et al., 2014]. These models have the potential to provide opportunities for the local organizations, and communities to dynamically manage their energy needs depending on the participating resources. The shortcoming of these models is the lack of automation in the grouping. Multiple-criteria goal programming (MCGP) techniques are used where all the goals are given priority by the respective prosumers and the goal equation is formulated. The authors of [Vergados et al., 2016] provide a clustering solution for the group formation to minimize the aggregate imbalance and the penalties of each group. The techniques used are:

1. Spectral Clustering: This method does dimensionality reduction using eigenvalues of the similarity matrix before performing the clustering. Due to this, this method is relatively faster than other techniques of clustering.

2. Genetic Algorithm: These algorithms can be used for various sets of the problem with very few changes in parameters and can efficiently manage constraints. This technique is used for clustering while getting improved results.

3. Dynamic Clustering: As the name suggests, this method recalculates the clusters dynamically at runtime. The idea for its usage in prosumer clustering is for adaption of the cluster creation during runtime, such that the estimation errors and penalties are minimized.

The experiments conducted in [Ilić et al., 2013] shows that group size and individual prediction affect accuracy. The aggregation approach can be characterized as some

Table 2.1: Energy trading literature review summary.

Features	Entities	Energy trade	Drawbacks	Aim of the work	Technologies Used	ML Predication
[Petri et al., 2020]	Prosumers/Consumers /Cluster Manager	Between the clusters	Centralized	Price optimization	Smart Meters, EMS smart grids, cloud storage	No
[Khalid et al., 2020]	Prosumers/Consumers Utility grids	Between the peers and the grid	No among grids trade	Price Optimization	Blockchain, Smart Meters, EMS, smart appliances	No
[Liu et al., 2017]	Prosumer/consumer and utlility grids	P2P and P2G	Centralized , no Grid to grid	Reducing Monthly bill	Optimization, Smart Meters, EMS, smart grids	Yes, only 24 hours prediction
[Sabounchi and Wei, 2018]	Prosumer and consumer in a smart grid	the direct energy trading among energy consumers and prosumers.	Centralized	Nash equilibrium using Game theory, Reduce dependency on central grids	Game theory, Smart Meters, EMS, smart grids	No
[Long et al., 2018]	Prosumer and consumers	The residential prosumers in the locality	Low financial	Self-sufficiency of the MG, but causes lower financial turnover	Cloud, Smart Meters, EMS, Blockchain	No
[Zhang et al., 2018]	Prosumers	In community	No hybrid approach	Increase Speed, Scale and Security	Smart Meters, EMS smart grids, Blockchain	No
[Mylrea and Gourisetti, 2017]	Smart homes, cloud	Between smart homes in Microgrids	Centralized	Reducing unfair pricing using Pareto Optimality and reduce the computation complexity	Cloud computing,Smart Meters, EMS, smart grids	No
[Lüth et al., 2018]	Heterogeneous prosumer and consumer houses/ centralized storage	Between grid and P2P trade	Centralized	The objective of this community is to minimise costs ofelectricity consumption from the transmission grid by prioritising self-sufficiency	Smart Meters, EMS smart grids	No
[Paudel et al., 2018]	Prosumer in community microgrid	Between prosumers	Centralized	Reduction in cost of the price in micro-grid	Smart Meters, EMS smart grids	No
[Thakur and Breslin, 2018]	Microgrids	Between Microgrids	Yes	Microgrid collation	Smart Meters, EMS smart grids Blockchain	No

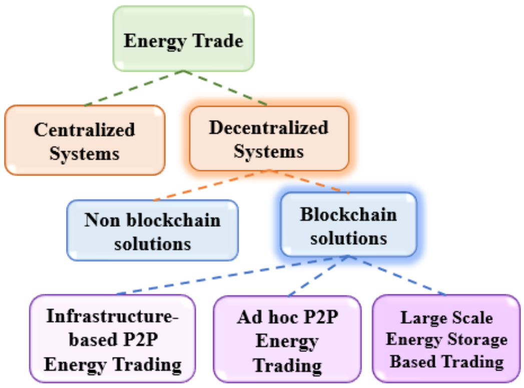

Figure 2.3: P2P energy trading system models

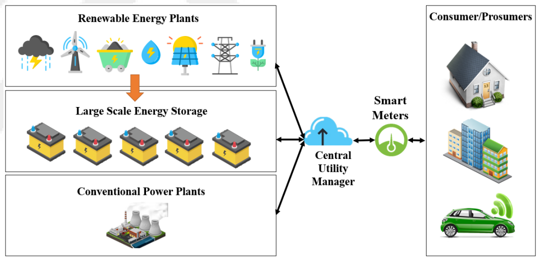

Figure 2.4: Overview of energy trading

kind of brute-force method [Ilić et al., 2013] where steps to grouping rely on random numbers. The paper uses the Monte Carlo method for the creation of a group of random smart meters with equally distributed probability (i.e. independent of group size) that ensure comparability between all group sizes. This work takes a large number of computational cycles that is a big challenge.

2.5 Blockchain and P2P Energy Trading

2.5.1 Blockchain in P2P Energy Trade

Energy trading requires management for secure energy trading and supply without any disruptions that can be achieved by a central utility manager, as depicted in Fig.2.4. However, a centralized infrastructure can cause issues, including single-point-of-failure, dependency on the central entity, and privacy concerns. This supports the transformation from centralized to decentralized models in cyberphysical systems.

Applying blockchain for energy trading has the potential to increase efficiency and security. There are several applications that can contribute to this, including grid management security and accountability, efficient utility billing, and P2P energy trading. With regard to security, blockchain can instill trust between parties which removes the dependency on centralized control and, due to immutability, provides integrity and a high degree of accountability. Moreover, blockchain supports multiple device types and reduces central intermediaries, which also helps reduce energy costs. In April 2016, the energy that was generated in a decentralized manner was sold directly between neighbors in New York via a blockchain system. This demonstrated that both energy producers and consumers can execute energy supply contracts without involving third-party intermediaries; thereby effectively increasing speed and reducing transaction costs [Power, 2017]. The steps involved in the procedure are as follows:

- Electricity is generated.
- Consumer buys energy.
- The transaction is updated on the blockchain via a smart meter.

Following the concept of blockchain in energy trade, [Khalid et al., 2020] uses blockchain for hybrid energy trading between consumers and prosumers, and utility grids. Bidding over the price is done before the start of the trade. The bid with the lowest price is selected as the trading price for the next timestep. During trading, the buying requests are matched with the available prosumers and the remaining amount is bought from te utility grids. The work uses Ethereum smart contracts for implementation and shows improvement in the energy price and performance of the system. [Thakur and Breslin, 2018] creates microgrids coalition for the microgrid to microgrid energy trading through blockchain. An asynchronous coalition formation method is proposed with is distributed and robust. Multiple coalition algorithms are executed to reduce computation time hence increasing the frequency of energy trade between the microgrids, The results shown, after successful simulation, are evidence that the proposed model achieves the desired results. Similarly, power ledger trailed blockchain energy trading in rural Australia, allowing commercial buildings to trade excess solar power. (https://www.ledgerinsights.com/power-ledger-blockchain-energy-trading-rural-australia/). A comparable idea is highlighted in [Mylrea and Gourisetti, 2017].

2.5.2 Blockchain Based Energy Trading Initiatives

Based on the design principles discussed earlier, there are several blockchain incentives that are working on different blockchain platforms including Ethereum, Hyperledger, and bitcoin. According to the report of IRENA 2019 [2], 36% of all the solutions in energy trading are providing energy trading. Moreover, almost 46% of these solutions reside in Europe. A detailed description of these initiatives can be seen in the IRENA report.

Other than these private initiates, the governments around the world are now turning towards P2P energy trading and the use of blockchain in management. Recently, Australia has run a small scale trial in Western Australia for low-cost trading [Colthorpe, 2020]. On the other hand, Japan has started building Blockchain backed systems to enhance "the safety of communication." [Contributor, 2020]. India has also successfully deployed blockchain for minimizing investment risks [Veerapaneni, 2020] and to trade solar energy [Xavier, 2020]. Other than households and

[2] https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2019/Feb/IRENA_Landscape_Blockchain_2019.pdf

buildings, the UK has also started to provide blockchain solutions for charging electrical vehicles through energy trading [Engineer, 2019].

2.6 Discussion

The use of blockchain technology in *P2P energy trading* has endless benefits to count. Several blockchain energy trading research is being conducted now. For instance, blockchain for the resilient market space for the prosumers has been presented in [Sabounchi and Wei, 2017]. They use a smart contract for the real-time P2P trade within the local markets. The concept of the aggregator is used for better energy management. Similarly, [Gai et al., 2019] uses the consortium blockchain concept for energy trading and crowd-sourcing for smart grids. Further, blockchain-based decentralized solution for demand management in [Pop et al., 2018]. A prototype is presented to validate and test the blockchain without the participation of the third party, this architecture decreases the overall cost and improves the reliability of the system. A very comprehensive review of various important use cases and business opportunities for blockchain innovation along with real case studies is recently published in [Andoni et al., 2019]. These papers provide proof of the concept of the blockchain impact on energy trading, however, they do not tackle the virtual grouping problem over the blockchain. Also, there is a need of creating a model that handles both P2P and M2M trading and sharing to have a positive impact on the environment by using all the energy within the system and reducing the load over utility grids. For that reason, we propose a *SynergyChain* and *FederatedGrids* mechanism that is elaborated in the following sections.

Chapter 3

BLOCKCHAIN ASSISTED MODELS FOR ENERGY TRADING

Motivated by the background of blockchain and its application in the energy trading, in this chapter, the common design principles in such environments are discussed. Then, we propose three decentralized cyberphysical blockchain-enabled P2P energy trading models [Ali et al., 2020d]. Building on that, we provide brief description of energy trading techniques and showcase the differences between them. This chapter concludes with blockchain limitations and discussion.

3.1 Blockchain Energy Sector: Common Principles

From the prior discussion, the choice of blockchain for the energy sector becomes evident. In this section, we will discuss how can blockchain be incorporated in energy trading and management.

3.1.1 Design Goals

Creating blockchain-based infrastructures requires defining a set of technical, regulatory, and economic design goals. These **technical design goals** include:

1. Decentralization: Blockchain systems do not depend on centralization, thus they can be used to create energy trading models with no central authority involved.

2. Scalability: The models must be scalable to include the sizes and geographical extents of new participants.

3. Heterogeneity: Due to the different types of devices, such as energy shared

between two cars or two houses, devices and systems need to be integrated into the models to support diversity. The system should be transparent and open to anyone who wishes to take part.

4. Intelligence: There are two aspects of intelligence. First, energy should be sold at the optimum (or stable) cost, there should be an intelligent bidding system and the client should have the power of choice. Secondly, energy requires effective management (e.g. demand and response).

5. IoT, Smart Devices, and Asset Management: IoT devices are the core components required by blockchain infrastructures. For example, electric vehicles are equipped with IoT devices and sensors that allow communication between autonomous vehicles, allowing them to trade energy between them.

The inclusion of the blockchain in the energy paradigm means regulatory challenges discussed in the previous section and the presented models have the following **regulatory design goals**. **Multi-source Cooperation:** For better collaboration between entities, privacy, and data management systems must be ensured. **Stakeholder interest:** Stakeholders include governments, national grid operators, microgrid owners, energy suppliers, and prosumers. To make optimal use of blockchain-based systems, the interests of all stakeholders must be incorporated into the models. For example, since so many policymakers are shifting to distributed energy, the conventional utilities are facing an uncertain future that can cause delays and disruptions in acquiring distributed energy.

The **economical goals** are as follows:

1. **Pricing:** Since energy trading can decrease the revenues of conventional energy players and disrupt the business of retailers, the pricing of the energy should be standardized.

2. **Energy markets:** in an economic zone where all parties can sell the energy. One potential pathway is P2P microgrids, which can aggregate prosumer supplies and provide ancillary services such as balancing support. Market com-

petition would benefit from an ancillary service market, and it would also be important for P2P microgrids as they could realize increased profit for voltage, frequency, restoration, peak load, and balancing support.

3.1.2 Entities

As shown in Fig.2.4, the entities involved in energy trading are divided into three groups, all of which can have access to smart meters and blockchain.

- **Central utility manager** are government, energy companies, and grid owners with the physical and technical infrastructure needed for energy sharing and transfers. They are responsible for regularization and global control of the system.

- **Energy generators** are conventional and renewable energy generators with large reservoirs of energy that provide energy for the network for national grid operators, microgrid owners, and solar cells and turbine holders.

- **Consumers and Prosumers** are energy users. Prosumers have extra capabilities that allow them to generate and sell surplus energy to other consumers in the system. Consumers/Prosumers These can be residential houses, electric cars or large buildings.

The transaction workflow is divided into three main categories:

1. **Energy transaction** includes all the communication and bargaining that takes place between buyers and sellers.

2. **Pre-trade communication** includes publishing users' supply and demands over the network. Different methods can be applied to ensure data privacy and anonymity [Wang et al., 2019a].

3. With **buyer-seller matching**, the seller makes a bid, and matching is conducted to identify the best price for the user. After matching, payment is performed using the transnational settlement method.

3.1.3 Payments, Rewards and Demand Response

To increase benefits for prosumers, consumers can pay with cryptocurrency [Kang et al., 2017]. This motivates prosumers since they will receive payments more quickly and have access to investment opportunities. Furthermore, there can be potential reward incentives for more active users. For example, if the prosumer is selling to the government the user can be rewarded in energy points, cryptocurrency, or by an adjustment to their bill.

Real-time control and management play an important role in P2P energy trading. Demand response is the transfer of energy from low demand users to high demand consumers. For example, in the morning, the household requires a lesser amount of energy than the office building. Similarly, at night the situation is reversed. Hence, the energy can be distributed according to the demand. The limitations of building such a management system and providing a smart contract-based demand response system are significant. This control unit is a major aspect of each of the three models defined herein.

3.2 Infrastructure-based Energy Trading

Traditional energy trading is managed by a centralized organization, but prosumers may not need centralized authority for P2P transactions. Where there are physical means of transferring energy, prosumers can directly communicate with one another. For example, two neighboring houses can connect via cable wire and the energy can be transferred directly. Another example is that two electric cars at the same station can transfer the energy without any intermediary party. In the infrastructure-based P2P energy trading model, it is assumed that prosumers are equipped with smart meters and have IoT devices installed to connect to whatever requires the energy (e.g. house, car). As illustrated in Fig 3.1, the devices communicate through the

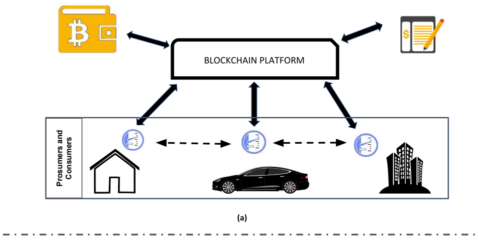

Figure 3.1: Infrastructure-based P2P Energy Trading

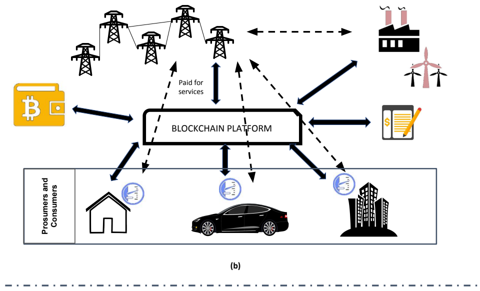

Figure 3.2: Ad hoc P2P Energy Trading

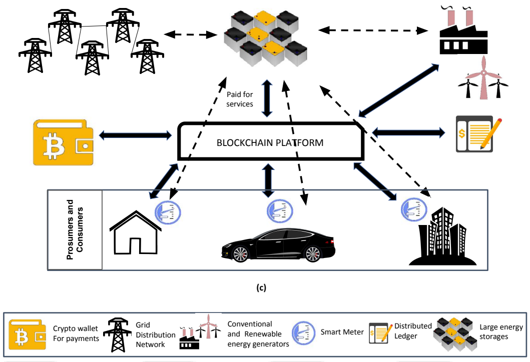

Figure 3.3: Large Scale Energy Storage Based Trading

blockchain to ensure successful transactions between entities. Another assumption is that prosumers have the physical means to transfer energy between them.

An architecture for pure P2P trading proposed in [Dorri et al., 2019] does not need third parties for the negotiations, since the prosumers and consumers communicate directly to create transactions. The architecture here fits this model, as it does not require grid stations during transactions, and if the prosumers have the physical means to transfer energy they can perform the transactions without involving intermediaries. This approach reduces communication from the blockchain layer since the negotiations are done on a separate network. For better security, two transactions have to be created within a time-period to be accepted. The solution provides better privacy and security at lower cost and time consumption.

Brooklyn microGrid [Mengelkamp et al., 2018a] in New York is another example that fits our infrastructure-based P2P energy trading model, as it is based on a small number of connected prosumers and consumers (e.g. 5 consumers/prosumers). The prosumers can sell their surplus energy to neighboring consumers using smart meters and e-wallets, and every member has access to all transactions via self-executing contracts. Users can specify the maximum payment they can afford, and they can prioritize the type of energy required (i.e. conventional or renewable).

The disconnected micro-grid explained here is premature and has technical challenges, such as adequate physical infrastructure for the energy transfer and the required global reach (e.g. when a seller is not near the buyer but still wants to sell their energy). This can lead to dysconnectivity and isolation of the prosumers, which in turn can cause irregular pricing in different locations. However, the model is efficient for local markets where the traffic in the system is low, and global reach is not required.

3.3 Ad hoc P2P Energy Trading

Considering the problems with conventional distributed energy trading and the blockchain-based model presented here (i.e. infrastructure-based P2P energy trading) energy trading must involve the grids and the infrastructure. Ad hoc P2P energy trading models have local micro-grids integrated with large scale energy producers through blockchain-based platforms (Fig 3b.). This allows consumers to buy energy from another prosumer or from conventional power plants. Due to the immutability and distributed nature of blockchain, the models ensure that all transactions are available to all the prosumers as well as the big energy companies, particularly governments. The platform can be provided by the government, which essentially gives them control over the energy sharing economy. This would mean all involved parties get more value and increased motivation. Since the distributors would be rewarded for these services, the Ad Hoc model also provides business opportunities for conventional distributors.

As shown in Fig 3.2, the structure of this model assumes smart devices are available for both prosumers and energy producers. The architecture is divided into four layers to enhance the development and analysis of the system:

1. **The energy grid** layers include all big companies and stakeholders that can

either create energy or are distributors.

2. **The communication layer** is where all the pre-bargain and communication is done. Here, the buyer initiates a request to buy energy, and the response can be through one of two protocols: matching or biding. The buyer is first matched with sellers according to the buyer's preference (e.g. minimum cost seller); the sellers submit their bids and the buyer commits to one. Another communication then takes place between the seller and the grids to establish the price of the services provided by the distributor.

3. **The blockchain layer** is responsible for all transactions and employs a smart contract-based system that is triggered whenever there is a transaction between two parties. Two separate ledgers are used, and a transaction between the buyer/seller and distributor also makes the system more robust, scalable, and efficient.

4. **The consumer/prosumer layer** contains all the consumers and prosumers and manages connections between members and the other layers through IoT devices.

An architecture that fits into our ad hoc P2P energy trading model and uses proof-of-stake based permission blockchain for efficiency is presented in [Siano et al., 2019]. Surplus energy is sold to other network participants via the energy manager, and a transaction controller makes energy bids and offers to buy or sell energy through the architecture; all transactions are retained by blockchain for accounting purposes. With regard to electric vehicles (EV), other research proposes [Knirsch et al., 2018] allowing EVs to use blockchain to locate a charging station, and the charging stations could bid to charge EVs. This process would help find the best price and location for both EV users and charging stations while providing privacy and security for the EVs. In this approach, there is no physical infrastructure but still, EVs can make benefit from energy trading.

3.4 Large Scale Energy Storage Based Trading

Prosumers can take advantage of the architecture available for energy transfers provided by the large scale energy storage entities by selling the energy to the local grids directly (Fig 3.3). This would require an agreement between the two parties that is recorded over the blockchain. In this case, the parties would be both prosumers and grids, so this model is under the paradigm of P2P energy trading.

A model that fits this category [Wang, 2019] provides a functional framework and model of energy distribution for crowdsourced energy systems, and also considers various types of energy trading transactions. The framework enables P2P energy trading at the distribution level, where distribution level asset owners can trade with one another. An operator is responsible for ensuring the transactions do not violate technical constraints. It considers two kinds of energy transactions, type 1 is where the prosumers feed for its usage. Type 2 is where the seller injects the power directly into the local grid. Type 2 kinds of transactions are controlled by the central authority. The model is two-phased. The first phase focuses on the day-ahead scheduling of energy generation, and controllable DERs manage the bulk of the grid operations. The second phase is developed to balance hour-ahead or real-time deficit/surplus in energy using monetary incentives. The developed two-phase algorithm supports arbitrary P2P energy trading between prosumers and utilities, which results in a systematic method to manage distribution networks while P2P energy trading is being conducted while incentivizing crowdsources to contribute to ecosystems.

This model motivates prosumers to participate in energy trading because there are no infrastructural costs to start selling energy. As well, the prosumer is not responsible for the biding and transactions. Scanergy [Mihaylov and et. al., 2015] research project uses this approach with prosumers assisting to balance supply/demand in return for rewards provided by the grid. This reward is given as an NRGCoin [Mihaylov and Jurado, 2014]. Prosumer energy exports are recorded by smart meters, and the prosumer is rewarded once its exported energy is used by another consumer. Though this approach can limit the control the prosumer has over the energy trading, it can also decrease management costs since the centralized energy storage systems are responsible for the distribution of the energy to the buyers.

3.5 Summary and Comparison

The purpose of these models is to address gaps in the current architecture. Table 3.1 provides a comparison of the blockchain solutions with traditional centralized and decentralized architectures present in the literature and highlights the blockchain improvements in security, heterogeneity, and openness of the systems. In the table, the terms high, medium and low are used for comparison purpose. For example, the blockchain provides high degree of decentralization whereas centralized systems such as utility grids and decentralized systems such as microgrids provide relatively lower decentralization. Similarly, the current centralized and decentralized solutions incorporate a certain type of participants, while in the proposed models, the participants can be buildings, houses or electrical vehicles hence providing high heterogeneity. There is also less central control in case of blockchain and notable efficiency improvements. Although decentralized models may provide better solution in terms of central control than the centralized model, there is still some degree of dependency over centralized authorities. For example, in case of microgrids, the control from utility is divided into small grids but still there exist central systems controlling such microgrids. Other than comparing with prior approaches, the table also indicates the significant differences between the three blockchain models that are summarized according to market approach, payment methods, and demand responses.

3.6 Challenges and Limitations

3.6.1 Inherent Limitations

Blockchain presents a new perspective on Internet security, but the traditional Bitcoin blockchain (which is used as the solutions discussed previous chapters) still has issues regarding huge power, energy, storage, and communication requirements [De Vries, 2018] (Bitcoin blockchain has been increasing over the years and is currently at around 200 GB). Moreover, the Bitcoin blockchain is secure assuming

Table 3.1: Comparison of the energy trade models.

	Blockchain Solutions			Non-Blockchian Solutions	
Features Models	**Infrastructure-based P2P Energy Trading**	**Ad hoc P2P Energy Trading**	**Large Scale Energy Storage Based Trading**	**Centralized**	**Decentralized**
Entities	Prosumers	Prosumers and grid stations	Prosumers, grids and energy storages	Prosumers, grids and energy storages	Prosumers, grids and energy storages
Single Point of Failure	No			Yes	
Energy Profile Anonymity	Yes			No	
Decentralization	High	High	High	Low	Medium
Blockchain	Smart Contracts	Smart Contracts	Smart Contracts hyperledger	-	-
Energy Agreement Verification	Through blockchain consensus between all nodes			By central authority	By distributed consensus
Energy trade	Pure P2P	Hybrid P2P	Hybrid P2P	-	Hybrid P2P
Rewards and Payment	Cryptocurrency or Energy coin	Cryptocurrency or Energy coin	Cryptocurrency or Energy coin	Visa/bank Transactions	Visa/bank Transactions
Heterogeneity	High	High	High	Low	Medium
Market Approach	Limited to local area	Limited to local city	Global Approach	Global Approach	Global Approach
Trust between trustless parties	High	High	High	Low	Low
Demand Response	Low	Low	High	By Central Authority (Medium)	
Central Control	Minimal	Partial	Complete	Complete	Partial

that more than 50% of the hashing power belongs to the honest parties in the system. There are further attacks on the incentive mechanism, such as selfish mining [Eyal and Sirer, 2018].

3.6.2 Energy Sector

Although blockchain is a promising technology for P2P energy trading, many challenges regarding its adoption have not yet been addressed.

Scalability and security: Blockchain still needs to demonstrate acceptable scalability, efficiency, and security for proposed cases. With the growing number of users of P2P energy trading, the implementation of a mature and secure blockchain architecture is critical.

Transaction and verification cost: Transactions become part of the blockchain after initial computation, and this can be a long process. VISA is capable of 20,000 [1] transactions per second, while most blockchains can only process a fraction of that number per second. Moreover, there is a transaction charge users must pay to create a transaction, and participating actors require a significant amount of synchronous

[1]https://usa.visa.com/dam/VCOM/download/corporate/media/visanet-technology/aboutvisafactsheet.pdf

storage for verification of the blockchain ledger; these factors increase the overall blockchain cost. As the number of participants in P2P energy trading increases, the number of transactions also grows and can overload the memory requirements for maintaining the ledger, as well as limit the efficiency of blockchain solutions. Other than cost, there is a transaction validation overhead. For example, Ethereum takes 15s to 5 minutes for transaction validation [2] which can affect overall speed of energy trading system.

Development cost: Another important challenge is blockchain development costs. For example, a transaction verifier needs high computation and network power, which is an extra cost not required for a traditional database system. Moreover, in the case of using smart contracts, there is a transaction cost that has to be paid by the participants.

Regularization: The blockchain process is beginning to demonstrate the potential of decentralized energy trading grids. However, the solutions presented in the literature have regulatory issues that include load balancing, integration with central control, and coordination with the main grids. For example, if the government wants to manage the power grids it is difficult for them to have greater control and more decentralization. In addition, when using the blockchain the price of energy trading should be regulated.

Government Limitations: The growth of P2P energy trading can lead to governments losing control over energy systems, and they want to retain control to allow for better policies and regularization. In addition, governmental intervention is crucial to supporting the development and commercialization of new technologies, while simultaneously creating markets with advanced energy systems. Blockchain provides P2P and distributed solutions that can limit governmental control, and this is a major challenge to blockchain adoption.

[2]https://ethgasstation.info/blog/ethereum-transaction-how-long/

Chapter 4

SYNERGYCHAIN: SYSTEM ARCHITECTURE AND ALGORITHMS

This chapter starts with the problem statement of what is grouping and how it is useful in the energy trading paradigm [Ali et al., 2020a]. Then a novel grouping goal along with the related algorithm and the mathematical formulation is discussed. The individual components of the SynergyChain [Ali et al., 2020c] system architecture is also explained. The chapter concludes with the explanation of the learning module and the motivation for using such a scheme.

4.1 Problem Statement

The introduction of prosumers has led to the emergence of peer-to-peer (P2P) energy trading where the energy transactions can be executed at the participants level; including prosumers, grids, and energy storage. This behavior can be seen in Figure 4.1 (Individual Trading Approach) where consumers can reach prosumers in a P2P fashion. However, many prosumers generate a very small portion of the overall energy and are often excluded during energy trades. This causes prosumers with lower energy generation to leave the network, which might lead to a rise in energy pricing. Under these circumstances, managing consumers and prosumers is becoming crucial within the energy sharing community. The formation of smart grids provides solution to these problems where many prosumers are attached through a dedicated infrastructure that facilitates accumulated energy sharing within the community. Prosumers within a smart grid together can provide a higher amount of energy for consumer demand satisfaction.

The smart grids increase profitability but are restricted to the locality of the prosumers. Moreover, this fixed architecture may introduce inflexibility as the prosumers may not leave or enter the system easily. For addressing this issue, a virtual prosumer grouping (VPG) system can be considered. In this, the prosumers may not be connected physically. Also, the prosumers are grouped to improve the mutual profit of the individuals. The VPG can create a dynamic ecosystem where the impact of the group behavior can be stronger than the individual entities. This dynamic grouping creates incentives for the prosumers that are in the areas where there is less amount of energy resources. Grouping behavior is illustrated in Figure 4.1 (Centralized VPG Approach), prosumers are connected through a centralized system, and all buying requests are processed through that system. Hence, all prosumers get an equal chance to sell the energy and collaboratively sell the energy. However, centralized strategies have the problem of scalability, a single point of failure, trust issues, high latencies, and a lower degree of decentralization. Blockchain is a candidate that can be used as the solution to the mentioned problems and can be used for industrial energy trading, as it allows fast, secure and reliable transfer of information [Wang et al., 2019d] that overcomes the problems faced by centralized systems. As can be seen in Figure 4.1, the introduction of blockchain promotes P2P energy trade without the need for central authority and hence has the potential to improve the overall performance of the system.

Although VPG already comes with many advantages; user satisfaction and profitability can significantly be increased by the use of a learning technique that can predict the profitability index and decide to dynamically dividing the prosumers into groups. In certain circumstances (discussed in section 4.3.1), prosumers might prefer to not participate in the grouping and sell the energy individually. Moreover, creating groups and doing transactions come with extra overhead that is reasonable if a large number of participants are contributing to the system, this behavior can also be learned based on which the decision of making groups can be taken. Hence, an adaptive approach not only has potentials to increase user satisfaction but can also improve the overall performance of the system. SynergyChain proposes to create VPGs using grouping techniques and blockchain to create robust systems against failures and to reduce load on the centralized grid while improving the overall system profitability.

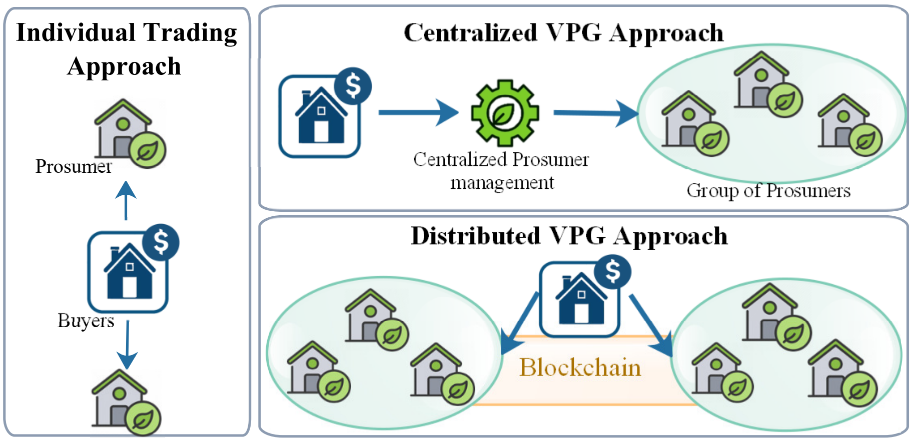

Figure 4.1: Individual and group P2P energy trading

Table 4.1: Related work comparison of prosumer management solutions and summary.

Solution	Participants	Aim of the work	Technologies Used	Adaptive Approach	Type of Goal
Individual P2P Trade [Mengelkamp et al., 2018a], [Tsikalakis and Hatziargyriou, 2011]	Prosumers/Consumers /Smart Grids	Better efficiency and decentralization for improved prosumer management and price optimization	Smart Meters, EMS smart grids, cloud storage. Blockchain	No	-
Energy Aggregation [Sabounchi and Wei, 2017]	Prosumers and consumers	Improving profit for prosumers in the trade	Smart Meters, EMS, smart appliances	No	Single Goal
Centralized Grouping [Rathnayaka et al., 2015], [Rathnayaka et al., 2014]	Prosumers	Multigoal prosumer grouping	Optimization, Smart Meters, smart grids	No	Multi Goal
Prosumer Management [Lukovic and Kovac, 2013], [Luna and et. al., 2016], [Menniti et al., 2013]	Prosumer	Enable prosumer to monitor and automate energy consumption and production	Smart Meters, EMS, smart grids	No	Single Goal
Random Grouping [Da Silva et al., 2013]	Prosumer and consumers	Study the effect of grouping on energy forecasting and trading	Smart Meters, EMS, smart grids	No	-
SynergyChain	Proumer and consumers	Better efficiency, decentralization and improving the profitability of the prosumers	Smart Meters, Smart Contract EMS, smart grids, MG	Yes	Single Goal

4.1.1 Proposed Solution

Our proposed mechanism is a platform for energy trading that uses blockchain for the creation of dynamic prosumer-consumer groups and uses blockchain where trusted prosumers that are part of the system can accept registration request of new prosumers for better trust and security. Blockchain is used for verification and group management that allows energy trading, in real-time. Moreover, SynergyChain incorporates reinforcement learning to create a self-adaptive grouping technique based on the prosumer/consumer behavior. Table 4.1 enlists the summary of energy trading prosumer management solutions and compares them with SynergyChain . It summarizes the participants (entities) involved, the aim of the work and type of technologies (for example smart meters, Blockchain technologies etc.) used in the

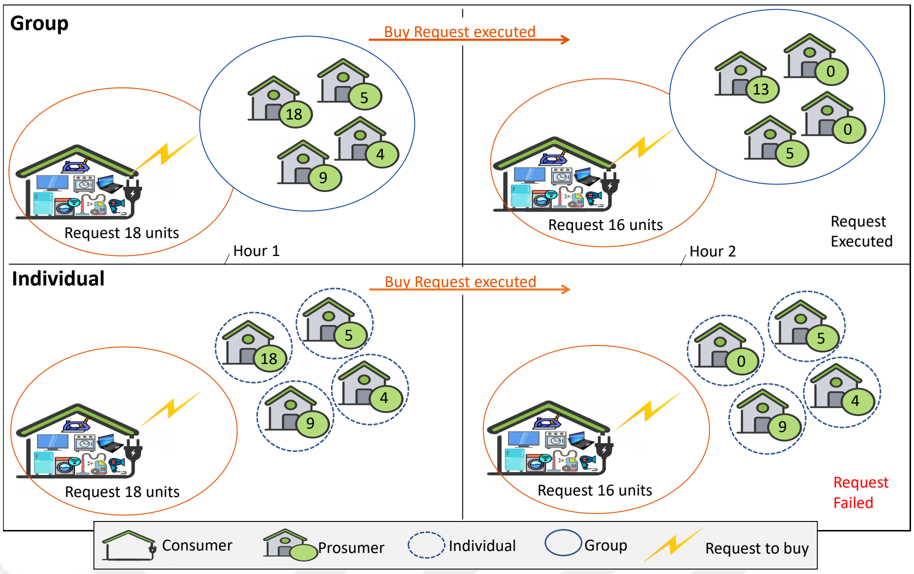

Figure 4.2: Example: individual and group P2P energy trading

solutions. The energy trade column explains amongst what entities, does energy trading happen? For example, in SynergyChain , energy trade happens between prosumer groups and the consumers. The table also shows if a solution uses an adaptive approach for better performance and the type of goal of the solution. In section 4.2, prosumer grouping concept details are provided. Section 4.3 describes the motivation behind using the learning module and its design choices.

4.2 Prosumer Grouping Concept

The concept of virtual grouping has been widely used for the management of the energy generators that are geographically distributed for the better management of the assets, operating expense reduction, and energy expenses, hence providing better services to the consumers [Rathnayaka et al., 2015] [Rathnayaka et al., 2014].

4.2.1 Problem Definition

The idea behind grouping is similar to the microgrids [Parag and Sovacool, 2016] in a way that both are used for the better management and orchestration of the energy sources. However, SynergyChain has significant differences in the concept, explained as follows:

1. Microgrids target large energy generators. SynergyChain targets small scale prosumers for collaborative trading of the energy.

2. Microgrids prosumers have to register with the centralized authority. SynergyChain handles the registration through smart contract that removes the centralized authority from the system.

The effectiveness of grouping is shown in form of an example in figure 4.2. As can be seen, in the first hour, all the energy requested by consumer is fulfilled by one prosumers while others had no participation. Also, in second hour, no prosumer could fulfill the energy request and request was failed. In case of grouping, all prosumers participated in the energy sale and energy requests were also completed. To set up the architecture for virtual grouping, the identified problems are:

- **Group creation**: How should multiple diverse prosumers be managed under different groups?

- **Performance improvement**: How to improve the overall performance of the architecture?

For grouping creation, the idea is to *group the prosumers such that the maximum profit for all the groups is equal (or has a minimum difference), for example, the prosumers with profit value* $[3, 5, 2, 10, 13, 3]$ *can be grouped in two with* $PG_1 = [10, 5, 3]$ *and* $PG_2 = [13, 3, 2]$*, in this way the difference of their respective profits is minimal. The algorithm works towards finding the solution to minimize the profit difference between the groups.*

To achieve performance improvement, SynergyChain suggests the use of blockchain as it provides better scalability (than centralized prosumer management) and trust between the participants without the need for centralized authority. Assuming the number of prosumers will be increasing over time, the use of blockchain in energy trading is primitive.

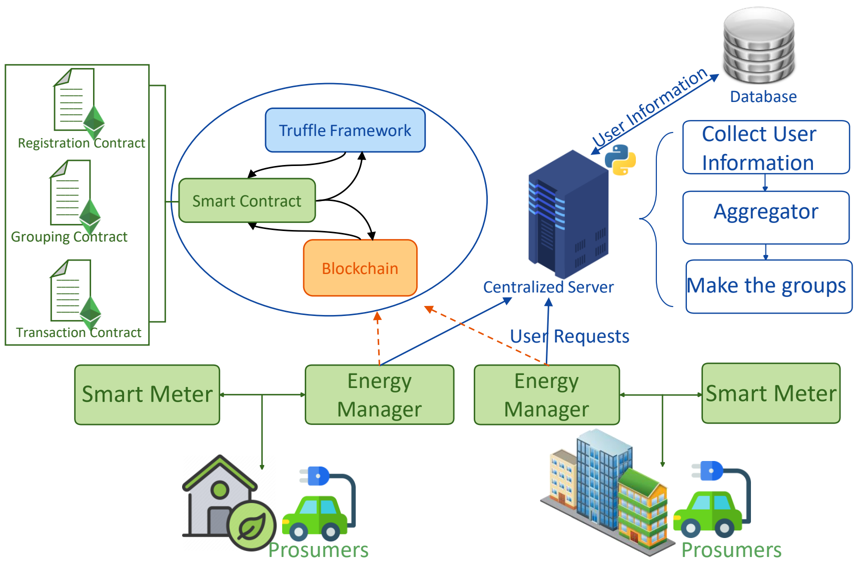

Figure 4.3: Blockchain-enabled vs centralized grouping

4.2.2 Role of Blockchain

Blockchain is a decentralized network of peers, each having a specific address and resources. These peers are responsible for keeping the consistent state of the network. All events on the blockchain are verified by all the participants, which eliminates the fraudulent activities and ensures secure transactions. **Smart contracts** are protocols that run on top of the blockchain to facilitate, verify, and implement the contexts of a normal contract with minimal human intervention [Szabo, 1997]. The energy trading prosumer and prosumer management is usually done in a centraized manner and require third party mediation to fulfil the contract between the two paties. Smart contract has potential as a substitute to provide more decentraled system. SynergyChain is hence developed on top of three smart contracts as shown in Figure 4.3, namely *Registration Contract*, *CreateGroups Contract* and *Transaction Contract*. The *Registration Contract* is used to keep the information of the registered prosumers and also the list of prosumers that willing to join the system. *CreateGroups Contract* contains the function that creates the groups of already registered prosumers. This contract communicated with *Registration Contract* to fetch the prosumer information. The core purpose of making the separate

Table 4.2: Notation used in the article

Symbol	**Explanation**
c_i, p_i	Consumption and Production of prosumer i
pl_i, e	Estimate penalty of prosumer i, Energy Price (EUR)
n, k	Number of prosumers in the system, group
$pind\_i$	Maximum profit of the individual prosumer
lf	The penalty factor due to the locality of the prosumer
tf	The reliability factor of type of the energy source and the
tc, PG_j	Transaction cost, Maximum profit of group j (EUR)
pa	Actual profit of prosumer during time step (EUR)
tu_i	Total energy traded by prosumer i in a time step
APC, APB	Actual average profit of the system (centralized/blockchain)
p, N	Array of profits in Algorithm 1, Number of groups
LEP, HEP	Prosumers that contribute low/high amount of energy

contracts is to keep them independent processes. Any prosumer that is part of the network can trigger the function and the groups are formed over the smart contract. The *Transaction Contract* performs the transaction between the customer and the individual prosumer or the prosumer group. It gets the information about the total group energy from the *CreateGroups Contract* and performs the transaction on the prosumer's/group's behalf.

These tasks are completely autonomous and do not require any third-party oversight. For registration, the prosumer first sends the request to join the network. This request is recorded on a smart contract after which the authentication of the prosumer is done and the unique address of the prosumer is added to the list of the prosumers that are willing to join the system. Then any trusted prosumer, that is already part of the system, can register the user. This type of process is implemented for the trust and fairness of the system.

4.2.3 Heuristic Approach

A heuristics-based single-goal (mentioned in Section 4.2.1) oriented approach for prosumer grouping is utilized to study the effect of blockchain in group management. For simplicity, SynergyChain adapts a greedy algorithm and modified it accordingly (Algorithm 1). Table 4.2 shows used symbols in the chapter. The information gathered from prosumers embedded in the user join request is as follows.

1. **Prosumer contribution**: An important factor for the calculation of the profit. It is the energy committed by the prosumer in the future energy transaction.

2. **Locality**: A profit factor lf is estimated and calculated based on the locality of the prosumer.

3. **Energy Type**: Renewable energy sources have reliability issues and depend on environmental factors. A reliability factor tf is calculated based on this information.

Based on the above information, the maximum profit generated by selling the available energy is calculated. Profit of an individual prosumer i calculated using Equation 4.1. The equation uses energy $((p_i - c_i))$, energy cost e and penalty value. The maximum achievable profit for a group j is then computed using Equation 4.2 using the individual profit values of the prosumers in the group where k is the group size. These profit values are used in Algorithm 1 for group formation.

$$pind\_i = (p_i - c_i) * e - pl_i * e \tag{4.1}$$

$$PG_j = \sum_{i=1}^{k} [pind_i] \tag{4.2}$$

Algorithm 1 and blockchain does extra checking of whether the groups for a time step are already made. This is because any prosumer can send grouping requests that should be declined if the grouping was done earlier. Additionally, the algorithm takes grouping decisions from the learning module before making the groups.

Algorithm 1 Greedy Grouping Algorithm

input: $p \leftarrow [pind_1, ..., pind_n],\ k$
output: $g,\ PG$
$pind_{min} \leftarrow \min(pind_n),\ \ \forall pind_n \in p$
$p \leftarrow \{pind_n \leftarrow pind_n + pind_{min},\ \ \forall pind_n \in p\}$
$g \leftarrow [g_1, ..., g_k]\ \ initialize\ \ each\ \ g_k \leftarrow []$
$PG \leftarrow [PG_1, ..., PG_k]\ \ initialize\ \ each\ \ PG_k \leftarrow 0$
for $(i \leftarrow 1$ **to** $i \leftarrow N)$ **do**
    $p_{max} \leftarrow \max(pind_n), \forall pind_n \in p$
    if $(p_{max} \geq 0)$ **then**
        $j \leftarrow$ **indexOf** $(\min(PG_n), \forall PG_n \in PG)$
    else
        $j \leftarrow$ **indexOf** $(\max(PG_n), \forall PG_n \in PG)$
    end if
    $g_j \leftarrow g_j.insert(indexOf(pind_{max}))$
    $PG_j \leftarrow PG_j + pind_{max}$
end for

4.3 Reinforcement Learning Module (LM)

Self-adaption is a process to learn the patterns through interaction without an explicit need for the supervision or the labeled data. In this approach, an agent independently does iterations to optimize its behavior in the environment where there are uncertainties [Sutton and Barto, 2011]. Recently, reinforcement learning has been used for improvement and forecasting [Wang et al., 2019b]. This kind of learning has several benefits but it comes at the cost of the training time and requires a lot of computational power.

4.3.1 Problem Statement

As mentioned earlier, the grouping algorithm has several advantages when used in the energy trading setting. However, creating groups is not always an optimal deci-

Algorithm 2 Random Grouping Algorithm

input: $p \leftarrow [pind_1, ..., pind_n],\ k$

output: g, PG

$g \leftarrow [g_1, ..., g_k]\ \ initialize\ \ each\ \ g_k \leftarrow []$

$PG \leftarrow [PG_1, ..., PG_k]\ \ initialize\ \ each\ \ PG_k \leftarrow 0$

for $(i \leftarrow 1$ **to** $i \leftarrow N)$ **do**

  $j \leftarrow random([1, ., k])$

  $g_j \leftarrow g_j.insert(i)$

  $PG_j \leftarrow PG_j + pind_i$

end for

Algorithm 3 Aggregation algorithm

Input: $amount_u \leftarrow \ \ RequestedEnergy$

Output: $cost_{total}$

$cost_{total} \leftarrow 0$ // Total cost of the requested energy

$P \leftarrow [p_1, .., p_i, .., p_n]\ \ where\ \ n \leq N$

while $(len(P) \neq 0$ && $amount_u > 0)$ **do**

  $e_{max} \leftarrow \max(e_n), \forall e_n \in P$

  $P.remove(e_{max})$

  $amount_u \leftarrow amount_u - e_{max}$

  $cost_{total} \leftarrow e_{max} * e$

end while

sion. Figure 5.2 shows the increase in the profit of the prosumers that produce less amount of energy (or low energy producers). The figure shows that in group trading introduces prominent improvement in the profit that results in better satisfaction of the participants, than individual trading. However, this increase comes at the cost of decreased profit of the prosumers that generate a high amount of energy (i.e. high energy producers). This phenomenon is depicted in Figure 5.4, which shows both the increase in profit of the low energy producers (LEP) and a decrease in the profit of the high energy producers (HEP). It was observed that the difference between

Table 4.3: The Performance Metrics

Metric	**Meaning**	**In terms of**	**Reward**
Throughput	Average number of transactions	Transactions/sec	R_{th}
Latency	Average round trip time	milliseconds	R_l
Profit difference	Difference between the decrease in the profit of the HEP and increase in the profit of the LEP	%	R_{pr}
Average Profit	Average profit of the systems	%	R_{avg}

the two values should be a high value so that the benefits of the system can be seen. Moreover, it can be seen that the difference is large when a high amount of energy is traded in the system. On the contrary, this difference becomes very small when the amount of energy sold is low. Creating groups in such a situation would negatively impact the encouragement of the prosumers to participate in the collaborative environment. Moreover, there is added overhead of creating the groups over the blockchain that can negatively impact the overall performance of the system. An interesting trend can be seen in Figure 5.7, where the transaction time increases when a small amount of energy is sold. To keep the prosumers in the system, a better adaptive technique for grouping is required that can decide on dividing the prosumers in the group during the runtime, based on the previous experiences. For example, an agent can interact with the environment and can decide whether to create the grouping or do the individual transactions.

Another aspect that encourages the creation of such a learning module is uncertainty in the energy produced by renewable energy sources. Both the quality and quantity of these energy sources are immensely affected by the weather conditions and can affect the profitability of the system. For example, if one solar energy producer commits to producing a specified amount of energy to the group and is not able to produce that amount of energy, there would be a penalty incurred by the group. Moreover, there might be an energy shortage and the buying request can not be fulfilled. In both of these cases, the individual energy trade performs better than the grouping technique (as shown in Figure 5.1).

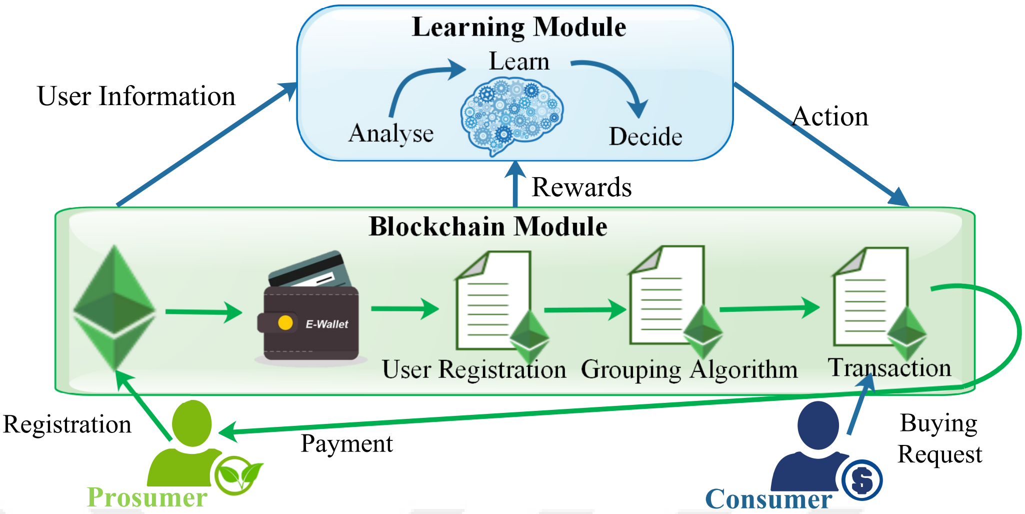

Figure 4.4: Architecture overview of SynergyChain

4.3.2 Mapping RL to Grouping Problem

The reinforcement learning agent learns from the interaction with the environment. During each iteration, the agent observes the environment state and based on that it takes an action. The reward is given to the learning agent based on the new state of the system. From this reward value, the agent learns the action to improve the decision making and achieving maximum cumulative awards. The choice of creating the group or not for the next time slot can be solved as an optimization problem or machine learning. Optimization requires running all the scenarios that are time taking and will not take advantage of prior knowledge. While machine learning requires prior knowledge but mostly requires a large amount of labeled data. Although the fact that reinforcement learning requires a longer time for convergence than a supervised approach, reinforcement learning provides a better solution for self-adaption as it does not require labeled historical data for training and does not require supervision. For the grouping scenario, there is not much-labeled data available. The agent-based learning would satisfy the need of the problem as it can learn even after it is deployed in the real world.

4.3.3 Model Representation

The first and most crucial task in learning is the creation of the model that represents the underlying system. Generally, system representations follow the model of *Markov Decision Process*, that has action space A and corresponding state-space S, policy that defines the state transition function T_a*(s, s')* and reward function R_a*(s, s')*. The reward function defines the goal and plays an important role in the convergence of the model. Several algorithms for reinforcement learning such as Deep Q-Learning, Deep Deterministic Policy Gradient, Q-learning, State–action–reward–state–action can be realized as the solution to the defined problem. These algorithms are directly impacted by the number of possible actions and respective states.

Figure 4.4 shows the way reinforcement learning is modeled for adaptive grouping for energy trading. The agent interacts with the smart contract and depending upon the reward, it gets the information about the next step. The state here is the performance of the system and profit earned by the prosumers. The action is the decision of making the groups or to have individual transactions of the energy.

The agent gains information about the prosumers and the predicted energy usage value from the smart contract. Every action changes the state based on which the rewards are calculated according to the performance metrics shown in Table 4.3. Every time step, the agent takes a decision, after that the smart contract uses that to calculate the performance. Then it inverses the decision and does the calculations, based on that the reward is calculated using Equation 4.3 - 4.7 where each reward value for performance matrix is accumulated.

$$R_t = R_{th} + R_l + R_{pr} + R_{avp} \tag{4.3}$$

$$R_{th} = \begin{cases} 1 & \text{if } R_{th}^{g} \geq R_{th}^{ng} \\ R_{th}^{g} - R_{th}^{ng} & \text{if } R_{th}^{g} \leq R_{th}^{ng} \end{cases} \tag{4.4}$$

$$R_{l} = \begin{cases} 1 & \text{if } R_{l}^{g} \geq R_{l}^{ng} \\ R_{l}^{g} - R_{l}^{ng} & \text{if } R_{l}^{g} \leq R_{l}^{ng} \end{cases} \tag{4.5}$$

$$R_{pr} = \begin{cases} 1 & \text{if } R^{g}_{pr} \geq R^{ng}_{pr} \\ R^{g}_{pr} - R^{ng}_{pr} & \text{if } R^{g}_{pr} \leq R^{ng}_{pr} \end{cases} \tag{4.6}$$

$$R_{avp} = \begin{cases} 1 & \text{if } R^{g}_{avp} \geq R^{ng}_{avp} \\ R^{g}_{avp} - R^{ng}_{avp} & \text{if } R^{g}_{avp} \leq R^{ng}_{avp} \end{cases} \tag{4.7}$$

where R_t is the total reward earned by the agent during the time step t. It is some of the rewards earned individually by the system from the performance matrix. Reinforcement learning works towards the goal of maximization of the expected value of the cumulative reward:

$$E = (\sum_{i=0}^{\infty} \gamma^i R_i) \tag{4.8}$$

where R_i is a reward given during epoch i and γ^i is a discount rate. The discount rate is a trade-off in reinforcement learning that is done between exploration and exploitation. Exploration focuses on discovering new solutions at the expense of taking risks of less reward. Once the model is defined, SynergyChain uses SARSA learning algorithm. SARSA is a model-free reinforcement learning algorithm [Zhao and et. al., 2016]. This algorithm is similar to the Q-learning algorithm [Hester and et. al., 2018] where it maximizes the Q-value by interacting with the environment but calculates the reward using future action and state.

Algorithm 4 SARSA algorithm

$S_t \leftarrow [S_{start}, ..., S_{terminal}]$
$A_t \leftarrow [A_0, ..., A_n]$
$R_f \leftarrow R_t$
$\alpha \leftarrow [0, 1]$, learning rate typically 0.1
$\gamma \leftarrow Discount\ \ factor$
$S_{current} \leftarrow S_{start} \in S_t$
while $(Q \geq Q_{converge}\ )$ **do**
    while $(S_{current} \neq S_{terminal})$ **do**
        Choose $A' \in A_t$
        Observe $S' \in S_t$, after A' action
        $Q \leftarrow R_{A'} + \gamma \times Q_{S'}$
        $Q_{S_{curr}} \leftarrow \alpha \times (Q - Q_{S_{curr}})$
        $S_{current} = S'$
        Update $R_{S_{current}}$
    end while
end while

Chapter 5

SYNERGYCHAIN: IMPLEMENTATION AND RESULTS

This chapter discuss the implementation and results of the presented SynergyChain [Ali et al., 2020c]. First the implementation of each module including the smart contract is explained. Then the experimental setup is discussed. Lastly, rigorous testing scenarios are presented, on which the experiments are run.

5.1 Implementation

SynergyChain works with two types of entities in the system, prosumers and consumers, that have a smart meter and energy management system for sending the energy generation or energy requirement information to the network. Also, every prosumer/consumer should have Ethereum Blockchain accounts with specific addresses and an array of specific characteristics including energy contribution, location, and energy type. The contracts can be deployed by any entity with the contract code and the address of deployed contract is provide to prosumers/consumers. This model assumes rational entities and gives minimum power of actions to the owner of the smart contract, hence improving the system security.

5.1.1 Smart Contract Implementation

Two types of architecture implementations, Centralized VPGs and Blockchain Prosumer Group, are done for the testing purposes. The Centralized Prosumer Group Module is simulated using *Python 3.6* and deployed over the local network. A server/client architecture is created where the prosumers first join the system and then are organized in groups. This implementation is explained in Figure 4.3. To implement the proposed algorithm on the Ethereum Blockchain, three tools were used. First, the smart contracts were written on *Remix IDE*[1] for testing the correctness of the contracts. The contracts are written in *solidity*[2] language. Once a contract is written we can deploy our contract into any of the Ethereum networks: the main network, Test networks (Ropsten, Kovan, Rinkeby), local-host, and custom RPC. In our case, the contracts are deployed on Ropsten and Rinkeby to create the decentralized system application and test the platform. Infura provides the virtual wallet that was used for the transactions of the Ethereum smart contracts. Once these contracts are deployed, the *Python web3*[3] library is used to interact with the Ethereum Contracts.

5.1.2 Independent Transactions

The learning module of the SynergyChain provides the decision of grouping the prosumers. In case the decision is to not perform grouping, the transactions are done independently. The difference between grouped and independent transactions is that each prosumer is considered a group of 1 (whereas there are N groups of p/N prosumers in group transaction). This was created to improve the overall profitability and performance of the system. When SynergyChain performs independent transactions, each prosumer creates a smart contract *TransactionContract*, where it gives the energy available to be sold, the location, and the price of the energy.

After the creation of the smart contracts, the consumers can send the buying requests to the system. A matching technique, shown in algorithm 5, is utilized to match these buying requests to the prosumers. The algorithm takes the user buying demand as an input and then returns the prosumers list that can fulfill the user demand.

[1] Remix - Ethereum IDE. [online] Available at: https://remix.ethereum.org [Accessed 20 June 2020].

[2] Ethereum IDE. [online] Available at: https://remix.ethereum.org [Accessed 20 June 2020].

[3] Web3py.readthedocs.io. 2020. Introduction — Web3.Py 5.11.1 Documentation. [online] Available at: https://web3py.readthedocs.io/ [Accessed 20 June 2020].

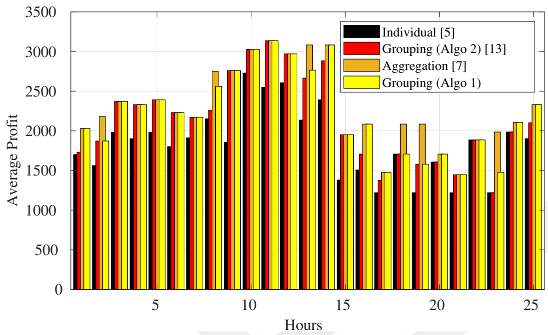

Figure 5.1: Average profit comparison with [Mengelkamp et al., 2018a, Sabounchi and Wei, 2017, Da Silva et al., 2013].

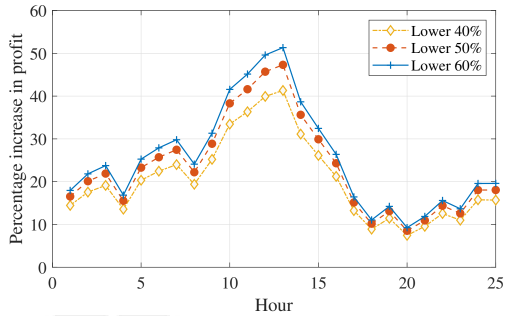

Figure 5.2: Profit increase with LEP.

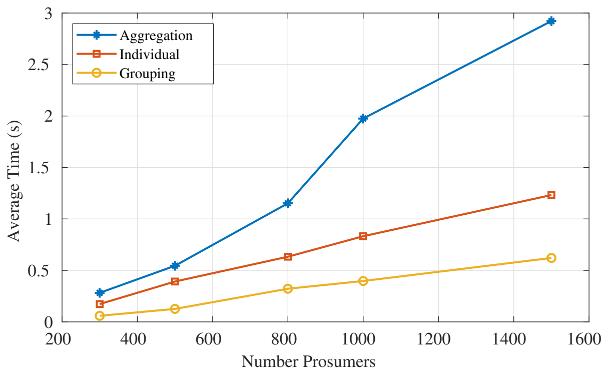

Figure 5.3: Execution time comparison with [Mengelkamp et al., 2018a, Sabounchi and Wei, 2017] for 1000 energy requests.

Algorithm 5 Prosumer/Consumers Matching

Function $FindMatching$
Input: $D_u \leftarrow userDemand$
Output: $offers, price_{total}$
$price_{total} \leftarrow 0$
$offers \leftarrow []$
$E_{avail} \leftarrow [p_1, .., p_i, .., p_n] \;\; where \;\; n \leq N$
$E_{avail} \leftarrow sort(E_{avail})$
for $(i \leftarrow 1$ **to** $i \leftarrow n)$ **do**
  if $(p_i \geq D_u)$ **then**
    $offers \leftarrow offers.insert(p_i)$
    $price_{total} \leftarrow p_i * e$
    break
  else
    $offers \leftarrow offers.insert(p_i)$
    $D_u \leftarrow D_u - p_i$
    $price_{total} \leftarrow p_i * e$
  end if
end for
End Function

5.1.3 Learning Module

SynergyChain has a learning module that automates the grouping decision. A reinforcement learning-based algorithm is used for the adaptive grouping. We planned to implement a learning module over Solidity, but the language has not reached that development phase where complex algorithms such as reinforcement learning can be executed, so a python-based implementation was done. The predictor module works independently from the rest of the system. The model is deployed over the cloud and each prosumer asks for the next iteration result from it. Once the decision is returned from the learning agent, that result is used in the grouping algorithm. In Section 5.2, with a use case study, we demonstrate the usage of the framework for the transformation of a grouping application.

5.2 Experimental Setup and Results

We used the Hourly Energy Consumption data-set [4] containing data from renewable sources from Spain and took the hourly energy values for our experiments. The energy consumption was then randomly distributed among 300 participants. This is because previous works (such as [Sabounchi and Wei, 2017] show around 300 participants in their results. Every time step (time step in one hour), the prosumers send the new value for the energy contribution for the hour ahead. The prosumers are grouped according to the algorithm 1. We calculate the individual and the group profit from Equations 4.1 and 4.2. Then random buying requests are generated (the overall sum is less than total energy consumption per hour). Additionally, we create a total of *5* groups for the experiments. There is added cost for the blockchain transactions that have to be paid. So the final sales of the system are calculated by deducting the transaction cost from the overall profit. In our study, the overall transaction cost is 0.26 EUR according to the gas prices.

[4] Jhana, Nicholas. “Hourly Energy Demand Generation and Weather.” Kaggle, 10 Oct. 2019, www.kaggle.com/nicholasjhana/energy-consumption-generation-prices-and-weather. Last Access: 31 March 2020

5.2.1 Profitability

These experiments were done to prove the efficacy of the grouping and to study its effect on the overall profit of the system. As explained earlier, we have used test data for performance evaluation. Moreover, three types of energy resources (wind, solar, and oil) and 7 localities for which reliability factor (for resource type) and penalty factor (for locality) values are set. Every prosumer is randomly allocated the location and the resources as this information is missing in the data-set (It was not possible to find a 100% free and available dataset for our testing purposes). The value for penalty pl_i is then calculated (Equation 5.1) as the average of two penalty factors which is then used in Equation 4.2 to calculate the maximum final profit value for the grouping.:

$$p_i = (lf + tf)/2 \tag{5.1}$$

The average profit values are calculated from the number of units sold in the time step (tu_i). The following Equations are used for the results [Equations 5.2-5.3].

$$pa_i = tu_i * e \tag{5.2}$$

$$APC = (\sum_{i=1}^{n} pa_i)/n \tag{5.3}$$

$$APB = (\sum_{i=1}^{n} [pa_i - tc])/n \tag{5.4}$$

Figure 5.1 shows the average profit of the system and the effect of grouping on the prosumer overall profit. The grouping approach used in figure 5.1 outperforms individual energy trading process by average 20.7%. This is predominantly because of the occurrences of failed energy buying requests in the case where participants sell the energy individually due to the inability to consummate the required energy amount. Moreover, as depicted in Figure 5.2, grouping approach introduces an average 22.4% increase in the profit of low energy contributors by providing equal share from the sale to the prosumers with lower energy contribution. This comes at

Table 5.1: Group request time, throughput, request time, failed requests and grouping time.

	Number of prosumers	300	500	800	1000	1500
Grouping Requests (s)	**VPGs**	5.72	11.47	21.89	38.96	67.32
	VPGs Blockchain Ropsten	1.53	6.43	13.64	26.41	40.52
	VPGs Blockchain Renkeby	1.83	6.73	13.94	26.71	40.82
	SynergyChain	0.83	5.13	12.34	25.11	39.22
Throughput (trans/s)	**VPGs**	0.46	0.19	0.12	0.08	0.04
	VPGs Blockchain without LM	0.30	0.23	0.18	0.13	0.10
	SynergyChain	0.43	0.33	0.18	0.13	0.10
Average Request Time	**VPGs**	2.23	8.26	18.69	33.56	61.12
	VPGs Blockchain Ropsten	5.68	9.23	15.64	27.89	52.21
	VPGs Blockchain Renkeby	5.68	12.23	18.69	33.56	61.12
	SynergyChain	2.23	8.26	15.64	27.89	52.21
Failed Requests	**Centralized Failed Requests**	5	27	56	143	273
	Blockchain Failed Requests	1	12	23	79	172
Grouping Time (s)	**Centralized**	0.004	0.006	0.009	0.011	0.023
	Blockchain	1.577	1.579	1.582	1.584	1.595

expense of decrease in energy profit of HEP, which is comparatively low as show in Figure 5.4. As the reason, the system encourages more prosumers to participate in the energy trade.

5.2.2 Performance

In this section, performance results for both centralized and blockchain-based architecture are compared. In this experiment, we exploited our system with a range of prosumer requests. Our system comprises of two communications. First is the request sent to the server to join the network. The second is the request to get the group information from the server. The average time to complete each request is shown in Figures 5.9 and 5.8. For calculations, we use Equation 5.5.

$$res = rec - req \tag{5.5}$$

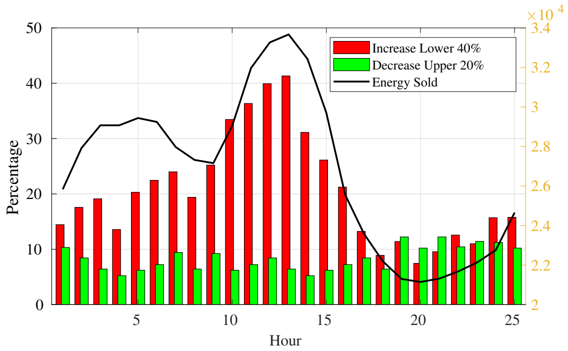

Figure 5.4: Profit increase of LEP and decrease of HEP after grouping.

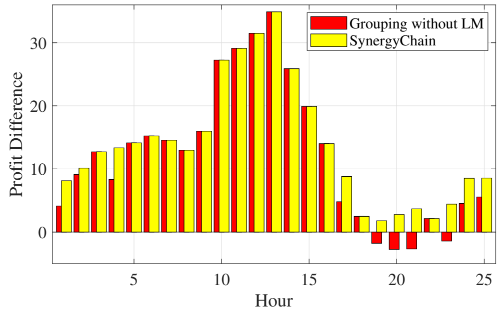

Figure 5.5: Difference of profit decrease

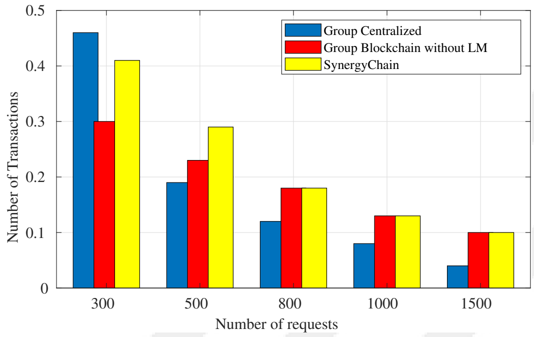

Figure 5.6: Transactions happening per second.

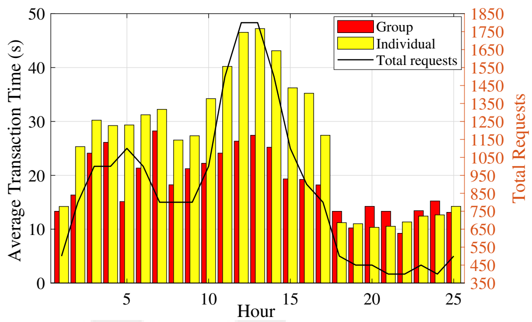

Figure 5.7: Transaction time

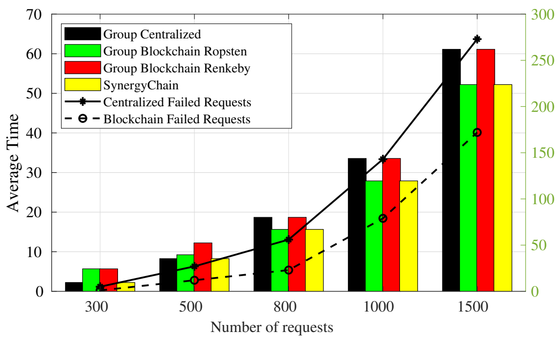

Figure 5.8: Average response times

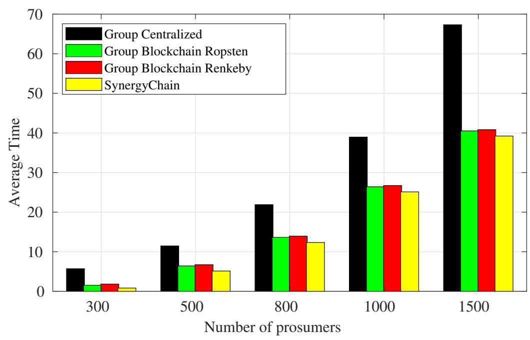

Figure 5.9: Average time for group info.

where *res*, is the response time, *req* is the time when the request is sent to the server and *rec* is the time when the first byte is received from the server.

In Figure 5.9, the response time of registration communication can be seen. The two-way communication is done in this process. First, the prosumer sends the join request and then the server responds with an ”accept” message. We assume that the registration request of the prosumer is accepted without any delay. It can be seen that for fewer requests, the centralized server performs better than smart contract-based infrastructure. That is because the Ethereum Smart contract has the time overhead for the execution of the transaction. Here the transaction is done when the prosumer appends itself to the list of participants that want to join the server. But with the increase in the number of requests, the average response time for blockchain is better than the centralized architecture which shows better scalability of the blockchain-based solution. Moreover, the number of failed requests is more for a centralized setup. This is because of the queueing delay and single point of failures. The graph has also compared the response time for Ropsten (PoW) and Renkbey (PoA), the minor added delay was experienced in Renkbey.

The comparison for response time for getting group information between the two setups is shown in Figure 5.8. The time to run algorithm 1 is excluded in the graph, as it takes negligible constant time as given in Table 5.1. The prosumers send the *get_info* request to the server and the server responds with the desired data. The response time is calculated as stated earlier, and it is observed that the blockchain-based solution performs better than the centralized system. From the two studies, we can deduce that the blockchain is more scalable, faster, and performs better in the prosumer grouping arrangement than the centralized based network.

5.2.3 Effect of Reinforcement Learning Module

The results in this section manifest the effect of the learning module on the profitability, user satisfaction, and the performance of the overall system. First, the model was trained and due to the low complexity of the problem, the model converges very fast. This trained model is then used for the future decisions of grouping. This phenomenon is also explained in detail in section 4.3.1. The tests were run with the learning module along with the group creation to better realize the effectiveness of this module. Figure 5.6 shows the increase in transactions happening per second where the transaction is a buying request executed in the system. Performance improvements are shown in Figure 5.8 and 5.9 where the grouping timing and the average time for registration are slightly better than that of blockchain because of the removal of grouping overhead improved the overall average time.

The improvements in the profitability and the overall user satisfaction can be seen in Figure 5.5 that shows the difference between the profit increase in lower prosumers and the profit decreased in the high energy prosumers. As argued earlier, this difference should be more to encourage the prosumers with high energy to participate in the grouping, otherwise, those prosumers would prefer to sell the energy independently. We can see that there are some instances when this difference is negative when LM is not used. But a significant improvement is achieved with the use of the PM module. Moreover, the *18.3%* overall increase in the profit of the system was also reported from the results.

5.2.4 Comparison

A general comparison with centralized solutions is provided in Figures 5.8 and 5.9 depicting better scalability in case of blockchain. In this section, a conclusive comparison of presented grouping technique (algorithm 1) against the solutions of Individual trading [Mengelkamp et al., 2018a], Energy aggregation [Sabounchi and Wei, 2017] (Algorithm 3), and Random grouping [Da Silva et al., 2013] (Algorithm 2) is provided.

Figure 5.1 presents the comparison of the average profit when running the above listed algorithms against SynergyChain in the same setup as presented in section 5.2.1. The results indicate that SynergyChain gives comparable profitability to the prosumers while providing preference to LEPs. The other mechanisms lack in providing such a solution. SynergyChain performs better than random grouping (algorithm 2) while aggregation based strategy, in particular situations, outperforms SynergyChain because it makes use of all the energy available rather than just looking in a singular group. However, the aggregation based strategy has run time overhead as shown in Figure 5.3. This is because this process looks for a prosumer with maximum energy which takes $\mathcal{O}(n)$ while this process is repeated until the energy demand is met. Hence, in the worst case, the time complexity becomes $\mathcal{O}(n^2)$. Whereas, the time complexity for completing energy requests in the group would be $\mathcal{O}(k)$ where k and $k < n$ is the number of groups. The results empirically show that the proposed grouping method improves profitability while providing a better solution in terms of the run time of the algorithm.

Table 5.2: Gas cost and price in Euro for different transactions.

Transaction Type	No of Groups	Number Prosumers	Gas Cost	Price in Euro
Join Request	5	-	107762	0.03394503
Grouping	5	300	207061	0.065224215
Grouping Decision	5	-	42655	0.013436325
Buy Request	5	300	74626	0.02350719
Grouping Information	5	300	53498	0.01685187

5.3 *Gas Cost Evaluation*

As the system is built upon smart contracts, that incurs cost for each transaction execution depending upon the task and number of changes done over the blockchain. There are 5 types of transactions: join request, creating groups, setting grouping decision, buying request ant getting grouping information. Table 5.2 summarizes the gas cost associated with each of the transactions. The gas cost for joining, setting group decision, making groups and getting information is paid by prosumers while buy request is paid by the consumer. The system automatically deducts this cost from the wallet of the participant that sends the request. As can be seen, the total price in Euros turns out to be small and hence is not included in profit results discussed in section 5.2.1.

5.4 *Discussion and Future Work*

These work exploit the opportunities that blockchain provides in the prosumer group construction within the P2P energy trading setting. Blockchain has been utilized to design a self-adaptive smart contract-based framework, namely SynergyChain, that ensures real-time collaborative trade of energy without the need for a central authority. SynergyChain shows that the aggregation of multiple energy prosumers through grouping outperforms the case in which individual prosumers participate in the energy market. Besides, the blockchain-based system is faster and provides improved scalability in comparison to the centralized approach. Moreover, a reinforcement learning module that decides on whether the system should act in groups or independently has been implemented. The learning module shows that the system performs better both in terms of profitability and performance than without the learning module. Better prosumer management would encourage more participants in the system. More prosumers would buy renewable energy sources from which both the industry and the world can get benefit. In the future, our predictor would also define the parameters of the grouping, for example, to take into account the effect of the group size on the system.

Chapter 6

FEDERATEDGRIDS: ARCHITECTURE AND ALGORITHMS

Motivated by the existing work and use of blockchain in energy markets, in this work, an autonomous, blockchain assisted hybrid P2P (peer to peer) and M2M (microgrid to microgrid) energy trading and sharing system is developed [Ali et al., 2020b] to improve the social utility and overall benefits of prosumers and consumers in the microgrids. This chapter discusses the problem definition and the algorithms that are developed over smart contracts.

6.1 Problem Statement

As the energy demand and energy generation are growing rapidly, new management systems need to be developed that provide monetary incentives to encourage users to take part in such energy trading management systems. Blockchain-based demand response programs are presented for smart management [Pop et al., 2018], [Mengelkamp et al., 2018b] that is done through load curtailment which is not efficient for participants comfort. Moreover, these methods are limited to local markets and do not include other microgrids and utility grids into energy trading. Another problem with such systems is they lack in a situation where the buying power of consumers is limited.

To overcome these limitations, a model that performs P2P and M2M energy trading/sharing is a promising solution. In such a model electricity prosumers can use RES to generate energy locally. The surplus energy, that remains after fulfillment of there energy demands, can then be shared or traded with energy deficit users or consumers within the microgrids and across the microgrids. These trading markets also require a pricing mechanism such that the price of buying from fellow prosumers is less than that of the utility grid. An open book trading mechanism (bidding) can be used for such markets [Inayat and Hwang, 2018] but in such systems, some consumers pay more than others that create an imbalance in trading. In [Park et al., 2018], a fixed price based mechanism is proposed that solves such issues but such a mechanism is not beneficial for prosumers as energy price is always 70% less than the electricity price of utility. A need for better formulation of energy prices based on availability and electric losses is there to maintain the sustainability of P2P and M2M trading. Other than that, the trading system should be autonomous and should be able to identify Will there be trading or sharing? Who to sell/share energy with? How can energy be traded? and How can resources be shared?

6.1.1 Proposed Solution

This work aims to reduce electricity consumption cost and consumer dependence over utility grids. A stable blockchain-based hybrid trading mechanism is proposed where consumers can both use P2P and M2M energy transactions to fulfill the energy requirements. This drastically reduces the load over utility and also helps utility grids to smartly use and distribute energy to the areas that are not self-sufficient hence improving the overall efficiency of energy systems of the country. Unlike [Khalid et al., 2020], the coalition between microgrids is can also be done to maximize the utilization of available renewable energy within the system. This work also formulates pricing mechanisms that take into account the load and generation of locality and energy losses due to transmission to provide low prices to consumers that also benefit prosumers in the system. Other than that, a sharing mechanism is developed that allows the consumers to fulfill their power demand in situations where they have limited buying power. The proposed solutions:

1. Eliminates the main grid monopoly buy decentralizing the energy trading markets and creates an environment where local consumers can get energy at lower prices.

2. Self-enforcing smart contracts are designed for devising transactions between entities hence reducing the demand from the main grid.

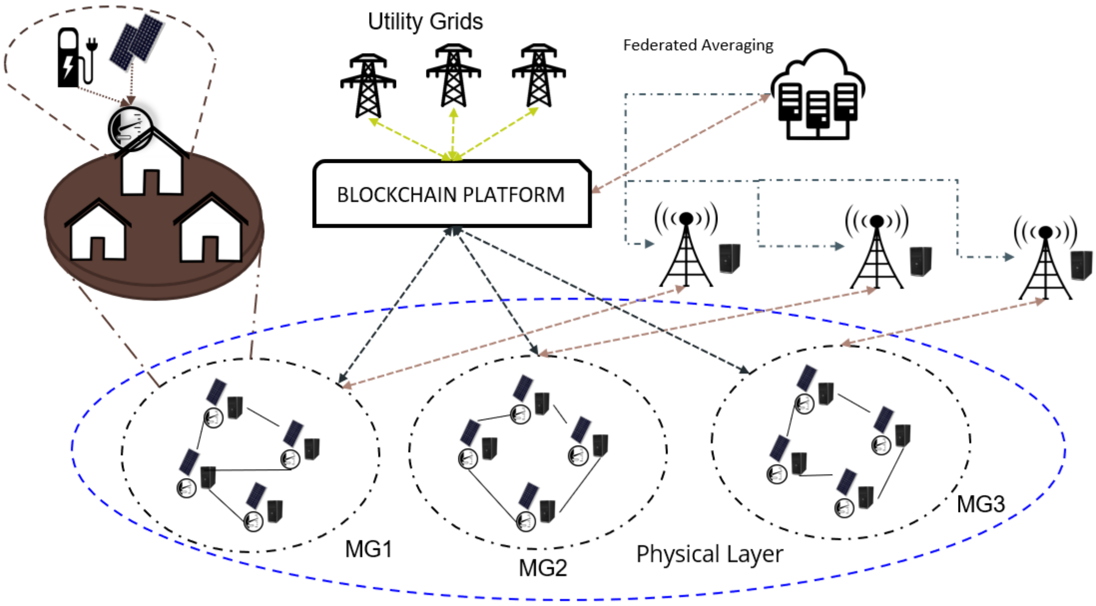

Figure 6.1: System architecture and interaction.

3. Helps utility grid by decreasing its load that can then be used in low energy areas for better management.

4. A matching mechanism is developed that takes into account energy requirements and match buyers with the sellers.

6.2 System Architecture

6.2.1 Model Components

The main components of our model are:

1. Prosumers/Consumers: That sell/buy the energy.

2. Microgrids: That fulfills the local energy demands.

3. Utility grids: The main grids that facilitate energy transfer.

4. Blockhain: For decentralization.

The overview of the system is presented in figure 6.1. In the proposed model, a cost-effective trade of energy and planning for distributed energy is achieved by leveraging the community prosumers and the utility grids. ***The utility grids*** are usually owned by the governments and are composed of:

1. Large Energy Storage: These are storage places containing large batteries that can store the surplus energy/

2. Conventional Energy Generators: These contain the energy generated by the gas, oil, and coal.

3. Renewable Energy Generators: These generate energy through renewable energy. The energy is taken from solar panels, windmills, and water power.

The utility grids are controlled using the centralized authority that has computational power and contains cloud storage. The energy is controlled using the energy manager that is responsible for handling the energy trading requests.

The prosumers are the consumers that can generate energy. The prosumers generate energy through the renewable energy sources as mentioned above but on a smaller scale than the utility grids. The prosumers are equipped by the smart energy management system (EMS) that is responsible for the smart utilization of the energy. Many prosumers have the capability of generating the surplus energy that is stored in the battery storage and might be traded or shared with the other consumers in the community. This is also done through the EMS technology which takes in the buying request and completes it by selling the energy available. Both the prosumers and consumers can be houses, office buildings, and electric cars. The collection of these prosumers gives out the concept of the ***Microgrids***.

6.2.2 Microgrid Cluster Definition

Microgrids are electricity distribution systems containing a collection of connected consumers, prosumers, and DERs. A Microgrid normally consists of a collection of individual premises and DERs in a local geographical area that shares the same medium-voltage/low-voltage (MV/LV) transformer [Zhang et al., 2018] and are connected with the electrical wires. The capabilities of the Microgrids contain:

1. The microgrids can operate, control, and coordinate the distributed energy resources.

2. Microgrids are connected with the main power network (utility grids) to buy the energy from the main grids.

3. Microgrids are connected through the centralized control systems that are responsible for the trading among the buyers and the sellers.

All the trading among the microgrids is considered P2P in nature as the buyers and sellers can directly sell to each other. Conventional microgrids are centralized in nature i.e. all the trades are done through the centralized authority. There has been a rigorous amount of work been done to make them more decentralized using the blockchain. Microgrids do not contain centralized storages like utility grids. Moreover, a microgrid does not contain information about other microgrids that might be connected with the utility grids neither do they have any physical connection with the other microgrids [Goranović et al., 2017].

The participants in a micro-grid fulfill the energy demands by trading energy amongst each other. The consumer buys the energy grom the prosumers at a fixed price. If the energy demand is more than the produced energy, the micro-grid communicated with the main power grid network to fulfill the energy demand which might be more expensive than the energy bought from the prosumer in the micro-grid.

6.2.3 Use of the Blockchain

The use of blockchain is to:

1. Handle the trade between the prosumers in the microgrid.

2. Handle the trade between the microgrids and the utility grids.

3. Store the information of the prosumers/consumer/utility grids and the microgrids.

The blockchain smart contracts will be used to create contracts that are used to facilitate the above-mentioned trades.

6.2.4 Interaction Between System Components and Participants

In this section, the review of different modules, and interaction between participants and system components are explained.

1. A prosumer sends the registration request with location, available energy, and microgrid information. This information is stored over the blockchain.

2. A consumer sends the registration request along with energy demand. This information is also kept over the blockchain.

3. Before starting the energy trading/sharing, a federated learning module (explained in section) is trained.

4. Using the federated learning model, the load prediction is done and set in the system.

5. System sets the energy price (explained is section 6.3.1) for M2M trading. Also, depending on the consumer requests in step 1, each microgrid separates out the energy that will be required for the local consumers. The rest of the energy can then be used for energy trade or share across the microgrids.

6. The consumers can then send energy trading (explained in section 6.3) request. The system matches the request with appropriate prosumer in locality or microgrid that has surplus energy.

7. If the energy request has to be fulfilled by the microgrid, another match loop is done to find the prosumers that will sell the energy, hence removing the

requirement of the microgrid manager. Also, the utility grid is paid for the usage of its resource.

8. The main contract transfer the payment from the consumer account to the prosumer account. If the consumer has no e-wallet, a payment request is sent to the consumer.

9. System keeps track of the amount of energy traded. The systems go into energy sharing (explained in section 6.4) phase when the energy traded becomes greater or equal to the predicted energy demand.

10. Now the consumers with resource deficiency can send the energy sharing request which is fulfilled if there is still energy in the system.

11. At the end of the timestep, prosumers update the available energy for the next timestep.

Algorithm 6 explains how energy requests are handled in the system. Moreover, figure 6.2 depicts the interaction between participants and the system.

6.3 Energy Trading

The energy trading process can be done in two stages. First, the prosumers post the advertisement for the energy that it has to sell. The consumers can see the advertisement and choose to buy the energy from any of the prosumers. The main aim of this model is to keep power and price balance within the microgrids and the prosumers. Also to reduce the dependency on the main utility grids that reduces the loss in energy delivery and also the overall price of energy.

As discussed earlier, the prosumers in a microgrid can sell the energy to the consumers, There can be a possibility where the energy demands of the consumers in a microgrid are fulfilled and still there is surplus energy to be sold. In this case, the prosumers in a microgrid can benefit by:

1. Selling the energy to the utility grid.

2. Selling the energy to other microgrids.

In the first case, the prosumers sell the energy directly to the main grids at a price fixed by the utility grids. The cost of transfer of energy through the wires and the cost of energy losses is also incorporated while the trade is done [Khalid et al., 2020]. Another case is that energy can be sold to the other microgrids using the P2P communication between the centralized systems of the two microgrids. This helps to reduce the loss of energy delivery. It also reduces the dependency of the community microgrid on the main grid to balance the discrepancy between supply and demand in the microgrid by sharing energy with other microgrids in the community. This work proposes a hybrid energy trading approach that performs the following types of trades:

1. P2P trade among the prosumers within a microgrid.

2. P2P energy trade between the community microgrids.

3. P2P energy trade between microgrids and the utility grids.

6.3.1 Energy Pricing for Energy Trading

All the energy prosumers within a microgrid comprise of different renewable energy resources. Moreover, there are consumers within the microgrid that have differnet load demands. The load demands and energy generations within a microgrid MG_n can be expressed as:

$$D^t_{MG_n} = [D^t_1, D^t_2, D^t_3, ..., D^t_i] \tag{6.1}$$

$$Gen^t_{MG_n} = [Gen^t_1, Gen^t_2, Gen^t_3, ..., Gen^t_i] \tag{6.2}$$

where D_{MG_n} is the demand while $Gen^t_{MG_n}$ is the generation of microgrid MG_n during timestep t.

Depending on the information from equation 6.1 and 6.2, the price calculation for energy trading for the next time step is done in two stages. First, the price calculation for the microgrid to microgrid trading is done. Let's call this value 'EPM'. Afterward, the price for peer to peer trading within a microgrid is calculated. Let's call this value 'EPP'. These energy price values depend upon the energy price of utility, *'EPU'*. To make the system beneficial, the following property should hold:

$$EPP \leq EPM \leq EPU \tag{6.3}$$

Now, for the calculation of the energy price for M2M trading, first calulate the total energy generation to demand ration:

$$TED_{MG_n} = \sum_{i=1}^{np} D_i^t \tag{6.4}$$

$$TEG_{MG_n} = \sum_{i=1}^{nc} Gen_i^t \tag{6.5}$$

$$GDR_{MG_n} = \frac{TEG_{MG_n}}{TED_{MG_n}} \tag{6.6}$$

where np is the number of prosumers and nc is the number of consumers within a microgrid. Now, using equation 6.6, the price of M2M energy trade for next time step becomes:

$$EPM_{MG_n} = \begin{cases} \frac{EPU}{GDR_{MG_n}} & \text{if } GDR_{MG_n} \geq 1 \\ EPU & \text{if } GDR_{MG_n} < 1 \end{cases} \tag{6.7}$$

EPM_{MG_n} is the price that has to be paid by the consumer that sends the trading request from any other microgrid to the microgrid MG_n and is decided at the start of the timeslot.

The energy price for P2P trading within the microgrid varies with for different energy trading requests because it depends on distrance between the consumer and the prosumer and the type of energy resource (ET) the prosumer is using. This price value is calculated when energy trading request from consumer C_i is received. The system then calculates energy price for energy request to prosumer P_i as follows:

$$GDR_{P_i} = \frac{Gen_{P_i}}{D_{C_i}} \tag{6.8}$$

$$Dist = Distance(P_i, C_i) \tag{6.9}$$

$$EPP_{P_i} = \begin{cases} \frac{EPM_{MG_n} + Dist \times ET_{P_i}}{GDR_{P_i}} & \text{if } GDR_{P_i} \geq 1 \\ EPM_{MG_n} & \text{if } GDR_{P_i} < 1 \end{cases} \tag{6.10}$$

If a consumer C_i buys EB_{UG}, EB_{MG_n}, and EB_{P_i} from utility grid, M2M and P2P trading, then the overall cost of a consumer C_i can then be calculated as:

- For peer to utility grid trading:

$$TC_{C_i}^{UG} = EB_{UG} \times EPU \tag{6.11}$$

- For M2M trading:

$$TC_{C_i}^{MG} = \sum_{i=1}^{n} [EPM_{MG_n} \times EB_{MG_n}] \tag{6.12}$$

- For P2P trading:

$$TC_{C_i}^{PP} = \sum_{i=1}^{i} [EPP_{P_i} \times EB_{P_i}] \tag{6.13}$$

Where TC is the total cost for different trading settings. After the price calculation, the power and financial transactions are settled down in real-time.

Algorithm 6 Energy request handling algorithm

input: *Number grids*, *Utility Price*, *Resource Price*, L

$N_g \leftarrow$ *number Grids*

$P_u \leftarrow$ *price Utility*

$P_r \leftarrow$ *Resource Price*

$E^t_{traded} \leftarrow 0$;

$Sharing \leftarrow False$;

$Load^t_{predicted} \leftarrow L$

Function energyRequest($energyAmount$, $reqType$)

consumer $\leftarrow$ request.address;

$C_{total} \leftarrow 0$;

$C_{share} \leftarrow 0$;

$amount_{left}$, prosumers $\leftarrow$ find Prosumers that match energy requirement

for ($p \in prosumers$) **do**

    $p_p \leftarrow$ calculate price using equation 6.10

    p.sale $\leftarrow$ p.sale + $p_p * et_p$;

    $C_{total} \leftarrow C_{total} + p_p * et_p$;

end for

if ($amount_{left} \neq 0$) **then**

    $amount_{left}$, microgrids $\leftarrow$ find microgrids that match energy requirement

    for ($m \in prosumers$) **do**

        $amount_{left}$, prosumers $\leftarrow$ find Prosumers within m

        $p_g \leftarrow$ calculate price using equation 6.7

        for ($p \in prosumers$) **do**

            $c_m \leftarrow p_g * et_m$

            $C_{total} \leftarrow C_{total} + c_m + P_r$

            $C_{share} \leftarrow C_{share} + P_r$

        end for

    end for

end if

if ($amount_{left} \neq 0$) **then**

    $c_u \leftarrow amount_{left} \times P_u$

    $C_{total} \leftarrow C_{total} + c_u$

end if

Algorithm 7 Energy request handling algorithm (Cont.)

if (reqType == 'Trade') **then**
  consumer c pays C_{total}
else
  if ($Sharing == True$) **then**
    consumer c pays C_{share}
  end if
end if
$E^t_{traded} \leftarrow E^t_{traded} + amount$
if ($Load^t_{predicted} \leq E^t_{traded}$) **then**
  $Sharing \leftarrow True$
end if
End Function

6.4 Energy Sharing

Other than buying/selling the energy, the prosumer might share the energy with the fellow participants for fulfilling their energy requirements in return for the benefits in the future that we will discuss later. Just like trading, the prosumer will share the information and the transaction will be done indicating that prosumer and the consumer for future references. Another aspect is the sharing of resources. The definition of sharing in our work deviates from the P2P trading in this sense. One scenario to make this idea apparent can be realized in a microgrid setting. Consider a microgrid with multiple houses holds. Some of them have RES while some have the storages available while others have both of them. In case of an increase in the energy load of one house, it can either increase the capacity of the RES or increase the storage capacity. Both of these actions require extra cost and infrastructural changes on the client's side.

The alternative solution to this can be sharing central energy storage or sharing the infrastructure of other participants. In the first case, the prosumers and consumers can mutually buy central storage that can hold the energy which can be distributed when required. Again it will cost to buy this kind of storage. In the second case, the use the batteries of the other participants to store the energy. This has potential to decrease extra cost for infrastructure.

All the cases of sharing mentioned above are indirectly related to each other. Assume a consumer gets energy from prosumer, the consumer is indirectly using a battery and the solar panel as the prosumer must be keeping the surplus energy on the batteries and which will be recharged once the energy is shared. In this case, the consumer is sharing the battery and RES of the prosumer as well. So a focus on direct energy sharing would suffice to incorporates all the cases and will reduce the complexity of the problem.

6.4.1 Limitations in Energy Sharing

There is a strong limitation over the scenario where energy sharing is feasible. This is because energy sharing promises a free supply of the energy on the prosumer account. There is no incentive for prosumers to share the energy and consumers are also free to get the free energy (through sharing) rather than buying it from the prosumers. If the prosumer agrees on sharing the energy, it will cost prosumer more, and eventually prosumer will leave the system, and if the consumer pays for the energy, that comes under the energy trading paradigm. ***The only case where it can work is when all the transactions for some time are completed and there are consumers that still require energy.***

The problem with that assumption is that it infers that there is a prosumer, let call it 'P1', that has energy left to sell and there is a consumer, call it 'C1', that needs energy. The two contradict each other because in that case, 'C1' would have bought the energy during the trading period. If no consumer is buying the energy, that means the energy requirements of all the consumers are already being fulfilled. So we have to add another assumption that ***the remaining consumers do not have the funds to buy the energy so they have to rely on the sharing of the energy.***

6.4.2 Energy Sharing Workflow

The workflow is similar to the workflow of energy trading with minor differences. The system goes to the sharing phase after few time windows. The consumer can send the sharing request afterward that is verified by the request handler. The main process will check if the sharing is available in the microgrid. If available, the energy will be shared without any cost. If not, the process will contact the other microgrids for sharing. In this case, the consumer will have to pay for the cost of the utility grid.

6.5 Federated Learning

Federated learning was first proposed and implemented by Google for the next word prediction [Hard et al., 2018]. This approach is a form of machine learning that is performed in a distributed fashion. Federated learning is an ideal machine learning approach when dealing with privacy-sensitive data or a huge amount of local data. Moreover, this increases the amount of available computational power where the number of devices available is more than nodes in a data center.

A very basic concept of federated learning iterations and round is as follows: First, a global model is created by a centralized authority that also decides on the number of rounds to conduct and the number of nodes to be selected in each round. Then the centralized authority starts the learning process. In each round, every selected node receives the global model, computed local gradient using local data, updates the weights of the global model, and sends the new model to the server where *federated averaging* is performed to create a new global model. *Federated averaging* is an algorithm to combine all the local updates done by the individual nodes into a single global model [McMahan et al., 2017]. To improve the efficiency of the overall system, this algorithm adapts different parameters. For example, it checks if the node has enough local data to perform gradient descent. Another approach is to check if the average loss of the new model is less than the loss of the global model.

Federated learning has fewer privacy risks than a centralized server as actual data is never sent to the main server. Moreover, the weights for the global model are calculated in memory and are discarded afterward. Several works are being done to improve the security and privacy of federated learning such as secure aggregation [Bonawitz et al., 2016] and differential privacy [Geyer et al., 2017].

6.5.1 Mapping FL to Energy Trading/Sharing

Federated learning has been used for several use-cases. [Taïk and Cherkaoui, 2020] uses federated learning for individual consumer load forecasting. A simulation using federated Long Short Term Memory LSTMs over Tensorflow is presented and the results show that the learning approach generates comparable results to the centralized learning techniques. In another work [Saputra et al., 2019], federated learning along with edge computing is used to predict the energy load for electrical vehicles using Deep Neural Networks (DNN).

As discussed earlier, the sharing technique requires the global prediction of energy load and energy generation values for all the prosumers and the consumers. Centralized machine learning can be used to have such a global model but as discusses in 6.5 these approaches have limitations that can be satisfied by the use of federated learning. Hence federated learning is used to predict the energy load of the system in the next time step. The same model can be used for other applications that will be discussed in section 6.5.4.

6.5.2 Improvement in Network Load

From earlier discussion, it is shown that federated learning helps in improving the privacy and security of the system, yet another important feature is the improvement in Network load when using federated learning. In the conventional machine learning, the participants have to send all training data to the server for it to do machine learning iterations. This imposes a huge load over the network as training data is usually in *Gigabytes*. Sending all this data negatively impacts network performance. While in federated learning, only weights of the model are sent to the server that is merely a fraction of the actual data (in kilobytes) hence decreasing the load over the network.

Table 6.1: Federated learning related literature review summary.

Article	Application	Type of Learning	Improvement	Shortcoming
[McMahan et al., 2016]	Global Aggregation for reduced communication	DNN federated learning	Decreased communication cost	Increased computation cost
[Wang et al., 2019c], [Liu et al., 2019b]	MEC-inspired edge server	Federated learning	Improved aggregation scalability issue	System Scalability issue
[Li et al., 2019]	Fair resource allocation to reduce variance of model performance	CNN and DNN learning	Improved resource allocation,	Introduce convergence delays
[Saputra et al., 2019]	Energy demand prediction (EV)	DNN Federated learning with clusters	Reduced communication overhead	Edge services not used
[Liu et al., 2019a]	Electric load forecast, do short term load forcasting with stocastic consumption profiles	LSTM federated learning with edge computing	Improved privacy and reduced networking load	Limited Accuracy
[Taïk and Cherkaoui, 2020]	Using cloud, edge and clients for improved learning	Hierarchical Federated Learning, handling Non-IID data	Improved convergence, used of edge computing	Require updating hyper-parameters

In this section, the farmulization of this phenomenon is presented. To calculate the gain in networking load the following equations can be used:

$$L_m = \sum_{i=1}^{N} S_i \tag{6.14}$$

$$L_f = 2 \times S_m \times NP \times R \tag{6.15}$$

where L_m is load for machine learning and L_f is the load for federated learning while S_i is the size of data sent by participant i, S_m is the size of the federated model, NP is the number of participating nodes in a round and R is the number of rounds performed. The factor of '2' is present in equation 6.15 because S_m amount of load is sent twice, once from server to client (while fetching the model) and then from client to server (for aggregation). Using equations 6.14 and 6.15, the gain is calculated as follows:

$$G_f = 1 - \frac{L_f}{L_m} \tag{6.16}$$

6.5.3 Federated learning and Blockchain

The use of federated learning assures better security by taking the computation to the participating peers but has inherent risks associated with centralized systems. As blockchain has emerged as a decentralized solution to many other applications, it has found its way to improve federated learning. [Lu et al., 2019] and [Kim et al., 2019] are very first works on the concept of on-device federated learning using blockchain. They create a designated blockchain that keeps the global model weights and other user information to conduct decentralized federated learning.

Other than creating new models of blockchain, using available blockchains such as Ethereum and Bitcoin is another promising solution for federated learning. The attempt to run aggregator free federated learning is proposed in [Ramanan et al., 2019]. The authors perform LSTM based learning to predict the location for a taxi driver that would generate maximum profit. In this work, we propose a Convolution Neural Network (CNN) based federated learning to predict the total load in the system using aggregation over the smart contract. Blockchain is used for both keeping the global model and doing basic federated averaging as depicted in algorithm 8. The local updates and predictions are performed off-chain (more description is section 7.2.1). Up to our knowledge, this is a first attempt of doing CNN federated learning over smart contracts to predict energy load.

Table 6.2: Resulting RMSE and MAPE for global models in the considered scenario with 20 nodes and 5 rounds.

Model	RMSE	MAPE	Epochs
Machine Learning	0.0116	1.39 %	20
Federate Learning	0.0237	2.23 %	100
Federate Learning Blockchain	0.0332	3.34 %	100

6.5.4 Other Applications

For this work, the blockchain-based federated learning was used to predict load of the system but the learning model (using smart contract) is designed such that with the appropriate amount of labeled data, the same model can be used for:

1. Predict the energy generation for prosumers.

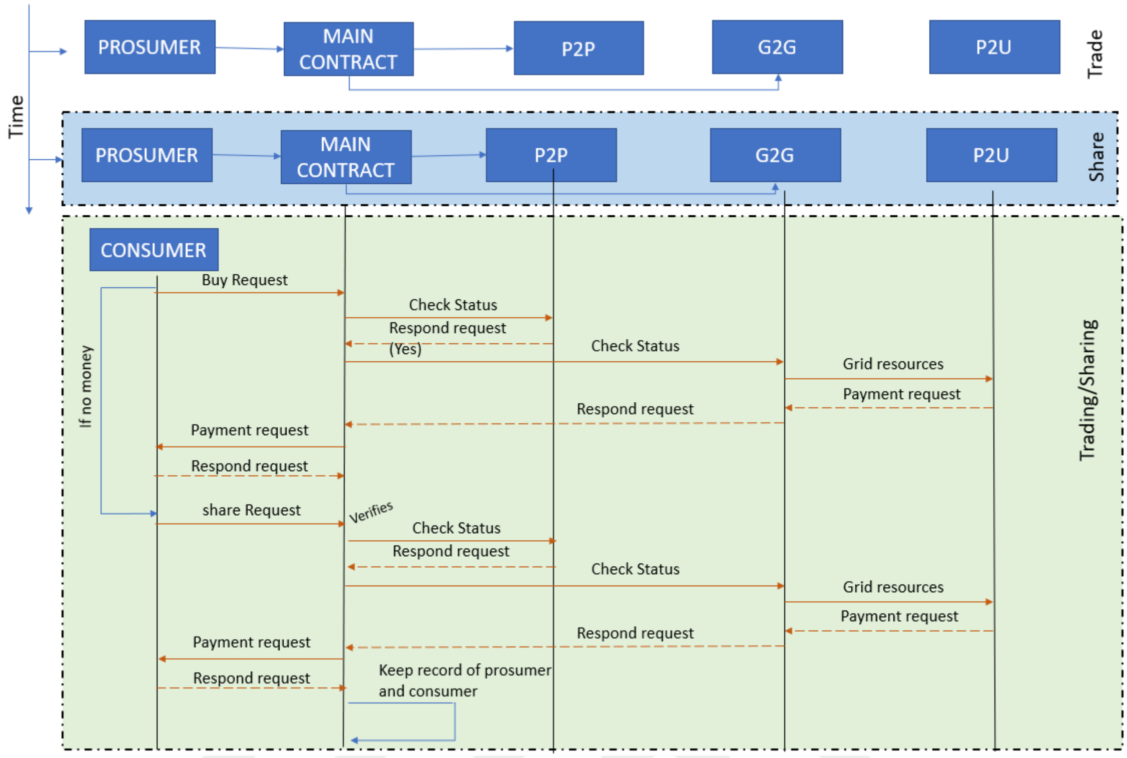

Figure 6.2: Interaction Between System Components and Participants.

2. Predict the energy load of individual consumers as being done in [Ramanan et al., 2019].

3. The amount of energy to share/trade in the next time step.

4. Predict the price of energy price in the next time step.

All of these predictions take in the data with specific fields such as weather information and the location of specific prosumer and consumer and output the predicted value. Hence, all of these predictions can be done using the Convolution layer and dense layer as described earlier and can be optimized using a different number of layers. As the presented model is flexible and can handle a different number of layers, it can be used for such predictions.

Algorithm 8 Blockchain Federated Averaging Algorithm

input: Initial model M, number of nodes to select NP, learning rate γ, number of total Rounds R
selected $= []$;
$\boldsymbol{M_g} \leftarrow$ Global model on blockchain set to M
initialize training round $\mathbf{r} = 0$;
while $r < R$ **do**
  selected $\leftarrow$ Select $\boldsymbol{NP}$ nodes from participants;
  for $s \in \textit{selected}$ **do**
    if s has local data **then**
      $m_l \leftarrow s$ fetches the model $\boldsymbol{M_g}$ from blockchain;
      $e \leftarrow s$ get number of epochs from blockchain;
      s computes local gradient g_l for epochs e;
      s updates the local model $m_l \leftarrow m_l - \gamma g_l$;
      s sends updated model to blockchain;
    end if
  end for
  Upon receiving NP models, the global model is formed using federated averaging over blockchain;
  next round start, $r \leftarrow r + 1$
end while

Chapter 7

FEDERATEDGRIDS: IMPLEMENTATION AND RESULTS

This chapter presents the implementation details of FederatedGrids and the detailed working of each component. The chapter starts with discussion of smart contracts. Then the implementation of federated learning is explained. The chapter concludes with experimentation setup and results discussion.

7.1 Smart Contracts

Main Smart Contract: A main smart contract is developed to control all operations of energy trading/sharing within the system. All participants directly communicate with this smart contract. This contract is used for:

1. Registering new prosumers and consumers.

2. Registering the microgrids.

3. Match prosumers and consumers within and across the microgrids for energy trading and sharing.

4. Calculation of pricing for the next time step and compensating the prosumers for their provided energy.

All participants (prosumers or consumers) need to be registered before generating any requests. All the energy surplus and deficit requests are also handled through the main contract. Upon receiving the energy surplus request, this contact updates the information of the M2M contract and P2P contract. In case an energy buying/sharing request is received, it first checks the type of request. If the request is for buying, it calls *findProsumer* function of P2P smart contract that matches the consumer with local prosumers. In case there is still energy to be bought, the *findMicrogrid* function is called that finds the microgrid with surplus energy for energy trade. The total cost for energy is calculated and transferred from proper buyers to the match sellers. In case there is still energy to be bought, the consumer's contact grids to buy the energy. In case the energy request is for sharing, the main contract first checks if the system is in the sharing phase. Then uses the same matching mechanism but this time, no transfer of resources is done instead all prosumers and consumers, participating in the sharing, are recorded in a list to be compensated later.

P2P Smart Contract: P2P contract is responsible to keep the information of the local prosumers and consumers when any participant is registered through the main contract. The participants in the system can not invoke the P2P contract directly. All the necessary information and commands for invocation can only be sent from the main contract to the P2P contract. The *findProsumer* function of the smart contract actually checks the microgrid id of the consumer and find the list of prosumers that can fulfill the energy demands of the buyer.

M2M Smart Contract: M2M contract keeps the information of all the microgrids in a dictionary-like data structure called maps. If there is surplus energy within a microgrid and there is energy requirement in another microgrid, M2M smart contract is used to facilitate consumers through energy trade amongst the microgrids. The price of this energy transaction depends on the price of energy provided by the utility. *findGrid* function in a smart contract is used to find an appropriate grid that can fulfill the energy demand of the consumer.

P2G Smart Contract: This smart contract is basically created to keep the energy price and resource price of the utility. Moreover, once all the energy transactions between peers and microgrids are done, this contact can be used by the buyers to buy the energy. This smart contract is directly accessible by the system participants and is deployed only by the utility grid.

FL Contract: FL (federated learning) contract keeps the global model in addition to performing the following operations:

1. Selecting the participants for the next round.

2. Providing and collecting the global model to/from the participants.

3. Does the aggregation at the end of the round performed.

The contract comprises of three main functions. *GetModel* is used to get the global learning model parameters from the contract. Each participant sends the request but only the selected participants are able to receive the model. Another important function is *SetModel* that is used to send in the updated parameters by the participants. This function keeps the count of submitted models and perform a comparison with the number of selected participants in the round 'n'. When 'n' number of models are received, SetModel calls another function *FederatedAveraging* that performs the parameter averaging of all the submitted models.

The overall system can be deployed by any entity with access to smart contract codes. The smart contracts are designed in a way that they can be deployed both over the public blockchain and the privately-owned blockchain. The benefits of using public blockchain are faster creation of blocks as there are many miners available while using a private blockchain, the participants can be limited and as most of the miners are self-owned, there is less probability of attack occurrence.

7.2 Experiments and Results

In this section, a energy trading system is simulated. Smart contracts are developed over *Solidity language* The smart contracts were written on *Remix IDE*[1] for testing the correctness of the contracts. The smart contracts were then deployed over local Ethereum using the Ganache framework and Geth to understand the performance of the system. Following constraints on the system are kept for energy requests:

1. All the participants have to register to the system using the main contract.

[1] Remix - Ethereum IDE. [online] Available at https://remix.ethereum.org [Accessed 20 June 2020].

2. A participant can either be a prosumer or a consumer in a specific timestep. The set of prosumers and consumers are disjoint.

3. A price for the microgrid to microgrid trading is calculated before the start of the time step.

4. Energy trading price between for M2M should be lower than the utility price and price for P2P should be lower than M2M energy price.

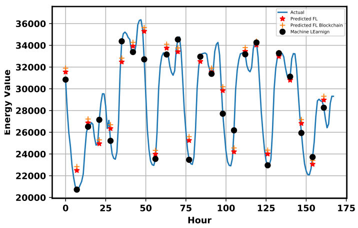

Figure 7.1: Predicted energy load for per hour.

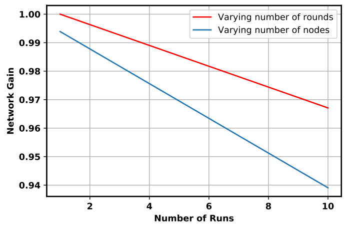

Figure 7.2: Total network gain achieved.

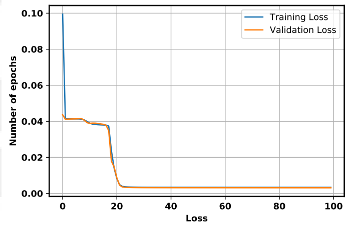

Figure 7.3: Training vs Validation loss

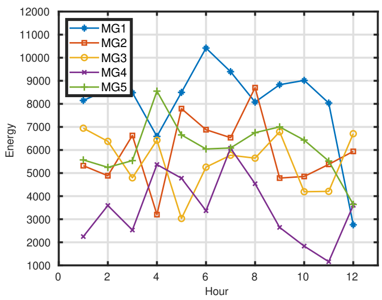

Figure 7.4: Energy Load of each microgrid.

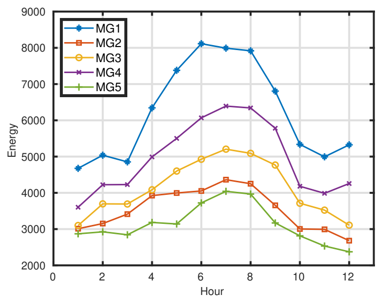

Figure 7.5: Energy generation of each microgrid.

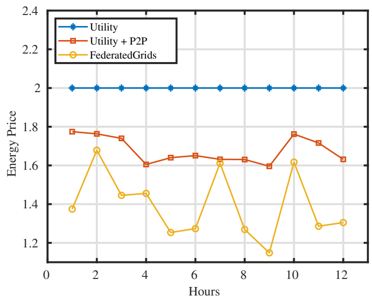

Figure 7.6: Average energy price with 5 microgrids.

All participants communicate with smart contracts through Energy Management Systems (EMS) that is developed using Python Language with *web3*[2] library. Moreover, each participant has an e-wallet that has different addresses. These wallets are

[2]Web3py.readthedocs.io. 2020. Introduction — Web3.Py 5.11.1 Documentation. [online] Available at: https://web3py.readthedocs.io/ [Accessed 20 June 2020].

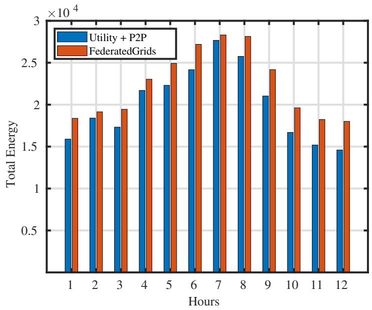

Figure 7.7: Total energy sale with 5 microgrids.

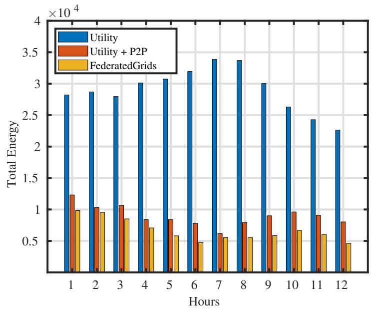

Figure 7.8: Load over utility with 5 Microgrids.

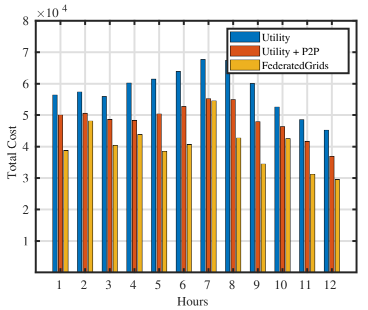

Figure 7.9: Total consumer cost with 5 microgrids.

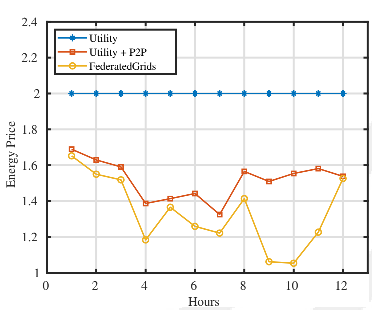

Figure 7.10: Average energy price with 3 microgrids.

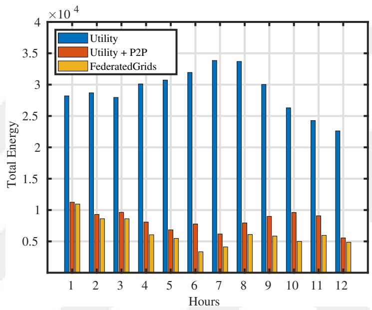

Figure 7.11: Load over utility with 3 Microgrids.

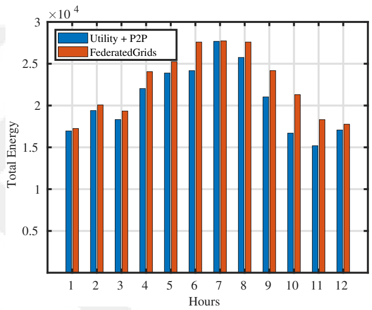

Figure 7.12: Total energy sale with 3 microgrids.

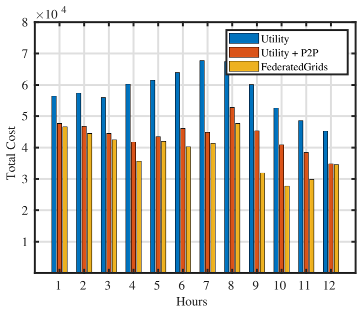

Figure 7.13: Total consumer cost with 3 microgrids.

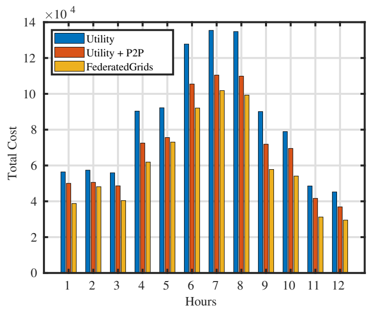

Figure 7.14: Total consumer cost with 5 microgrids and varying utility energy price.

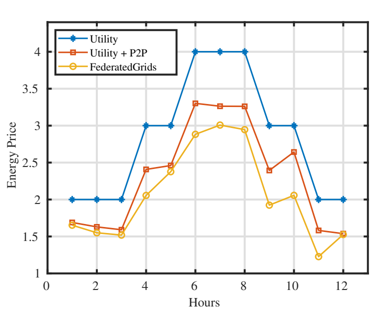

Figure 7.15: Average energy price with 5 microgrids and varying utility price.

provided by Ganache. Web3 library loads wallet using *web3.eth.accounts* function. The system is also connected with the utility grid to get energy if required. There are two types of energy consumers, one with resources to buy the energy and sec-

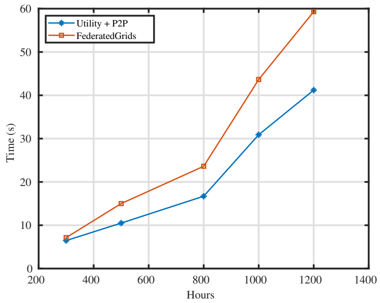

Figure 7.16: Average request completion time varying number of energy requests.

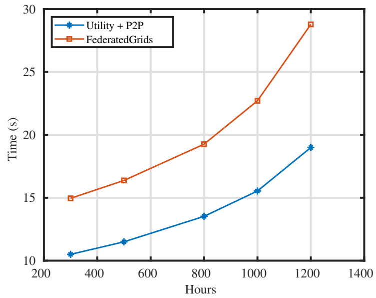

Figure 7.17: Average request completion time varying prosumers.

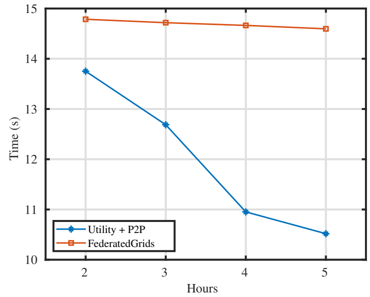

Figure 7.18: Average request completion time varying number of grids.

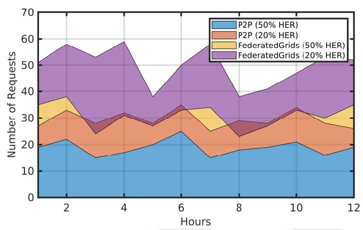

Figure 7.19: Fulfilled share requests with small and large energy requests.

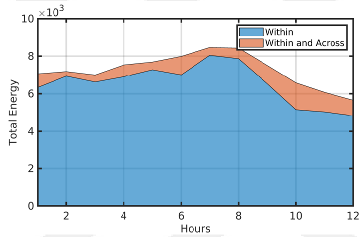

Figure 7.20: Energy shared within and across micro-grid.

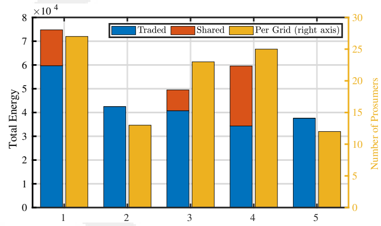

Figure 7.21: Energy shared and traded by each migro-grid.

ond without the available resource to fulfill energy buying requirements. The energy prosumers produce energy from Renewable energy resources such as PV panels and windmills. The energy transaction is executed using the blockchain assisted *FederatedGrids* system suggested in this work. Different studies are made that prove the efficacy of the system that i.e. increase in the energy profit of the prosumers and decrease in energy prices of energy consumers. An added advantage is for utility grid where the load over utility decreases, increasing the stability of energy in an area where utility works.

The Hourly Energy Consumption data-set [3] containing data from renewable sources from Spain and took the hourly energy values for our experiments. The uti-

[3] Jhana, Nicholas. "Hourly Energy Demand Generation and Weather." Kaggle, 10 Oct. 2019, www.kaggle.com/nicholasjhana/energy-consumption-generation-prices-and-weather. Last Access: 31 March 2020

lized dataset contains 4 years of electrical consumption, generation, pricing, and weather data for Spain. Consumption and generation data were retrieved from ENTSOE a public portal for Transmission Service Operator (TSO) data. Settlement prices were obtained from the Spanish TSO Red Electric España. The dataset also contains the Weather data which was purchased as part of a personal project from the Open Weather API for the 5 largest cities in Spain. The idea behind the information is that renewable energy is hugely affected by weather conditions in the area. The prediction method can be used to know about future production and the consumption of energy in the area. Using that value, a self-adaptive grouping is visualized. The dataset contains many fields including *generation biomass, generation fossil brown coal/lignite, generation fossil coal-derived gas, generation fossil gas* but our study focuses on the energy generated by the renewable sources and hence the fields of *Generation wind offshore, generation wind onshore, a solar day ahead, wind offshore day ahead, forecast wind onshore day ahead, total load forecast, total load actual, price day ahead, price actual* are used for the experimentation. The values of the price and the energy generation are used in the grouping and the values of the forecast are used for decision making for either executing the grouping or doing the transaction independently.

The energy consumption was then randomly distributed among participants within a different number of microgrids. This information is kept over smart contracts. The prosumers send the new value for the energy contribution for the hour ahead. Once the system goes online, random buying requests are generated (the overall sum is less than the total energy consumption per hour). Additionally, we create a total of *3* and *5* groups for the experiments. There is added cost for the blockchain transactions that have to be paid. So the final sales of the system are calculated by adding the transaction cost from the overall energy cost.

7.2.1 Federated Learning Simulation Setup

The simulations were conducted over HP Spectre X360 with an Intel i7 processor and 16GB memory. Keras CNN model was used to run the local training. The resultant weights are then extracted and sent to Ethereum Smart Contracts (developed over Solidity) using the function *set_model)* that takes weights as input. Upon receiving N models, the *federated_averaging)* function is called to perform weight averaging. The participating nodes receive global model using *get_model* function that returns model weights. These weights are then plugged in Keras model to run epochs for the next round.

Hyper parameters and optimizations are important for deep learning models, however this work focuses on giving proof of concept of federated learning paradigm for load prediction using blockchain. Moreover, a very simple model using 1 CNN layer and 2 Dense layers is used to reduce the amount of weights (and hence the size) of the model. For the model evaluation, RMSE and MAPE is used (shown in table 6.2 the expressions for which are:

$$RSME = \sqrt{\frac{\sum_{i=1}^{NP}(ya_i - yp_i)^2}{NP}} \tag{7.1}$$

$$MAPE = \frac{100\%}{NP}\sum_{i=1}^{NP}\frac{ya_i - yp_i}{ya_i} \tag{7.2}$$

where yp_i the is predicted value, ya_i is the actual value and NP is the number of predicted values.

7.2.2 Federated Learning Results

For prediction evaluation, three types of models were used. (1) Machine learning model, (2) Federated Learning centralized. and (3) Federated Learning blockchain. Table 6.2 shows the RMSE and MAPE values when models are trained. Here 20 epochs were used for machine learning models while 100 epochs per participant were used for federated learning models. This is because when the amount of data decreases, more epochs are required to get desired results. Moreover, there were 5 rounds with 20 nodes in each round, in federated learning scenarios. The MAPE results show that federated learning performs comparably to machine learning. The MAPE for the blockchain model is high because Solidity does not handle floating-point numbers, hence techniques were used to store weights over Solidity such as multiplication with 2^{32}. Figure 7.1 and 7.3 shows the prediction results and training/validation loss. As can be seen, the federated learning predictions are compared with machine learning.

For evaluating the gain of the network load, we calculated the size of the model which is 0.312Kb, and the size of data which is 25.6Mb. These values are then used in equation 6.16 to calculate the gain. To see the effect number of rounds and the number of participants overload, different values for R and NP were used and the graph was plotted as shown in 7.2. As can be seen, the effect of the number of rounds over gain is lower than that of the number of participating nodes. Also, the maximum of 99.7% gain was realized when using 1 round with 5 participants in that round.

7.2.3 Cost and Profit Analysis in Energy Trading

To validate the performance of the system in terms of consumer energy cost and prosumer profit, the energy production and energy load values are taken from the dataset. The system is tested for different scenarios. First, load and generation values are divided into five microgrids. Moreover, the total prosumers in the system are considered to be 300, while total consumers to be 500, divided amongst the microgrids. Figure 7.4 and 7.5 shows the load and generation respectively, of each microgrid in the system. A total of 12 hours are considered where hours between 6-10 are peak hours. Once the registration of prosumers and consumers is done, the energy trading requests are sent. Figure 7.6, 7.7, 7.8 and 7.9 show the average energy price, total prosumer sale, load over utility and total consumer cost respectively. It can be seen that the proposed model improves the system by decreasing the load over utility when compared to the model that considers only P2P trading. This reduction stabilizes the utility grid and prevents the possibility of blackouts. For the consumer side, the system reduces the price of energy (figure 7.6) that results in a decrease in the total cost of the consumer as shown in figure 7.9. These reduced prices motivate consumers to buy from local markets and hence increase the energy sale of prosumers (figure 7.7). This shows that the system benefits all participants including utility grids, prosumers, and consumers.

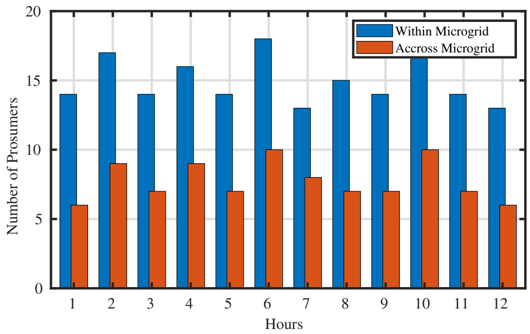

Figure 7.22: Prosumer sharing energy within grid and across grids.

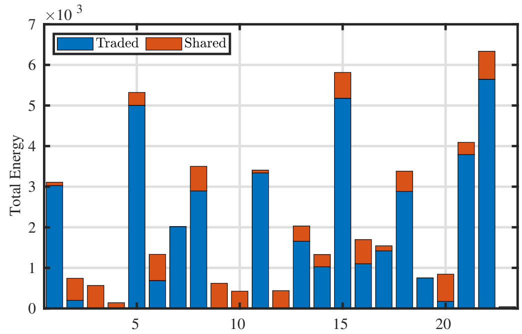

Figure 7.23: Total Energy shared and traded per prosumer for grid 3.

Another set of results was produced to understand the effect of the number of microgrids, while keeping the same amount of total prosumers and consumers, over the cost and profit of the system. Figure 7.10, 7.11, 7.12, and 7.13 show the average energy price, total prosumer sale, load over utility and total consumer cost respectively. It was noticed that the decrease in the number of microgrids effects utility+P2P trading more than the proposed model. As can be seen by figure 7.11, there is no increase in the sale of prosumers in the case of the proposed system but the increase in sales is noticed for utility+P2P trading. This is because now more energy resides in local microgrids, which increase the chances of consumers getting enough energy from the local grids. The effect of microgrids can be seen in figure 7.10. The decrease in energy prices can be seen. This trend is in accordance with equation 6.7, here more energy availability within microgrids decreases the energy price, hence decreasing the overall energy cost. Figures 7.14 and 7.15 shows the effect of varying price on the total cost and average energy price of the system. The energy price of the proposed system stays less with different utility energy prices.

7.2.4 Energy Sharing Results

This section discuss the effect of introducing energy sharing in the system. As argued in section 6.4, there can be a scenario where the consumers lack the resources to buy the energy and the prosumers have unsold energy. In this case, the consumer can benefit from extra energy of prosumers if the energy can be shared. To realize this, a system of 100 prosumers and 300 consumers is created that are divided into 5 microgrids. The total energy generation is presented in figure 7.5. After energy trading phase, 100 energy sharing requests are sent from the consumers. As can be seen in figure 7.19, several consumers benefited from energy sharing as there energy requests are completed. The figure also shows that more consumers are satisfied if small amount of energy is requested to be shared. This is because the system works on *first-come-first-served* basis. If requests for large amount of energy are sent, the system will run out of energy faster. It can also be seen that more energy requests are satisfied if the energy is shared within microgrid along with across the microgrids. This is because, now there is more energy to be shared (figure 7.20). Figure 7.22 shows that there are prosumers that can contribute towards sharing. The number of available prosumers effects the energy shared in a microgrid. Also, the system prefers to satisfy the consumers within the microgrid before sharing the energy with other microgrids. The total energy traded and shared in each microgrid (irrespective of energy sharing request amount) is shown in figure 7.21. It can be noticed that microgrid 2 and 5 do not have any energy left after trading phase but still can take advantage of surplus energy that microgrid 4 has (this is largest contributor). Moreover, energy summary of each prosumer for grid 3 is provided in figure 7.23. It can be seen that the prosumer with more energy have more contribution towards both trading and sharing. This is because the algorithm matches the energy request with the prosumer with the most energy. Overall, energy sharing has potential to create a collaborative environment, where the consumer satisfaction can be increased.

7.2.5 Network Performance

In this section, performance results for blockchain-based architecture are compared. There are two types of communication between participants and a smart contract. First is the registration request and the second is the energy request. The performance is evaluated by figuring average time to complete an energy request using equation 7.3. In this experiment, we exploited our system with a range of requests using:

1. Varying the number of prosumers in the system while sending 500 energy requests.

2. Varying amount of energy requests while keeping 500 prosumers in the system.

3. Varying the number of grids in the system while keeping 300 prosumers and sending 500 energy requests.

$$res = rec - req \tag{7.3}$$

Figure 7.16 compares the average request time by sending a varying amount of energy requests. The presented system has comparable performance with the systems that consider utility + P2P energy trading. The request time is more because, in case of FederatedGrids , the system has to do an extra search across the micro-grids. The time for request completion increases with the increase in the number of energy request. Another comparison of request time with varying amounts of prosumers is shown in figure 7.17. As the number of prosumers in the system increases, the search space for the system increases which affects the request completion time. The third result, shown in figure 7.18, depicts the effect of grid size on request time. As can be seen, the number grids effects FederatedGrids system more than Utility + P2P scenario. This is because an increase in the number of grids decreases search space for Utility + P2P systems as fewer prosumers are present in one grid, but this is not true for FederatedGrids case. The figures show that the system is scalable with very little overhead in comparison with utility + P2P energy trading systems.

7.3 Gas Cost Evaluation

FederatedGrids is built on top of smart contracts that are self executed when certain functions are called and transaction is created. There are 5 main type of transactions that can be made in FederatedGrids , namely submit model, aggregation, add

Table 7.1: Gas cost and price in Euro for different transactions.

Transaction Type	Number of Grids	Number of Prosumers	Gas Cost	Price in Euro
Submit Model	-	-	805042	0.25358823
Aggregation	-	-	1450655	0.456956325
Add Prosumer	5	-	247270	0.07789005
Add Consumer	5	-	181499	0.057172185
Energy Request	5	100	151946	0.04786299
Energy Request	5	200	188439	0.059358285
Energy Request	5	300	225194	0.07093611

prosumer, add consumer and energy request. There is a gas cost associated with each function call that varies with the type of commands a specific function on the smart contracts executes. The gas cost evaluation is summarized in Table 7.1. For example, submit model has less cost as it just takes in the model values and puts them in the dictionary like structure, while aggregation takes significantly larger cost as it goes over all the submitted models and perform mathematical operations on them. The gas cost for submitting and aggregation is paid by the participants of federated learning. This is a problem as there is less incentive for peers to take part in federated learning. A valid solution can be to pay them for their participation, however this is not in scope of this thesis. Other than that, a prosumer pays cost for adding itself and same goes for a consumer while energy request cost is paid by the consumer. As shown in the table, the overall cost is negligible taking in regard that large amount of energy transactions happening in the system. Hence these are not included in the experimental analysis performed.

Chapter 8

CONCLUSION

8.1 Remarks

To conclude, despite the current limitations on the use blockchain and distributed ledgers, they still have potential to transmute the marketplace of the energy sector. Blockchain has offered novel solutions to encourage RESs and local energy providers to play a role in relying on the energy resources that are cleaner and do not contribute towards the greenhouse effect. The works in this thesis provide detail review of blockchain technologies and how it can be used in decentralization of the energy trading paradigm. The overview of smart contracts and Ethereum architecture is also discussed in this thesis. The development of a prosumer management system in a decentralized manner is an important and challenging task. This is necessary for creating sustainable energy markets and providing comparable monetary advantages. [Ali et al., 2020d]. We first discuss the background of blockchain in P2P energy trading. Then, we enlist open issues in the field and propose three decentralized cyberphysical blockchain-enabled P2P energy trading models. We also walk the reader through the common design principle in such environments, analyze the models, and conclude the work with future research directions.

Addressing the significance of emerging P2P energy trading, we exploit the opportunities that blockchain provides within the energy trading setting. In our model, we utilize the blockchain to design a smart contract-based framework that ensures the real-time collaborative trade of electricity without the need for a central authority. Two different prosumer management systems are provides, that can be used separately or together to improve the profitability of the prosumers, hence encouraging more prosumers to join the system. The proposed solutions also have the potential to decrease the energy cost of the consumers. Both the implementation techniques provide in depth analysis in the form of two different case studies. The first system in SynergyChain , where grouping mechanism is developed over Ethereum smart contracts. We have discussed how it improves the overall profitability of the prosumers and have tested the system in different scenarios. FederatedGrids is proposed to create a framework for trading across multiple microgrids to ensure consumer satisfaction.

8.2 Future Directions

The design model of the application is first step towards the use of blockchain in energy trading. The more regerous studies can be done to improve on these designs that reduce the storage demand and the transaction creation time. Also, Currently, we have used the Ethereum platform for implementation. Hyperledger based architecture will be done to investigate how the system behaves with permission blockchain. Improving on that, different distributed ledgers such as hashgraphs can be explored as an alternate to blockchain technologies.

In SynergyChain , the predictor decides on whether to do grouping or not, but in the future, our predictor would also define the parameters of the grouping, for example, it will take into account the effect of the group size on the system. We also aim to utilize different heuristics or machine learning approaches that can be compared with P2P energy trading systems. Moreover, pays all the prosumers equally regardless of the individual energy contributions of the prosumers, our future work will involve a game-theoretical aspect where the prosumers with more energy contribution will be incentivized. Similarly, in FederatedGrids , with proper data, federated learning can be used to make smart decision on the amount of energy to be shared/traded. Moreover, a better optimization technique can be devised for improving prosumer profit and consumer satisfaction.

Other than above mentioned open issues, following are the direction energy trading research can go towards:

Integration of Energy Distribution without Smart Meters: Many energy distribution architectures used by prosumers/consumers do not involve smart devices. Since most of the energy trading solutions discussed here assume that prosumers and consumers are equipped with smart devices, a new system can be implemented that can incorporate these architectures with blockchain.

Blockchain as a Black Box: Most blockchain-enabled solutions use the blockchain as a blackbox. For example, several solutions [Dorri et al., 2019] [Wang, 2019] use smart contracts as a blockchain protocol for developing the architecture. This reduces control over the cost and efficiency of the overall architecture, as no changes can be applied to the blockchain used by the smart contracts. Problem-specific blockchains (and consensus) could be implemented for energy trading, rather than using blockchain as a black box.

Optimizations: Demand response and price optimization are still not fully addressed, and further research could include machine learning-based solutions for energy management and cost analysis for P2P energy trading. For example, a real-time billing system that can optimize the energy costs by current and predicted future prices, and charge the user accordingly.

Blockchain at Edge: Edge [Perk et al., 2020] computing is an efficient solution to cloud based applications. It brings the computing power closer to the users, hence providing improved latency and throughput. Currently, blockchain is being used in edge computing [Song et al., 2020] and a better solution for energy trading is possible if the two technologies are combined together.

BIBLIOGRAPHY

[gri,] Gridplus. `https://gridplus.io/`, Last Accessed: 2020-08-07.

[Ali et al., 2020a] Ali, F., Aloqaily, M., Ozkasap, O., and Bouachir, O. (2020a). Blockchain-assisted decentralized virtual prosumer grouping for p2p energy trading. In *21ST IEEE International Symposium on a World Of Wireless, Mobile And Multimedia Networks (IEEE WOWMOM)*. IEEE.

[Ali et al., 2020b] Ali, F., Aloqaily, M., Ozkasap, O., and Bouachir, O. (2020b). Federatedgrids: A hybrid peer-to-peer blockchain-assisted energy trade and share. (To be Submitted).

[Ali et al., 2020c] Ali, F., Aloqaily, M., Ozkasap, O., and Bouachir, O. (2020c). Synergychain: Blockchain-assisted adaptivecyberphysical p2p energy trading. *IEEE Transactions on Industrial Informatics (Under revision)*.

[Ali et al., 2020d] Ali, F. S., Aloqaily, M., Alfandi, O., and Ozkasap, O. (2020d). Cyberphysical blockchain-enabled peer-to-peer energy trading. *IEEE Computer*, 53(9):56–65.

[Andoni et al., 2019] Andoni, M., Robu, V., Flynn, D., Abram, S., Geach, D., Jenkins, D., McCallum, P., and Peacock, A. (2019). Blockchain technology in the energy sector: A systematic review of challenges and opportunities. *Renewable and Sustainable Energy Reviews*, 100:143–174.

[Androulaki et al., 2018] Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., et al. (2018). Hyperledger fabric: a distributed operating system for permissioned blockchains. In *ACM EuroSys*.

[Bonawitz et al., 2016] Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. (2016). Practical secure aggregation for federated learning on user-held data. *arXiv preprint arXiv:1611.04482*.

[Buterin et al., 2014] Buterin, V. et al. (2014). A next-generation smart contract and decentralized application platform. *white paper*, 3:37.

[Colthorpe, 2020] Colthorpe, A. (2020). Peer-to-peer trading of solar technically feasible, australia trial finds. `www.pv-tech.org/news/peer-to-peer-trading-of-solar-technically-feasible-australia-trial-.finds.`, Last Accessed: 2020-08-07.

[Contributor, 2020] Contributor, A. (2020). Japan turns to blockchain for trading extra power. `https://asiatimes.com/2020/06/japan-turns-to-blockchain-for-trading-extra-power/`, Last Accessed: 2020-08-07.

[Da Silva et al., 2013] Da Silva, P. G., Ilić, D., and Karnouskos, S. (2013). The impact of smart grid prosumer grouping on forecasting accuracy and its benefits for local electricity market trading. *IEEE Transactions on Smart Grid*, 5(1):402–410.

[De Vries, 2018] De Vries, A. (2018). Bitcoin's growing energy problem. *Joule*, 2(5):801–805.

[Dorri et al., 2019] Dorri, A., Luo, F., Kanhere, S. S., Jurdak, R., and Dong, Z. Y. (2019). Spb: A secure private blockchain-based solution for distributed energy trading. *IEEE Communications Magazine*, 57(7):120–126.

[Douceur, 2002] Douceur, J. R. (2002). The sybil attack. In *IPTPS*, pages 251–260. Springer.

[Engineer, 2019] Engineer, T. (2019). First blockchain-based solar energy trading initiative works. `https://www.theengineer.co.uk/blockchain-ev-charging-trust/#:~:text=Researchers.`, Last Accessed: 2020-08-07.

[Eyal and Sirer, 2018] Eyal, I. and Sirer, E. G. (2018). Majority is not enough: Bitcoin mining is vulnerable. *Communications of the ACM*.

[Gai et al., 2019] Gai, K., Wu, Y., Zhu, L., Qiu, M., and Shen, M. (2019). Privacy-preserving energy trading using consortium blockchain in smart grid. *IEEE Transactions on Industrial Informatics*, 15(6):3548–3558.

[Geyer et al., 2017] Geyer, R. C., Klein, T., and Nabi, M. (2017). Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*.

[Goranović et al., 2017] Goranović, A., Meisel, M., Fotiadis, L., Wilker, S., Treytl, A., and Sauter, T. (2017). Blockchain applications in microgrids an overview of current projects and concepts. In *IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society*, pages 6153–6158. IEEE.

[Hard et al., 2018] Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., and Ramage, D. (2018). Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*.

[Hester and et. al., 2018] Hester, T. and et. al. (2018). Deep q-learning from demonstrations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[Ilić et al., 2013] Ilić, D., da Silva, P. G., Karnouskos, S., and Jacobi, M. (2013). Impact assessment of smart meter grouping on the accuracy of forecasting algorithms. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 673–679.

[Inayat and Hwang, 2018] Inayat, K. and Hwang, S. O. (2018). Load balancing in decentralized smart grid trade system using blockchain. *Journal of Intelligent & Fuzzy Systems*, 35(6):5901–5911.

[Kang et al., 2017] Kang, J., Yu, R., Huang, X., Maharjan, S., Zhang, Y., and Hossain, E. (2017). Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains. *IEEE Transactions on Industrial Informatics*, 13(6):3154–3164.

[Khalid et al., 2020] Khalid, R., Javaid, N., Almogren, A., Javed, M. U., Javaid, S., and Zuair, M. (2020). A blockchain-based load balancing in decentralized hybrid p2p energy trading market in smart grid. *IEEE Access*, 8:47047–47062.

[Kim et al., 2019] Kim, H., Park, J., Bennis, M., and Kim, S.-L. (2019). Blockchained on-device federated learning. *IEEE Communications Letters*, 24(6):1279–1283.

[Knirsch et al., 2018] Knirsch, F., Unterweger, A., and Engel, D. (2018). Privacy-preserving blockchain-based electric vehicle charging with dynamic tariff decisions. *Computer Science-Research and Development*, 33(1-2):71–79.

[Li et al., 2019] Li, T., Sanjabi, M., Beirami, A., and Smith, V. (2019). Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*.

[Liu et al., 2019a] Liu, L., Zhang, J., Song, S., and Letaief, K. (2019a). Client-edge-cloud hierarchical federated learning. *arXiv preprint arXiv:1905.06641*.

[Liu et al., 2019b] Liu, L., Zhang, J., Song, S., and Letaief, K. B. (2019b). Edge-assisted hierarchical federated learning with non-iid data. *arXiv preprint arXiv:1905.06641*.

[Liu et al., 2017] Liu, N., Yu, X., Wang, C., Li, C., Ma, L., and Lei, J. (2017). Energy-sharing model with price-based demand response for microgrids of peer-to-peer prosumers. *IEEE Transactions on Power Systems*, 32(5):3569–3583.

[Long et al., 2018] Long, C., Wu, J., Zhou, Y., and Jenkins, N. (2018). Peer-to-peer energy sharing through a two-stage aggregated battery control in a community microgrid. *Applied energy*, 226:261–276.

[Lu et al., 2019] Lu, Y., Huang, X., Dai, Y., Maharjan, S., and Zhang, Y. (2019). Blockchain and federated learning for privacy-preserved data sharing in industrial iot. 16(6):4177–4186.

[Lukovic and Kovac, 2013] Lukovic, S. and Kovac, E. B. (2013). Adapting multi-agent systems approach for integration of prosumers in smart grids. In *Eurocon*, pages 85–91.

[Luna and et. al., 2016] Luna, A. C. and et. al. (2016). Cooperative management for a cluster of residential prosumers. In *International Conference on Consumer Electronics (ICCE)*, pages 593–594. IEEE.

[Lüth et al., 2018] Lüth, A., Zepter, J. M., del Granado, P. C., and Egging, R. (2018). Local electricity market designs for peer-to-peer trading: The role of battery flexibility. *Applied energy*, 229:1233–1243.

[McMahan et al., 2017] McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282.

[McMahan et al., 2016] McMahan, H. B., Moore, E., Ramage, D., Hampson, S., et al. (2016). Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*.

[Mengelkamp et al., 2018a] Mengelkamp, E., Gärttner, J., Rock, K., Kessler, S., Orsini, L., and Weinhardt, C. (2018a). Designing microgrid energy markets: A case study: The brooklyn microgrid. *Applied Energy*, 210:870–880.

[Mengelkamp et al., 2018b] Mengelkamp, E., Notheisen, B., Beer, C., Dauer, D., and

Weinhardt, C. (2018b). A blockchain-based smart grid: towards sustainable local energy markets. *Computer Science-Research and Development*, 33(1-2):207–214.

[Menniti et al., 2013] Menniti, D., Pinnarelli, A., Sorrentino, N., Burgio, A., and Brusco, G. (2013). Demand response program implementation in an energy district of domestic prosumers. In *Africon*, pages 1–7. IEEE.

[Merkle, 1980] Merkle, R. C. (1980). Protocols for public key cryptosystems. In *IEEE Symposium on Security and Privacy*, pages 122–122.

[Mihaylov and et. al., 2015] Mihaylov, M. and et. al. (2015). Scanergy: a scalable and modular system for energy trading between prosumers. In *International Conference on Autonomous Agents and Multiagent Systems*, volume 7, pages 1917–1918.

[Mihaylov and Jurado, 2014] Mihaylov, M. and Jurado, S. (2014). Nrgcoin: Virtual currency for trading of renewable energy in smart grids. In *11th International conference on the European energy market*, pages 1–6. IEEE.

[Muller, 2017] Muller, H. A. (2017). The rise of intelligent cyber-physical systems. *Computer*, 50(9):18–28.

[Mylrea and Gourisetti, 2017] Mylrea, M. and Gourisetti, S. N. G. (2017). Blockchain for smart grid resilience: Exchanging distributed energy at speed, scale and security. In *Resilience Week (RWS)*, pages 18–23. IEEE.

[Nakamoto, 2008] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

[Parag and Sovacool, 2016] Parag, Y. and Sovacool, B. K. (2016). Electricity market design for the prosumer era. *Nature energy*, 1(4):1–6.

[Park et al., 2018] Park, L. W., Lee, S., and Chang, H. (2018). A sustainable home energy prosumer-chain methodology with energy tags over the blockchain. *Sustainability*, 10(3):658.

[Paudel et al., 2018] Paudel, A., Chaudhari, K., Long, C., and Gooi, H. B. (2018). Peer-to-peer energy trading in a prosumer-based community microgrid: A game-theoretic model. *IEEE Transactions on Industrial Electronics*, 66(8):6087–6097.

[Perk et al., 2020] Perk, B., Bayraktaroğlu, C., Doğu, E. D., Ali, F. S., and Öznur Özkasap (2020). Joulin: Blockchain-based p2p energy trading using smart contracts. In *IEEE Symposium on Computers and Communications (ISCC 2020)*. IEEE.

[Petri et al., 2020] Petri, I., Alzahrani, A., Reynolds, J., and Rezgui, Y. (2020). Federating smart cluster energy grids for peer-to-peer energy sharing and trading. *IEEE Access*.

[Pop et al., 2018] Pop, C., Cioara, T., Antal, M., Anghel, I., Salomie, I., and Bertoncini, M. (2018). Blockchain based decentralized management of demand response programs in smart energy grids. *Sensors*, 18(1):162.

[Power, 2017] Power, P. G. (2017). Utilities.“. *Blockchain opportunity for energy producers and consumers.*

[Ramanan et al., 2019] Ramanan, P., Nakayama, K., and Sharma, R. (2019). Baffle: Blockchain based aggregator free federated learning. *arXiv preprint arXiv:1909.07452*.

[Rathnayaka et al., 2014] Rathnayaka, A. D., Potdar, V. M., Dillon, T., Hussain, O., and Kuruppu, S. (2014). Goal-oriented prosumer community groups for the smart grid. *IEEE Technology and Society Magazine*, 33(1):41–48.

[Rathnayaka et al., 2015] Rathnayaka, A. D., Potdar, V. M., Dillon, T., and Kuruppu, S. (2015). Framework to manage multiple goals in community-based energy sharing network in smart grid. *International Journal of Electrical Power & Energy Systems*, 73:615–624.

[Sabounchi and Wei, 2017] Sabounchi, M. and Wei, J. (2017). Towards resilient networked microgrids: Blockchain-enabled peer-to-peer electricity trading mechanism. In *IEEE Conference on Energy Internet and Energy System Integration*.

[Sabounchi and Wei, 2018] Sabounchi, M. and Wei, J. (2018). A decentralized p2p electricity market model for microgrids. In *2018 IEEE Power & Energy Society General Meeting (PESGM)*, pages 1–5. IEEE.

[Saputra et al., 2019] Saputra, Y., Dinh, H., Nguyen, D., Dutkiewicz, E., Mueck, M., and Srikanteswara, S. (2019). Energy demand prediction with federated learning for electric vehicle networks. In *IEEE Global Communications Conference (GLOBECOM)*.

[Sasson et al., 2014] Sasson, E. B., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., and Virza, M. (2014). Zerocash: Decentralized anonymous payments from bitcoin. In *IEEE Symposium on Security and Privacy*.

[Siano et al., 2019] Siano, P., De Marco, G., Rolán, A., and Loia, V. (2019). A survey and evaluation of the potentials of distributed ledger technology for peer-to-peer transactive energy exchanges in local energy markets. *IEEE Systems Journal*, 13:3454–3466.

[Song et al., 2020] Song, J., Gu, T., Ge, Y., and Mohapatra, P. (2020). Smart contract-based computing resourcestrading in edge computing. *arXiv preprint arXiv:2006.15824*.

[Sutton and Barto, 2011] Sutton, R. S. and Barto, A. G. (2011). Reinforcement learning: An introduction. Cambridge, MA: MIT Press.

[Szabo, 1997] Szabo, N. (1997). Formalizing and securing relationships on public networks. *First Monday*.

[Taïk and Cherkaoui, 2020] Taïk, A. and Cherkaoui, S. (2020). Electrical load forecasting using edge computing and federated learning. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE.

[Taïk and Cherkaoui, 2020] Taïk, A. and Cherkaoui, S. (2020). Electrical load forecasting using edge computing and federated learning. In *ICC 2020*.

[Thakur and Breslin, 2018] Thakur, S. and Breslin, J. G. (2018). Peer to peer energy trade among microgrids using blockchain based distributed coalition formation method. *Technology and Economics of Smart Grids and Sustainable Energy*, 3(1):5.

[Tsikalakis and Hatziargyriou, 2011] Tsikalakis, A. G. and Hatziargyriou, N. D. (2011). Centralized control for optimizing microgrids operation. In *IEEE power and energy society general meeting*.

[Veerapaneni, 2020] Veerapaneni, N. (2020). India: Deployment of blockchain technology to minimize investment risks in india's power sector. https://www.mondaq.com/india/fin-tech/952124/deployment-of-blockchain-technology-to-minimize-investment-risks-in\\-india39s-power-sector, Last Accessed: 2020-08-07.

[Vergados et al., 2016] Vergados, D. J., Mamounakis, I., Makris, P., and Varvarigos, E. (2016). Prosumer clustering into virtual microgrids for cost reduction in renewable energy trading markets. *Sustainable Energy, Grids and Networks*, 7:90–103.

[Wang et al., 2019a] Wang, N., Zhou, X., Lu, X., Guan, Z., Wu, L., Du, X., and Guizani, M. (2019a). When energy trading meets blockchain in electrical power system: The state of the art. *Applied Sciences*, 9(8):1561.

[Wang et al., 2019b] Wang, S., Bi, S., and Zhang, Y. J. A. (2019b). Reinforcement learning for real-time pricing and scheduling control in ev charging stations. *IEEE Transactions on Industrial Informatics*.

[Wang et al., 2019c] Wang, S., Tuor, T., Salonidis, T., Leung, K. K., Makaya, C., He, T., and Chan, K. (2019c). Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6):1205–1221.

[Wang, 2019] Wang, S. e. a. (2019). Energy crowdsourcing and peer-to-peer energy trading in blockchain-enabled smart grids. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49:1612–1623.

[Wang et al., 2019d] Wang, W., Hoang, D. T., Hu, P., Xiong, Z., Niyato, D., Wang, P., Wen, Y., and Kim, D. I. (2019d). A survey on consensus mechanisms and mining strategy management in blockchain networks. *IEEE Access*, 7:22328–22370.

[Xavier, 2020] Xavier, J. (2020). First blockchain-based solar energy trading initiative works. `https://www.thehindu.com/sci-tech/technology/first-blockchain-based-solar-energy-trading-initiative-works/article31895548.ece`, Last Accessed: 2020-08-07.

[Zhang et al., 2018] Zhang, C., Wu, J., Zhou, Y., Cheng, M., and Long, C. (2018). Peer-to-peer energy trading in a microgrid. *Applied Energy*, 220:1–12.

[Zhao and et. al., 2016] Zhao, D. and et. al. (2016). Deep reinforcement learning with experience replay based on sarsa. In *IEEE Symposium Series on Computational Intelligence (SSCI)*.