Reinforcement learning based electricity price forecasting in Blockchain based smart grid environment

by

Md Mahraj Murshalin Al Moti
17101301
Rafsan Shartaj Uddin
17101311
Abdul Hai Anik
17101312
Tanzim Bin Saleh
17101310

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
June 2021

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Md Mahraj Murshalin Al Moti
17101301

Rafsan Shartaj Uddin
17101311

Md Abdul Hai
17101312

Tanzim Bin Saleh
17101310

Approval

The thesis titled "Reinforcement learning based electricity price forecasting in Blockchain based smart grid environment" submitted by

1. Md Mahraj Murshalin Al Moti (17101301)
2. Rafsan Shartaj Uddin (17101311)
3. Abdul Hai Anik (17101312)
4. Tanzim Bin Saleh (17101310)

Of Spring, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on June 6, 2021.

Examining Committee:

Supervisor:
(Member)

Md. Golam Rabiul Alam,PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Electricity is deeply integrated into both our modern society and the economy. However, with our ever-growing society and increasing demand for electricity, the scarcity of resources is deeply felt through load shedding in most third world countries. Moreover, since most of the world depends on electricity systems built around more than 60 years ago, they are becoming increasingly inefficient and fail to solve the problems of modern-day global challenges. A Smart grid is an intelligent electricity network that allows efficient and optimal electricity distribution from source to consumers through smart integration of power technologies, information, and telecommunication through the existing system. The current system is a one-way interaction that only supplies electricity to consumers. That limits the ability to respond to the ever-changing and rising demands of society. However, smart grids allow the exchange of electricity and information between producers and customers. A smart home will communicate with the grid and allow consumers to manage electricity usage through a smart meter efficiently, and that will also efficiently manage electricity bills. Inside a smart home, the Home Area Network (HAN), will integrate all smart appliances into one energy management system so that these appliances can adjust the run schedule to lessen the demand on electricity at peak times, therefore, lowering bills. Reinforcement learning and a decentralized local market through block-chain can be used for electricity load and price forecasting. It is possible to fine-tune parameters to increase overall distribution and performance through efficient feature selection and feature extraction methods. The use of block-chain will connect prosumers and suppliers in a secure and decentralized system that will be used to forecast usage and bills. Also, through the use of reinforcement learning techniques and the block-chain's information, it will be possible to analyze prosumer behavior. So, the integration of block-chain and smart grids will increase flexibility and scalability, leading to an overall optimized system.

Keywords: Smart Grid, Block-chain, Price Forecasting, Electricity demand and supply, Smart Meter, Reinforcement Learning.

Dedication (Optional)

Our research is dedicated to the bright future of our nation and its citizens whom we envision to be part of a better country where there will be no more energy deficiency like the past. The electricity smart grid solution with block-chain implementation that we hope to discuss will make our country more digital, energy efficient and enable our fellow countrymen to live life with ease, security and minimal cost. We hope our research will shed more light on future improvements in electricity consumption to make underprivileged countries more prosperous.

Acknowledgement

All praises to Almighty Allah for being merciful and gracious for blessing us with the confidence, determination and eagerness to complete our research. We would like to express our sincere gratitude to our respected thesis supervisor and co-supervisor Md. Golam Rabiul Alam, Associate Professor, Department of Computer Science and Engineering, BRAC University. His guidance and immense knowledge have paved us the way to do our research with enthusiasm, confidence, motivation. We could not have imagined such an extraordinary supervisor for our research like him. We also like to express our gratitude to our family and peers who encouraged us and provided ongoing support to our work. Last but not the least, we want to express our thankfulness to all the people who support us directly or indirectly and interested in our research.

Table of Contents

List of Figures

List of Tables

List of Acronyms

The next list describes several symbols & abbreviation that will be later used within the body of the document

CU Customer

$DApp$ Decentralized Application

DPL Distributed Public Ledger

DR Demand Response

EVM Ethereum Virtual Machine

GO Grid Operator

MDP Markov Decision Process

$P2p$ Peer-to-peer

RL Reinforcement Learning

RPC Remote Procedure Call

SP Service Provider

Chapter 1

Introduction

Each country has its own economic and political system, and no matter if it is socialism, communism, or capitalism; the demand for electricity is ever-growing and will continue to be so in the future. Countries like Bangladesh, India, Lebanon, and many more suffer from the scarcity of electricity which results in load-shedding for hours at a time. Residential, Commercial, and Industrial areas are three different consuming categories that use the most power. Amongst them, the residential sectors dominate over the rest, especially in Bangladesh where the population is vast. It is in situations like this that Smart Grid comes into play. A smart grid is far better than a traditional grid primarily because it understands the consumption and distribution of energy, and it does so in a more efficient way. Traditional Grids tend to waste a lot of energy while generating and distributing electricity. In modern days, energy reservation is highly indispensable in this competitive world where cost and time efficiency are everything. A smart grid adds telecommunication feature with a traditional grid to make energy demand and supply more efficient and reliable. Perfectly utilizing this will almost completely solve the energy crisis to reduce and abolish blackouts. It is because of this that more and more cities are moving towards Smarts Grids. Countries like the United States, Denmark, and Korea are already heavily invested in Smart Grids. In short, the primary goal of a Smart Grid is to find and maintain a balance between the consumption of electricity and the demand for it using digital communication technology, which will be able to generate as well as react to changes in usage of a consumer. The implementation of Smart Grids happens from city to city and not nationwide directly. This depends heavily on Smart Meters and Smart Homes. Consumers at home have Smart Meters that monitor the usage of power and send the data to power generators to respond to their demands. It basically monitors when the usage of power is high and low, so that power plants can better direct electricity when only necessary to reduce wastage. In a Smart Grid, every consumer has a smart meter; and therefore, the collection of information is vast. Every Smart Grid has a database that is used for research purposes and billings. However, since this is a centralized system with no direct way to access it; it is hard for consumers to check the structure and information of which should be public information. Moreover, such a centralized system is highly prone to malicious attacks. One of the best ways to overcome this problem is to make a decentralized system where everyone can access the usage and billing information when necessary. Such a thing can be done with the help of blockchain where information is stored in the computers of consumers [41]. This not only reduces the burden of the Smart Grid, but it also stops it from failing at a single main point that can shut

down a city. Moreover, it provides immutability of the transactions as it will be written in smart contracts; this will ensure transparency between provider and consumer as there will be a proper transaction history that cannot be manipulated. Consumers can easily access the information on the blockchain to reduce the strain of consumption on peak-hours and shift consumption to times when there is a surplus of electricity. This kind of system also reduces the cost of electricity to a significant degree.
Demand response (DR) is an efficient method to use in smart grid system to reduce cost and improve grid efficiency. United Stated Department of Energy states that DR motivates changes of electricity prices over time so that it can incur low usage of energy during peak time usage [2]. DR can be categorized into two parts – price-based DR and incentive-based DR. Price based DR refers to influencing the customers' electricity usage with the variable electricity prices. And the latter being providing variable incentives based on electricity usage. To solve the dynamic pricing of a hierarchical energy market, this paper would like to propose a DR algorithm that copes up with the dynamic pricing and also helps to reduce the service provider's (SP) and customer's (CU) costs. With the help of Reinforcement Learning (RL) and Q-Learning we are predicting the price of electricity. The SPs get the flexibility to set the price dynamically with accordance to demand and level of dissatisfaction. Furthermore, blockchain integration to every customer's profile will help to secure the decentralized transaction of electricity between SP and CU.

1.1 Problem Statement

The biggest problem about traditional grids is that when they were first introduced, the plants only powered select few areas, and not entire cities and even countries. Therefore, efficiency was never a question during those times. In short, these traditional grids are now over a century old and even the dominating employee age group is now over 52 years old. Moreover, even in the USA, more than 70% of transformer lines and power transformers are over 25 years old [4]. Furthermore, these traditional grids were first made to power linear load with sinusoidal voltages. However, with the introduction of transistors and other modern devices on a large scale, power efficiency and convenience comes into question. Since these modern technologies are highly sensitive to the smallest voltage changes, a surge in power may even bring down entire computer servers, assembly lines, and control systems [4]. A century ago, the traditional grids were only used to power light bulbs across a country, and this caused almost no energy congestion. However, today, with increased demand and the aim to reduce cost; long range electricity transmission is used for reliability. This in turn, greatly increases stress over the entire network. A default implementation of Smart Grid keeps a simplified and centralized Smart Grid Database that is not only prone to hacks, but a small error will cause the entire system to shut down. Moreover, a consumer can request the grid's electricity demand through their Smart Meter which is then very prone to hacks. Moreover, privacy also comes into question since the attacker can then easily extract the electricity behavioral pattern of other consumers. Most current Smart Grids use cloud-based systems to store large amounts of data and anyone can send a request to access information of users. But, since the exchange of information between the cloud and consumer is not secured, a hacker can easily change data of users which can potentially alter the bills of other consumers. This is a big problem as current world is

heavily revolving around online transaction so security is a big factor.
In DR efficiency, we can see several works regarding DR models that help to subsidize customer usage by minimizing costs. One such example mentions in [3], [7], [10], where electricity consumption of different home appliances was monitored and time-of-use (TOU) pricing helped to minimize customers' costs. In [5], [11], [12], [14], [18], [20],we can see the benefit of predetermined next-day electricity prices, and efficient scheduling helps keep the costs of CU in check. Furthermore, in [24], besides the day-ahead price model, we also get to see the predetermined incentive-based model for customers. Though the contributions of the papers mentioned above helped the electricity demand response field immensely, it still lacks the dynamic market where demand is changing now and then. Thus, a DR strategy with dynamic pricing compatibility is impeccable for modern usage.
Another smart way to provide variable amount of service to customers is by dynamic pricing model which changes the price of energy time to time for perfect resource allocation [13]. This pricing model has been apparent in smart grid systems for quite a while. One of those implementations can be found where retailer profit has been maximized with quadratic programming problem [29]. Stackelberg games further influenced energy trading by determining the retail price from the energy usage scheme, and the customers minimized their usage of appliances according to the costs. This dynamic pricing model approach benefited though the prices massively fixed by the service providers were predetermined. Abstract models such as linear models were used, and these models can be inefficient in reacting to the ever-changing customer demands in the worldwide market. Thus, deterministic dynamic pricing approach and abstract linear models are sometimes unable to provide optimal performance due to any variable change in demand and resulting in loss of money and secondly, abstract models visualize an approximation of energy distribution and more dependent on the modeler's experience. This is where reinforcement learning comes into rescue as it is model-free and can efficiently react to the ever-changing demand of CUs and benefit both CUs and SPs. Addition to that, a decentralized blockchain system of end users' transaction can help maintaining the distribution of energy without any hassle.

1.2 Research Objective

This paper mainly proposes a smart system that balances electricity demand and supply, which will be implemented through a Blockchain-based network.

- Our primary goal is to propose and create a city and electrical grid where power congestion and load shedding will be eradicated. There are many third world countries where load shedding is a serious concern and there are cities and towns where there are blackouts for hours at a time. Our aim is to reduce these blackouts to zero.

- We can also use blockchain to negate any third-party intervention where it is unwanted. Moreover, cost consumption will also be reduced to a significant degree.

- A successful implementation of Smart Grid can also help introduce electric cars that are crucial to future developments, especially in countries like Bangladesh where the use and production of electric cars is being discussed.

- The smart system is implemented with the help of Q-Learning to determine retail price of electricity, as well as customers' demand and dissatisfaction level. It is highly efficient in response to demand and prioritized both retailer profit and customer satisfaction.
- Smart Contracts and authentication will be employed to bring consumers and producers together so that all users can participate in buying and selling. Moreover, they can also access information on the grid.
- Another important objective of Smart Grid is to encourage consumers to implement solar panels as other energy production methods so they too can contribute to the Smart Grid as a whole when electricity production is in a surplus.

1.3 Thesis Orientation

The chapters following this section are organized in the following orientation so that it closely matches the process in real life. Chapter 2 gives a brief description of most of the works and applications that exists for the blockchain market system, as well as the reinforcement learning mechanisms in the Smart Grid Industry. Chapter 3 describes all the similar studies that has been done on the similar topic and shows a detailed analysis on the paper at hand. Chapter 4 shows the proposed model of the blockchain that works as the medium for transaction between the retailers and customers based on cryptocurrency. Chapter 5 shows a detailed explanation on reinforcement learning, particularly on Q-learning through which the optimal price forecasting is done for both the customers and retailer. Chapter 6 explains on the implementation details as well as the algorithms and data that is used. Lastly, chapter 7 concludes the paper with a small summary, future scope of work, as well as references.

Chapter 2

Background Study

2.1 Blockchain

2.1.1 Overview of Blockchain

The blockchain is a decentralized database that stores and distributes records of all transactions and digital events.[44] It is a technology that allows for the digital exchange of units of value, much like the internet does. In other words, a blockchain network can tokenize, store, and trade everything from currency to land rights to votes. The system's majority of users double-checks every transaction, and it has each transaction's complete record. Blockchain technology is currently available in several forms. Some blockchains were established to meet the needs of a small group of users with limited network connectivity. This category includes private blockchains, often known as permission blockchains. Apart from safe value transmission, blockchain technology also provides a permanent forensic record of transactions and a single version of the truth where a network state is completely visible and exhibited in real-time to benefit all players. Furthermore, the blockchain is a decentralized database that stores and distributes records of all digital transactions and events. Every day blockchain is increasing its popularity, and because of that, more people are interested in using the blockchain. In October 2018, the blockchain size was 188 GB, where on May 18, 2021, the blockchain size is around 335.65 GB. So, with the growing size of the blockchain network, anyone can predict the popularity of blockchain after 5 to 10 years. In a smart grid, people are going to buy and sell electricity from the national grid. So, this transaction needs to store somewhere safe. That's where the blockchain is going to play a key role. Prosumers can use smart contracts to sell their unusable electricity through the blockchain. It will definitely improve power efficiency.

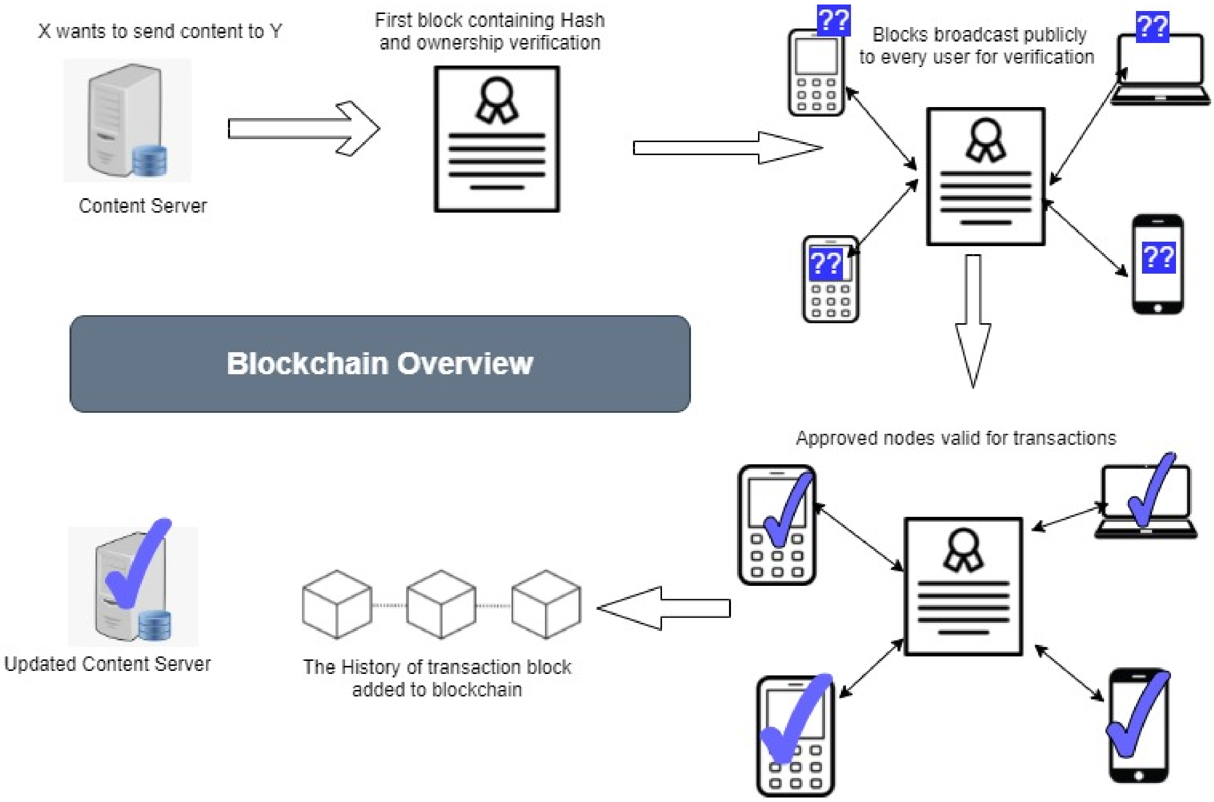

Figure 2.1: Blockchain Overview.

2.1.2 Blockchain History

Blockchain Technology is one of the greatest inventions of this 21st century. As it has started to gain popularity a few years back, many people do not know that the story of blockchain has been started in the early 1990s.
The idea behind blockchain technology was described as early as 1991 when research scientists Stuart Haber and W. Scott Stornetta introduced a computationally practical solution for time-stamping digital documents so that they could not be backdated or tempered with [38]. Their system involves a cryptographically secured chain of blocks where the time-stamped documents were stored. In 1992, Merkle trees were incorporated into the design making it more efficient by allowing more documents that could be collected in a single block. However, this technology became unused for a long time and the patent lapsed in 2004, four years before the inception of bitcoin.
In 2004, Hal Finney, a computer technology developer the RPoW (Reusable Proof-of-Work) network [45]. This system works by receiving a non-exchangeable or a non-fungible HashCash based on the proof of work token and in return created an RSA-signed token that could then be transferred from one person to another. This RPow can be referred to as an early cryptocurrency prototype that solves the double-spending problem by keeping the ownership of tokens registered on a trusted server. The server was designed in such a way that users throughout the world could verify its correctness and integrity in real-time.
In late 2008 a white paper introduced a decentralized peer-to-peer electronic cash system, named Bitcoin [38]. It was posted to a cryptographic mailing list by a person or group by using the pseudonym Satoshi Nakamoto based on the HashCash proof of work algorithm. However, rather than using hardware trusted computing function

like the RPow the double-spending protection in Bitcoin was provided by a decentralized peer-to-peer protocol for tracking and verifying transactions. Mainly, Bitcoins are mined for a reward using the proof of work algorithm by individual miners then confirmed by the decentralized network nodes. On 3 January 2009, Bitcoin came into existence when the first block of the Bitcoin blockchain was mined by Satoshi Nakamoto [45]. It had a reward of 50 bitcoins. The first recipient of Bitcoin was Hal Finney, he receives 10 bitcoins from Satoshi Nakamoto in the world's first Bitcoin transaction on 12 January 2009 [45].

In 2011, the idea of a Proof-of-State (PoS) consensus algorithm arrived. In 2013, Vitalik Buterin, a programmer and founder of the Bitcoin magazine stated that bitcoin needed a scripting language for constructing decentralized applications. Though he failed to gain agreement in the community, he started the development of a new blockchain-based distributive computing platform, Ethereum, that featured a scripting functionality called smart contracts. Smart Contracts are programs or scripts which deployed and executed on the ethereum. These smart contracts are written in specific programming languages and compiled into bytecode, which is a decentralized Turing- complete virtual machine known as the Ethereum virtual machine (EVM). Moreover, developers also able to create and publish applications running inside the Ethereum Blockchain. This application is known as decentralized apps (dApps). There are already hundreds of dApps running in the Ethereum blockchain.
However, blockchain history and evolution do not end with Ethereum and Bitcoin. In modern times, a remarkable number of new projects have cropped up for developing blockchain technology capabilities. For example, China has launched NEO, the first open-source, decentralized, and blockchain platform, despite the country has banned cryptocurrencies, it remains active when it comes to blockchain innovations.

2.1.3 The Characteristics of Blockchain

- **Better Security** Blockchain is a decentralized system, and every day it is growing rapidly. But how it maintains the security of the network? Basically, a blockchain [40] uses a unique cryptographic key to secure the blockchain network. It is virtually impossible to hack, and every time a new record is written on the same block, everything from their old record, including the content and key, is placed into a formula to create the key for the new one. [42]This interaction creates dependency. When a third block is created, the content of the third block is the content, and the keys of the first two records are put into a formula to establish the third key. So, every node makes it impossible to alter its previous block's history. The blockchain [39] uses SHA–256 cryptography algorithm to encrypt the data, and when an input is given on the blockchain, it generates a fixed-length random hidden value. Besides, Hashing is irreversible.

- **Faster Settlement** The primary benefit of blockchain technology is that [28] it can reduce settlement times by eliminating fragmented post-trade infrastructure. It also provides a more flexible settlement cycle. The traditional banking system requires 24 hours to ensure the transaction of the currency, but because of a decentralized digital network, blockchain can handle the transaction in a short time. Apart from that, for those who need faster transactions, blockchain allows faster transactions by paying additional truncation fees.

- **Decentralized** Blockchain is a decentralized network which means no single entity is in charge of running the network.[47] But what's the benefit of not having someone in charge? Because it is a decentralized network, the network is regulated by the majority of the nodes, and if we need to change something on the blockchain, it has to be approved by the majority of nodes. It makes it secure and extremely difficult to hack. So, if someone intends to hack the blockchain, they need to have a huge amount of computational power. Apart from that, as this system is established on algorithms, no one has the power to break from those algorithms' rules.

- **Immutability** The term immutability refers to unchangeable, which is one of the most crucial characteristics of blockchain. [39] In a blockchain network, every time a new node enters the blockchain, it automatically copies its previous node it's a cryptographic key. So, if anyone wants to alter any node, they have to replace all the nodes because of their background history entirely. This feature proved very beneficial in the case of cryptocurrency.

- **Transparency** Another important objective of Smart Grid is to encourage consumers to implement solar panels as other energy production methods so they too can contribute to the Smart Grid as a whole when electricity production is in a surplus.

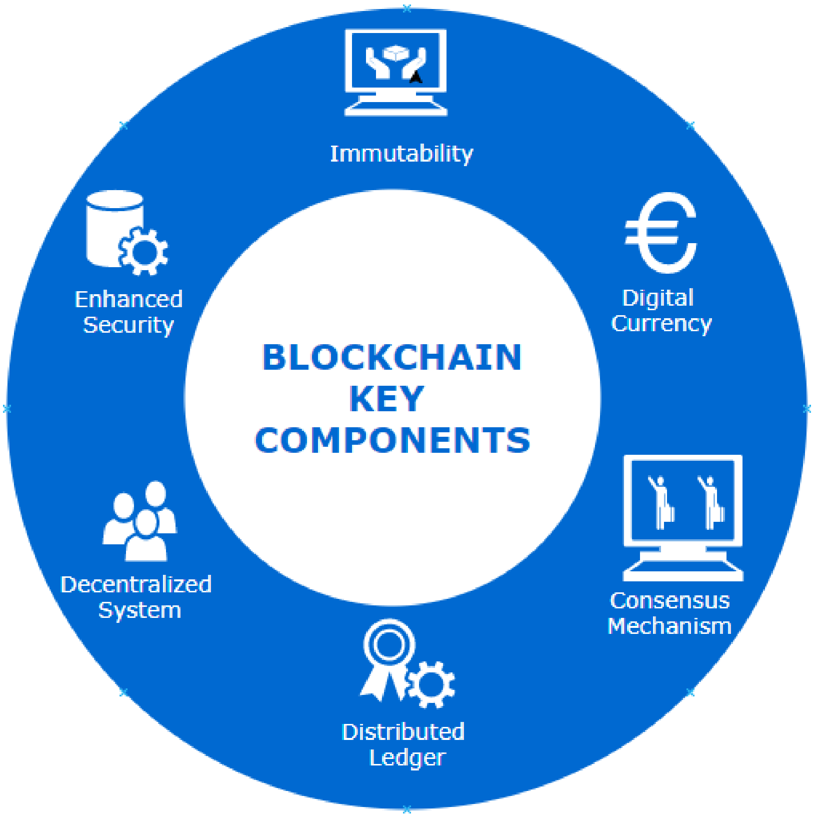

Figure 2.2: Blockchain Components

2.1.4 Core components of Blockchain technology

- **Node** Nodes are communication endpoints, which implies any client or application that needs to collaborate with the Blockchain does that through nodes. These nodes can be any type of device, including a computer, phone, or tablet. There are different types of nodes, and each of them has different functionality. However, there are few things we need to consider that it is not necessary that every device which is connected to a blockchain is not a node and different nodes in the blockchain network carry different functionality. Based on the functionality of the blockchain network, there are different types of nodes such as full node, partial node, supernode etc.

- **Full Node** A full node is basically a computer that will have the entire Blockchain. It is the backbone of the blockchain network. If any node wants to connect with the blockchain network, the full node first verifies the new block. The current size of the blockchain network is 337.35 GB till May 16, 2021. Everyday blockchain network is drastically increasing, and to store the full node; you need to have more than 340 GB space in your computer. So, the key functionalities of full nodes are, storing the entire Blockchain verifying the newly added nodes.

- **Partial Node** A partial node does not contain the whole blockchain network but a part of the network. You can turn your device into a partial node by downloading the part of the Blockchain which requires using SPV (Simplified Payment Verification) mode. It is used for transaction purposes. Mobile phones or tablet computers are some of the most common partial nodes as you can't store the entire blockchain network on your mobile.

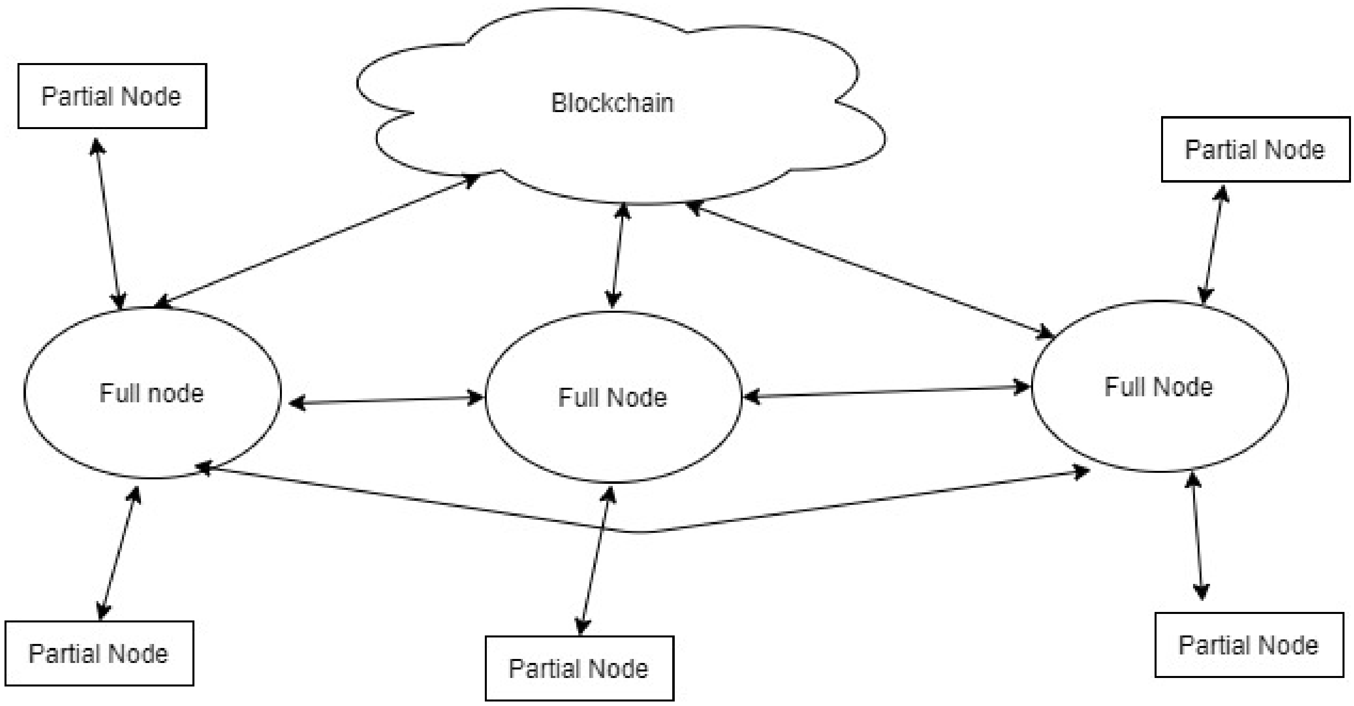

Figure 2.3: Blockchain Nodes.

2.1.5 Consensus Algorithm

In a centralized setup, a single entity controls the network, but as we know, Blockchain is a decentralized setup, then how can we conclude to add an entity or not? Apart from that, how will the whole network[43] understand the current condition of the ledger? The distributed ledger technology solves the problem using certain protocols, commonly known as the consensus algorithm. A consensus algorithm is a predefined group of rules which ensures the security of the Blockchain. It also makes sure the transparency of the network. Though in a blockchain network, nodes have the freedom to leave the network anytime they want. But consensus algorithm functions when it comes to add a new node on the Blockchain. The payout for the miner is determined by the consensus algorithm, which also determines the difficulty of block mining. Furthermore, the protocol also punishes malicious nodes. Now there are many consensus algorithms introduced to operate the blockchain network preciously. Some of the popular algorithms are Proof of Work, Proof of Stake, Proof of Authority, etc.

Various types of Consensus Algorithms

There are different types of consensus algorithms, and each of them has its own advantage and disadvantages. We look into various consensus algorithms like Proof of Work, Delayed Proof of Work, Proof of Weight, Proof of Authority, etc. Among those algorithms, we choose Proof of Work for our blockchain. Now we will discuss why we select Poof of work over other algorithms.

– **Proof of Work** The Proof of work is the most popular consensus algorithm, and it was first implemented in bitcoin though the basic idea is the same. POW is the process of selecting a miner for the next block. Here the miner solves a complex mathematical puzzle and provides a solution. These mathematical puzzles require high computational power to solve. The purpose of POW is to maintain transparency. Miner generates a cryptographic hash of the next block, and once the block is generated, it depends on the majority of the nodes to add or reject the block. But what will happen when more than one miner mines for the same block? In that case, they will generate a different kind of chain which is known as FORK. However, the miner with more computational power can generate a longer chain, and the network approved the longer chain as they are more reliable. Though Proof of work is energy consuming but it is secure and reliable. Besides, in this algorithm, miners get reword for both block and transaction fees. That's why we choose another algorithm.

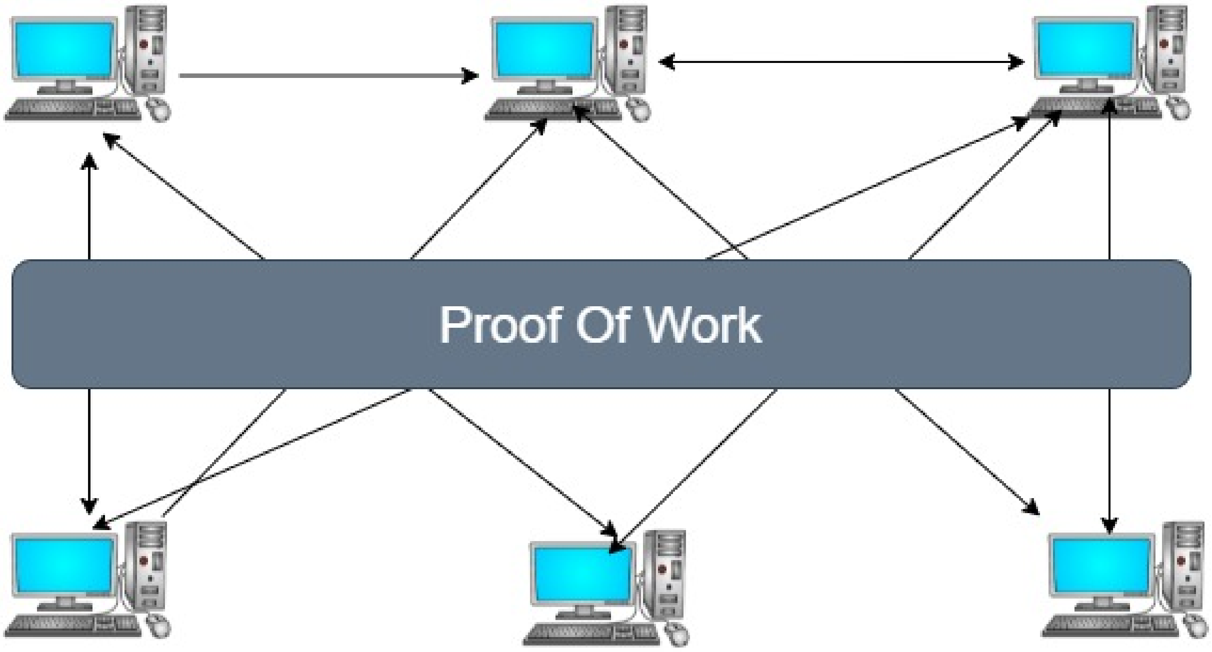

Figure 2.4: Proof of work(POW).

– **Proof of Weight** Proof of weight is an alternate approach to Proof of work. It is energy-efficient, customizable. In this algorithm, your percentage of tokens owned in the network affects on the probability of discovering the next block. However, Incentivization is a challenging part of this algorithm, and because of that, we didn't use this algorithm.

– **Proof of Authority** Proof of Authority is another popular consensus algorithm that required low computational power. Besides, its transaction time is much faster than the other algorithm. However, it is not entirely decentralized, and we notice about the absence of security. As security is our main priority for our Blockchain, that's why we didn't use this algorithm.

2.2 Reinforcement Learning

In this paper, an extensive application of Reinforcement Learning as well as Q-Learning has been applied for price forecasting.
With the booming success of artificial intelligence, data scientists have opted for different machine learning techniques. Reinforcement Learning is one of the areas of machine learning with gradual growing interest. According to [48], Reinforcement Learning (RL) is a subsection of machine learning which is inspired by behaviorist psychology. It includes learning through an agent's own actions and experiences. An agent interacts with a particular stochastic environment. Agents' motives will be to maximizing reward though action. For example, in this paper the agent, which is the service provider, chooses from a set of actions which are the retail prices and sends them to the environments which are customers. When the algorithm returns a feedback, the agent in question receives a reward and the algorithm goes to a new state for the environment.

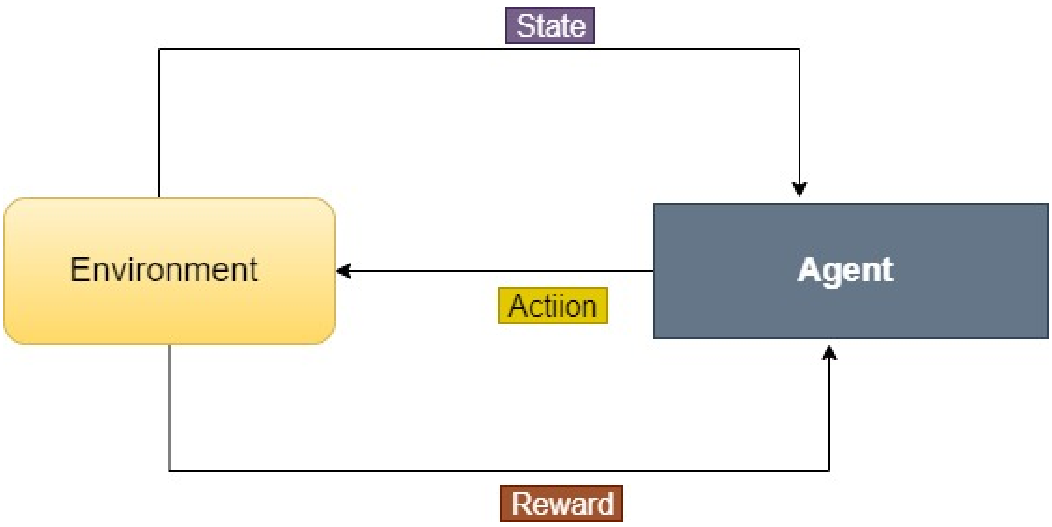

Figure 2.5: Reinforcement Learning.

Reinforcement learning history can be tracked as two independent threads contributing in AI learning until a third thread was included. The major two threads were both rich and prospectus in the 1980s. The first one involved using the trial-and-error strategy in animal learning psychology. This is some of the earliest work of reinforcement learning in AI technology. The latter thread focuses on optimal control and its solution. Value functions and dynamic programming helps generating solution without the implementation of learning technique. Later, a third thread concerning temporal-difference methods were also included. The optimal control thread helped to minimize any parameter of a dynamic system. Richard Bellman approached this problem using the dynamic system's state and value function (also known as optimal return function); finally deriving the Bellman equation. This equation later had a discrete stochastic form for optimal control problem known as Markov decision process (MDP). The dynamic programming implemented by Bellman was far more efficient though it required more computing power. Improvement on existing dynamic programming strategy kept going on from the 1950s. Some notable improvements include extensions to partial MDP ((surveyed by Lovejoy, 1991), approximation methods (surveyed by Rust,1996) and asynchronous methods (Bertsekas, 1982, 1983). These researches improved upon the optimal quality control over time [46].
The most major thread that consumed the modern-day reinforcement learning is the trial-and-error learning. This term was introduced in psychology first where "reinforcement" ways of learning a particular environment was natural. Edward Thorndike explained the trial-and-error method using animal context where he defines that animal will, followed by satisfaction is deeply connected to the environment and the recurrence will more likely to occur. Same goes for the discomfort and dissatisfaction level as well. Thorndike referred this phenomenon as "Law of Effect" that states that the effect of recurrent event tend to have relation with the selection of events. The Law of Effect inherited two important features- one being Selectional and other one being Associative. Selectional means that a wide array of selection is possible for actions and actions will be selected based on consequences. Associative means that the alternatives are related to particular situations. Thus, we can state that Law of Effect work as

a "search and memory" process where both searching amongst many actions in each situation for desirable result and memory being keeping log to remember the beast action for future purposes. Combining these two factors are the foundation of reinforcement learning. The trial-and-error method caught the interest of many researchers like Farley and Clark. Minsky particularly mentioned about the credit assignment problem using reinforcement learning in his book "Step Toward Artificial Intelligence" (Minsky, 1961) [9], [46].
Other works were followed shortly by researchers Rosenblatt (1962) and Widrow and Hoff (1960) where their work on language of rewards and punishments was motivated by reinforcement learning. One of the next works on reinforcement learning was from John Andrae (1963) who developed STeLLA system which interacts with its environment and learn through trial-and-error. Donald Michie described a system which will learn to play tic-tac-toe game called MENACE using trial-and-error technique. Learning automata improvised reinforcement learning as those are low memory machines for solving computational problem. Klopf combined the trial-and-error with the temporal-difference learning resulting more learning efficiency in large scale system. Sutton further incorporated animal learning theories based on Klopf's ideas [9].

In the papers [6], [8], [15], [17], [19], [31], Energy scheduling was used using RL algorithm which helped to provide efficient charge or discharge policy. As this required a small amount of space and actions, it was done effortlessly. More examples can be seen as RL helped maximizing profit by using it as a tool to choose a strategy for buying or selling energy-trading. Thus, having a unique "model-free" feature which can be applied in any dynamic stochastic environment, RL has huge potential to be used in energy trading smart grid system where dynamic pricing and electricity consumption needs to be set effortlessly.

Chapter 3

Related Work

Smart Grid is a new system of power distribution, and there are already numerous researches on the subject. However, with new technology comes quite a number of shortcomings. Price forecasting and the accuracy of price forecasting is closely related to how the demand and supply will turn out. In one such paper, the author proposed a Multi-Layered Neural Network for price forecasting [26]. However, with such a complex network, we get a high computational time; moreover, the loss of neurons is also very high [41].

There is another paper where price forecasting is done using a Hybrid Structured Deep Neural Network; however, this too has a high computational time [32]. On the other hand, Long Short-Term Memory and Recurrent Neural Network can also be used to determine the accuracy of price forecasting [34]. In this case, the overfitting problem will increase; meaning that the result will show low bias with high variance.

On the other hand, there are price forecasting methods using Deep Learning techniques. One such paper has Deep Neural Networks with a hybrid Long Short-Term Memory and Deep Neural Networks structure to greatly improve the accuracy of prediction. However, the paper is only compared using only a single dataset. Therefore, it is not suitable to use such a paper for real life experiments and appliances since there are so many factors to look out from [33].

In this paper [36], the researcher proposed an architecture where all the components active in the smart grid system are considered a node and these nodes communicate in a blockchain network with each other. Using Ethereum Blockchain, a decentralized open-source cryptocurrency network that supports smart contracts, a proper transaction method is introduced. The blockchain system runs on Proof-of-Work (PoW) consensus mechanism that confirms the transactions between consumers and producers and adds new blocks to the chain using mathematical hash equations. This is done through data mining, and a considerable amount of computer power is needed to ensure the complete transaction. The producer node usually generates a token with its corresponding address value and sells electricity to the consumers using that particular token by placing it in the market system. Consumers buy that token with the same corresponding value, and the system checks if the consumer has enough balance. A smart contract is created and added to the blockchain that keeps the transaction history immutable. After the electricity is consumed, the token is burned. The consumers can also sell their tokens to other consumers as they are transferable, which can solve other consumers' demands. A smart contract acts as a trusted escrow service and ensures transparency between consumers and producers. The framework is also pro-

vided with a client-based application to access its smart contracts using the API. The issue regarding this system is that only up to 15 transactions per second can be handled by the Ethereum blockchain, and the currency rate is not constant all the time. Using the Proof-of-Importance mechanism that is based on the overall contribution of the node to the system, an enhanced and improved framework can be implemented.
Here [21], the author proposed a price-based novel demand-response (DR) model to enhance the electricity management system between the utility company and the consumers. Using a Pricing Function, the real-time price (RTP) is manipulated, and the balance of supply and demand is obtained. Stackelberg game model is used to flatten the system's aggregated loads while maintaining the utility company's benefit and the user's cost minimization. An iterative algorithm is offered between the utility company and consumers to derive the Stackelberg equilibrium, through which the optimal power generation is measured. Numerical and graphical results confirm flattening peak demands and fill the vacancy of valley demands using the Utility Company Model and User Model. This proposed model achieved the lowest Peak-to-Average Ratio (PAR) and highest Load Factor (LF), which are advantageous for the utility company in balancing loads in the power system. Although the system could not provide an optical communication network, and transactions were not secured or recorded. Moreover, lower computational time can be obtained using different models.
On the paper [23], researchers have talked about the security threats, challenges and solutions in fixing the threats regarding smart grid. Such cyber-physical attacks can harm the integrity of the grid system by Data Injection Attacks (DIA) and Time synchronization Attacks (TSA). DIAs consist of an adversary manipulating exchanged data such as sensor readings, feedback control signals, and electricity price signals. It can be done by compromising the state estimator which enables complete monitoring of the power and current flows throughout the grid. TSA mainly attacks the phasor measurement units (PMU) which are high-speed measurement units capable of measuring the voltage and current phasors as well as local frequencies. Distant measurement devices are spread worldwide and are subject to transmission delay. That is why time synchronization is essential as time referencing provides a time stamp to each collected measurement based on their GPS location. An adversary using TSA can manipulate the time reference of the time stamped measured phasors to create a false visualization of the actual system conditions thus yielding inaccurate control and protection actions. Using TSAs, the GPS signal is spoofed and counterfeited by the attacker so that PMU sampling is done at the wrong time hence generating measurements with wrong time stamps. It might result in disconnection, cascading of electric lines and often blackout. Researchers opted for Scanning and Detection techniques to counter these security threats. provides a methodology for detecting stealthy data injection attacks (DIA) targeting the state estimator. Further proposed was a detection mechanism against TSAs targeting PMUs. On another paper [37], we can find the necessity of smart grids integrating in smart homes. Smart meters are implemented as a smart grid interfacing between consumers and service providers. These smart meters operate on EMS making these more energy efficient. This smart meter can be used in home appliances and the grid network makes a two-way communication between grid and customers. In this paper, we can also find the concept of a self-healing power distribution system. A self-healing grid makes use of digital technology and components and real time secure communications automated technologies. The self-healing grid removes problems as soon as they are detected so that the continuity in the network

is maintained. The whole process of detection of fault, removing it and restoring the network is self-healing action of the grid. Furthermore, distribution intelligence in an effective smart grid system is mentioned. It refers to the distribution of power with the components such as transformers, feeders, isolators, circuit breakers which help in outage management. In smart grid the real time outage management is priority so to achieve real time monitoring the components carrying power to the consumers home should be smart enough for monitoring the outage management.

In this paper [1], Blouin and Serrano suggested peer-to-peer agreements LEM with a decentralized, anonymous purchaser, and seller matching. LEM (local energy market) is developing as a possible option for organizing an increasingly dynamic decentralized energy infrastructure. LEM structure allows agents the chance of virtually trade energy within their community. It not only reduces the cost of energy but also plays a significant role in the local economy. In their proposed system, trading was conducted bilaterally performed between directly affected agents. Their local energy market uses auction formats that include buying (bid) and selling orders for energy to a (public) order book. These bids are monitored continuously, and no new entrants are allowed once the market is open. There is also a possibility where a large amount of bids goes untraded. However, when you are combining a new market model, it is crucial to increase public acceptance. Auction involves a centralized marketplace, but they can be run decentralized on a distributed economic system through a blockchain.

This paper [16] discusses electrical forecasting, which is getting more and more popular due to the deregulation and integration of renewable resources. Because of the limited power source, it is crucial to predict the power prediction accurately. So, this paper proposes a short-term load predictor that can forecast for the next 24 hours of loading for predicting power consumption. To detect the most affecting past samples on potential loads, they used an autocorrelation plot to determine the similarities between the signal and its lagged versions. Therefore, it constructs a model for each hour of the day rather than using one model for a day. The load values of the previous two days simultaneously are used as the indicator inputs and the morning and evening peaks of the last day. They used ANN (artificial neural networks) and SVM (support vector machines) technology in their project. Tunisian Power Company tests its proposed system to produce accurate and acceptable results one day in advance, with an average error rarely exceeding 2.3 percent. However, their computational time is very high.

Chapter 4

Proposed Blockchain-based model for electricity trading in Smart Grid

4.1 Smart Contract

A smart contract is a bunch of rules and agreements approved by both parties. Once you have formulated a smart contract, you can verify this computer protocol and upload it if you want. It offers authenticity to verify the effectiveness of any operation or action. After deploying the smart contract, some functions and events are performed to confirm the transaction. It can assist us in transferring money, transferring assets and shares, and other significant transactions in a transparent manner.

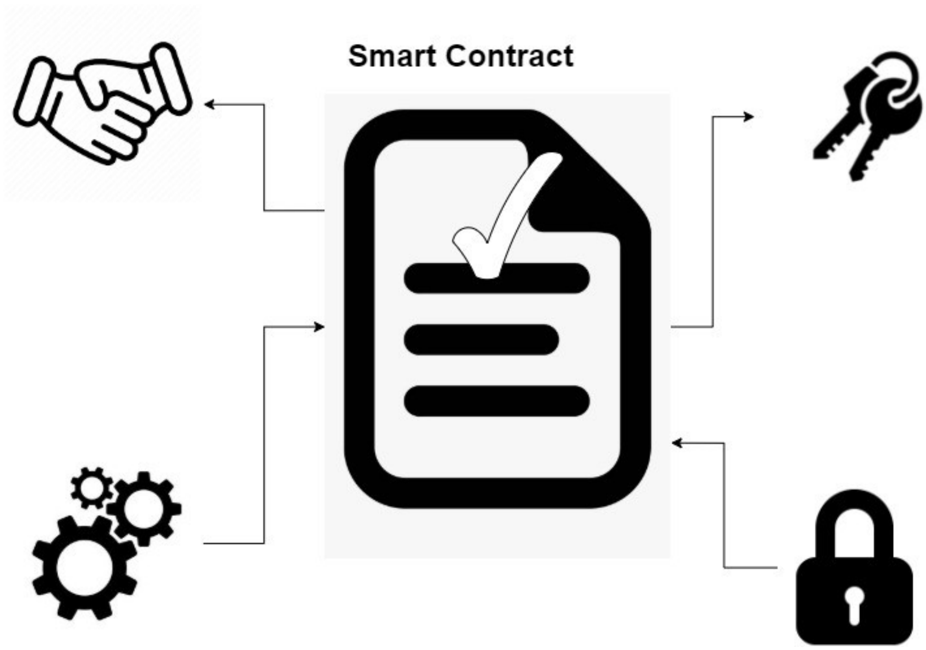

Figure 4.1: Smart Contract.

4.1.1 How Smart Contract works

Smart contract plays as a third party in blockchain and involves dealing and ensuring trust among buyer and seller. It's an agreement between buyer and seller with multiple stored conditions in the blockchain and can not be manipulated. Nodes request the primary node to deploy the contract using the parameters, and after that, the nodes and the primary node hold the updated smart contract. It contains three main mechanisms: the parties' contractual agreements, the administration of predetermined conditions essential for the contractual responsibilities to be fulfilled, and the deployment of the smart contract[25]. As, it is a fully decentralized system that does not require any additional party to conduct it, it is immutable, secure, and distributed among all the nodes.

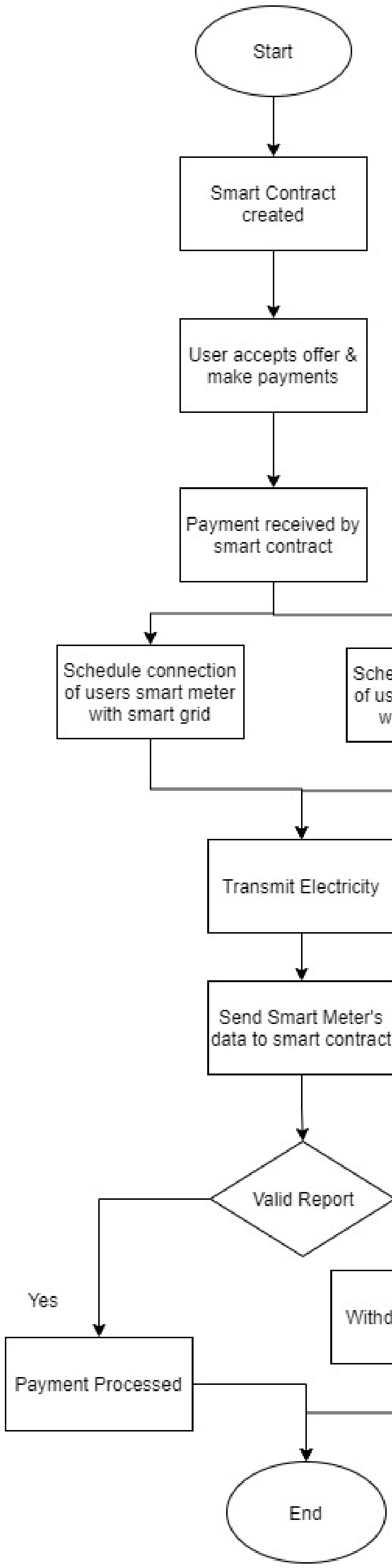

Figure 4.2: The flow diagram of blockchain-based smart grid system design

4.2 Key in Cryptography

4.2.1 Private Key

Private Key is a crucial piece of equipment in the world of cryptography [51]. It allows someone to show ownership of your public address and spend the cash linked with it. QR code, 256 characters long binary code, 64 digit hexadecimal code, or Mnemonic phrase are some of the forms of a private key. A private key, in whatever form, is an astronomically large number, and it has its own reasons behind it. One of the features of private is, you can generate a public key by using your private key, but you can't use the opposite.

4.2.2 Public Key

Anyone can receive cryptocurrency transactions using a public key. [35] It is based on the "Trapdoor Functions" mathematical primitive, which is a math problem that is easy to solve in one way but nearly hard to reverse. [51] A private key is linked with a cryptographic code, and anyone can submit a transaction with the public key if they have the correspondent private key. It proved you the owner of the cryptocurrency received in the transaction.

4.3 User Layer

The user layer comprises of all the users who purchase electricity from the market for their routine work. No central party is needed for communication between prosumers and consumers, rather the communication is conducted directly. Blockchain can be used to share information between prosumers and consumers. Information can consist of selling a certain amount of electricity or buying a certain amount of energy from prosumers. A small community of registered users are able to do trading. The registered and authorized users can access information regarding available electricity sell and buy. A solar panel-enabled home acts as a prosumer for electricity generation. If, after its own use, a home has surplus energy, it may sell this surplus energy to those homes that are energy deficient. On the other hand, homes purchasing electricity to fulfilll electricity demand are consumers. Users provide information to the information, connected with user layer, to get registered in the blockchain network.

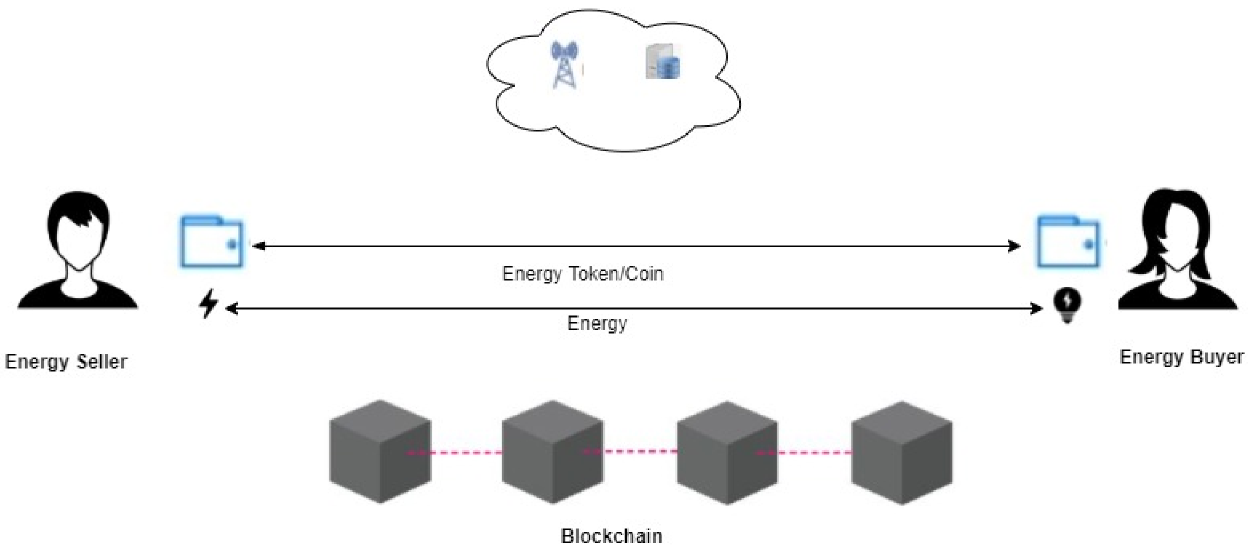

Figure 4.3: Proposed Blockchain Architecture.

4.4 Information Layer

The information layer consists of records and ownership of energy. A buyer or seller must register himself in the smart grid system by providing important information such as name, address. Upon joining the system, a user receives a private ID and profile that shows users' information which is stored in the form of hash in a block of the blockchain. After authenticating, users can view their electricity usage history. It helps to monitor user activity reliably and SG stores all information of users at a decentralized system, stores data in encrypted form. Based on the buying information, one can understand how much electricity is needed and if any of the energy is being wasted without proper usage. These information is stored in the blockchain and can be manipulated with the help of smart contract which can perform functions if proper parameters are provided. The measurement calculated by the SM can effectively convey whether a user has surplus energy or not. This energy surplus and deficiency is also stored in a blockchain. Thus, every user in the smart grid network has all the transaction history data.

4.5 Users Authorization

Smart contract allows users to become a part of the network and check the previous electricity trading history. The users need to register for Authorization and authentication, which obviously increase the security issue. For registration into smart contract, the initial step is being verified seller or buyer. After the authentication, users can become a part of the electricity trading and can pick trading time and price. When all transactions are completed, a block is created. In any transactions, hashes are calculated to verify the original owner address. However, even after trading, no one can check others name, address, or anything and only address of the account is just visible. Without proper Authorization of the transactions, the money will not be transferred and the trading request will not be accepted thus the transaction will not go through.

When the transactions are done without any issues, the seller sends the amount of energy bought by the buyer using the smart grid. This information is stored in the blockchain and the smart contract must be called when the transactions are made.

Chapter 5

Proposed Q-learning method for Smart Grid price forecasting

5.1 Problem Formulation

5.1.1 Customer Model

There are two types of energy load profile that a customer can have. One is critical load and the other being the curtailable load. These are classified based on their energy requirement and priorities [30].
Critical load is the load that is in high priority and must be met at any cost for customer satisfaction and priority. An example can be use of power in data centers and power station. The equation that satisfies the critical load demand is denoted as follows:

$$ec_{t,c}^{\text{critic}} = ed_{t,c}^{\text{critic}} \tag{5.1}$$

Where t is divided into 24 hour segments which represents each hours of the day. The price will be updated every 24 hours.c $\in \{1,2,3\ldots C\}$ represents the customers c. $ed_{t,c}$ refers to the energy demand, and $ec_{t,c}$ refers to the energy consumption of a customer c at a specific time t.

Curtailable load, on the other hand, is more flexible with price. The demand of customers for curtailable load can change with price as the demand falls with the increase of electricity price. For a certain customer CU c, consuming a certain amount of $ec_{t,c}$ at time t will correspond to the customers' that amount of load satisfaction. Subtracting that load satisfaction from the total amount of energy demand gives us the dissatisfaction level at time t which is $ed_{t,c}$- $ec_{t,c}$. This dissatisfaction level is denoted as $\phi_{\text{t,c}}$. This signifies the degree of dissatisfaction that customers can experience when prompted to reduce their electricity demand due to high prices. This co-efficient is convex in nature and tend to increase massively if energy reduces significantly [27]. The equation that satisfies the critical load demand is denoted as follows:

$$\begin{aligned}
&ec_{t,c}^{curt} = ed_{t,c}^{curt} \cdot \left(1 + \xi_t \cdot \frac{\lambda_{t,c} - \pi_t}{\pi_t}\right) \\
&\xi_t < 0 \\
&\lambda_{l,c} \geqslant \pi_t
\end{aligned}$$

Here $\xi_t, \lambda_{t,c}, \pi_t$ denotes elasticity coefficient, retail price for customer c and wholesale price at time t, respectively.
In economic terms, Elasticity ξ_t measures responsiveness of the change between two variables with respect to one another. Price elasticity of demand can show how the change in product price can impact on the energy demanded of a particular good. In the context of smart grids, this elasticity refers to the change in the demand of electricity with the 1% increase in price of that particular time. Thus, this elasticity between the demand and price is inversely proportional. Researches regarding elasticity on smart grid energy conclude that the demand for electricity is most elastic during peak hours and long-run elasticity results better compared to the short one [30]. The elasticity values used in this paper to study the RL implementation are acquired from published papers.
Dissatisfaction cost function can be defined as follows:

$$\begin{aligned}
&\varphi_{t,c} = \frac{\alpha_c}{2}\left(ed_{t,c}^{curt} - ec_{t,c}^{curt}\right)^2 + \beta_c\left(ed_{t,c}^{curt} - ec_{t,c}^{curt}\right)\\
&\alpha_c > 0\\
&\beta_c > 0\\
&D_{\min} < ed_{t,c}^{cur} - ec_{t,c}^{\text{curt}} < D_{\max}
\end{aligned}$$

αn and βn are parameters varying dependent on customer to customer. The former is the response to a customer's consumable energy reduction where a higher value denotes that a customer is likely to get more dissatisfied if the electricity prices get lower. The latter parameter is predetermined constant. $\mathrm{D}_{\min}$ and $\mathrm{D}_{\max}$ are the lowest and highest energy reduction respectively [22].
So, the minimized cost of a CU n can be described as follows:

$$\min \mathscr{P}_{t=1}^{T} \left[\lambda_{t,c} \cdot \left(ec_{t,c}^{\text{curt}} + ec_{t,c}^{\text{critic}}\right) + \varphi_{t,c}\right]$$

Where both the cost of a customer n to buy electricity and the dissatisfaction efficient from demand reduction are used.

5.1.2 Service Provider Model

Buying electricity at wholesale prices from the GO and later selling that electricity at a retail price to CUs, SPs need to ensure maximum profit. The dynamic pricing can be denoted as follows:

$$\begin{aligned}
&\max \mathscr{P}_{c=1}^{N} \mathscr{P}_{t=1}^{T} \left(\lambda_{t,c} - \pi_t\right) \cdot \left(ec_{t,c}^{\text{curt}} + ec_{t,c}^{\text{critic}}\right)\\
&\kappa_1 \pi_{t,\min} \leqslant \lambda_{t,n} \leqslant x_2 \pi_{t,\max}
\end{aligned}$$

κ_1 and κ_2 are predetermined. These are the parameters that keeps the price fair for both the SP and the CUs, and they are the retail price bound coefficients.

5.1.3 Objective function

Both the SP and CU benefit function can be denoted as follows:

$$\max \mathscr{P}_{c=1}^{N} \mathscr{P}_{t=1}^{T} \left[\rho \cdot \left(\lambda_{t,c} - \pi_t\right) \cdot ec_{t,c} - (1-\rho) \cdot \left(\lambda_{t,c} \cdot ec_{t,c} + \varphi_{t,c}\right)\right]$$
$$ec_{t,c} = ec_{t,c}^{\text{curt}} + ec_{t,c}^{\text{critic}}$$

Here the value is $\rho \in [0,1]$ is significant as it is the relative importance between SP's profit and CU's costs.

5.2 Q learning

This paper features a hierarchical electricity market consisting of the grid operator (GO), service provider (SP) and customer (CU). Service provider monitors the energy demand and dissatisfaction level of customers. Monitored data is then reflected by adjusting the dynamic pricing strategy. Wholesale electricity prices provided by grid operators are also monitored for energy consumption. In this system, the service provider works as an agent who determines a retail price to serve the customers. Here, customers serve as the environment. Time is segmented in an hourly basis and each time slot reward is generated from customers as electric bill. Customer's energy demand and dissatisfactory factors will influence the stated of dynamic pricing and it will be determined with the help of Q-Learning.

Here, implementation of Q-Learning can give significant advantages. As RL is model free, so it enables to determine price actions without any model environment. The trial-and-error process of RL comes into play as customers and service providers dynamically set up the price and profit. Secondly, adaptability of Q-Learning is a key factor. The electricity market is changing massively day by day with the demand-supply, price factor and other factors like customer dissatisfaction. Q-learning is adaptive to cope up with the changes through its ongoing learning process. Thus, the flexibility of the dynamic energy market is kept on check.

The proposed system will support a pricing strategy which will be applied in a hierarchical electricity market. The hierarchical framework of this pricing is formulated with RL as Markov decision process (MDP). Finally, this decision making of dynamic retail pricing is solved with Q-Learning. Being model free, the system will learn gradually about the flexibility and uncertainties of the demand change and system requirements over time through on-line learning process. Customer's dissatisfaction levels are also taken as a key factor to change customers' demand usage through the on-Line learning process.

Algorithm 1 Q-learning Algorithm

S = State, A = Action

Initialize

Initialize Q(S, A) using arbitrary values

for *each iteration i,* **do**

At time interval t Choose an action $\lambda_{t,c}$ and execute for state $\left(ed_{t,c}\right)$

Observe the reward $r\left(ec_{t,c}|ed_{t,c},\lambda_{t,c}\right)$ and the new state $\left(ed_{t+1,c}\right)$

$$Q\left(ec_{t,c}|ed_{t,c},\lambda_{t,c}\right) \leftarrow Q\left(ec_{t,c}|ed_{t,c}\lambda_{t,c}\right)+\theta\cdot\left[r\left(ec_{t,c}|ed_{t,c}\lambda_{t,c}\right)+\gamma\cdot\max Q\left(ed_{t+1,c}|ed_{t+1,c},\lambda_{t+1,c}\right)\right.$$
$$\left.-Q\left(ed_{t,c}|ed_{t,c}\lambda_{t,c}\right)\right]$$

end

5.3 Reinforcement learning methodology

As previously mentioned, the proposed system can be structured in a RL method where service providers will serve as an agent, customers will serve as the environment, the action will be the retail price that the customers are provided by the service providers, state being represented by the energy demand, consumption and dissatisfaction levels and finally both the SPs' profit as well as CUs' minimized cost can be seen as reward. Here, Markov decision Process (MDP) is used to formulate retail pricing and then using Q-Learning, the dynamic pricing algorithm is formulated.

5.3.1 Formulating system model to Markov decision process

As the dynamic electricity market has a stochastic environment, MDP is beneficial for reforming the system model. Only the current time slot will be considered for reward and energy consumption so no historical data will have any impact to maintain the stochastic feature of the environment. MDP includes these major four components:

(1) t defines the time interval for the actions that represent retail price. It has to be discrete.

(2) $\lambda_{t,c}$ is the retail price chosen at time t for CUc.

(3) $\mathrm{ed}_{\mathrm{t,\,c}}$ represents a CUs energy demand before getting notified of the retail price from SP. $\mathrm{ec}_{\mathrm{t,\,is\,the}}$ consumption that occurs after the price signal.

(4) $\mathrm{r}\left(\mathrm{ec}_{\mathrm{t,c}}|\mathrm{ed}_{\mathrm{t,c}},\lambda_{\mathrm{t,c}}\right)$ is the reward that defines a minimal cost of CUc and SP's maximum profit at time t.

Thus, for one episode the reward will be

$$R = r\left(ec_{1,n}|ed_{1,c},\lambda_{1,c}\right) + r\left(ec_{2,c}|ed_{2,n},\lambda_{2,c}\right) + \cdots + r\left(ec_{T,c}|ed_{T,c},\lambda_{T,c}\right)$$
$$r\left(ec_{t,c}|ed_{t,c},\lambda_{t,c}\right) = \mathscr{P}_{c=1}^{C}\left[\rho\cdot\left(\lambda_{t,c}-\pi_t\right)\cdot ec_{t,c} - (1-\rho)\cdot\left(\lambda_{t,c}\cdot ec_{t,c}+\varphi_{t,c}\right)\right]$$

The total future reward will be

$$R_t = r\left(ec_{t,c}|ed_{t,c},\lambda_{t,c}\right) + r\left(ec_{t+1,c}|ed_{t+1,c},\lambda_{t+1,c}\right) + \cdots + r\left(ec_{T,c}|ed_{T,c},\lambda_{T,c}\right)$$

As the environment is stochastic, the rewards for the same actions can also diverge significantly. So, a discounted future reward is used.

$$\begin{aligned} R_t =& r\left(ec_{t,c}|ed_{t,c}\lambda_{t,c}\right) + \gamma\cdot r\left(ec_{t+1,c}|ed_{t+1,c},\lambda_{t+1,c}\right) + \gamma^2\cdot r\left(ec_{t+2,c}\left|ed_{t+2,c}\right|\lambda_{t+2,c}\right) \\ &+\cdots+\gamma^{T-t}\cdot r\left(ec_{T,c}|ed_{T,c},\lambda_{T,c}\right)\end{aligned}$$

where $\lambda\in[0,1]$ is discount factor that compares future reward with current reward system. A value of 1 for means that the same action implemented on the environment will result in the same reward each time, resulting in a deterministic environment. The pricing strategy that maps current states to action will be $\upsilon:\lambda_{t,n}=\upsilon\left(ed_{t,c}\right)$. With the help of Q-Learning, optimal policy will be determined to maximize reward.

5.3.2 Using Q-learning for dynamic pricing problem

Q-Learning is a subsection of RL which is model-free. It can be used to get optimal policy which is the dynamic policy referred in this paper. The Q-learning algorithm is as follows:

The main Q-learning process involves a Q-value $Q(ec_{t,c}|ed_{t,c},\lambda_{t,c})$ which is assigned to every state-action pair at a time slot t. Then it is updated over each episode that promotes good behavior. $Q^*\left(ec_{t,c}|ed_{t,c},\lambda_{t,c}\right)$ refers maximum discounted reward for future when taking t,c action. This optiomal Q value can be refered as below:

$$Q^*\left(ec_{t,c}|ed_{t,c},\lambda_{t,c}\right) = r\left(ec_{t,c}|ed_{t,c},\lambda_{t,c}\right) + \gamma \cdot \max Q\left(ec_{t+1,c}|ed_{t+1,c},\lambda_{t+1,c}\right)$$

According to Table 1 [126], maximum Q value is calculated. If the learning rate factor is 0, then the agent learns nothing from the existing data in this case it is the SP who gains no new information regarding optimal policy. On the other hand, learning factor of 1 will make the agent reevaluate the recent state of the environment, in this case it is considering the recent demand response of the customers.

In the dynamic electricity market, service provider interacts with the customer through their dynamic pricing. Then the demand of CUs change overtime and the SPs receive a new state. The trial and error of these set of actions which are the dynamic pricing implemented by the SPs, generated Q-values are stored and updated time to time. Eventually the values are going to converge at a maximum value. If the maximum expected profit of SPs with action $\lambda_{\mathrm{t,c}}$ at a demand $ed_{t,c}$ the optimal policy can be referred as:

$$C = \operatorname{argmax} Q\left(ec_{t_u D}|ed_{t_t,0},\lambda_{t,0}\right)$$

Below is the flowchart of how the algorithm works:

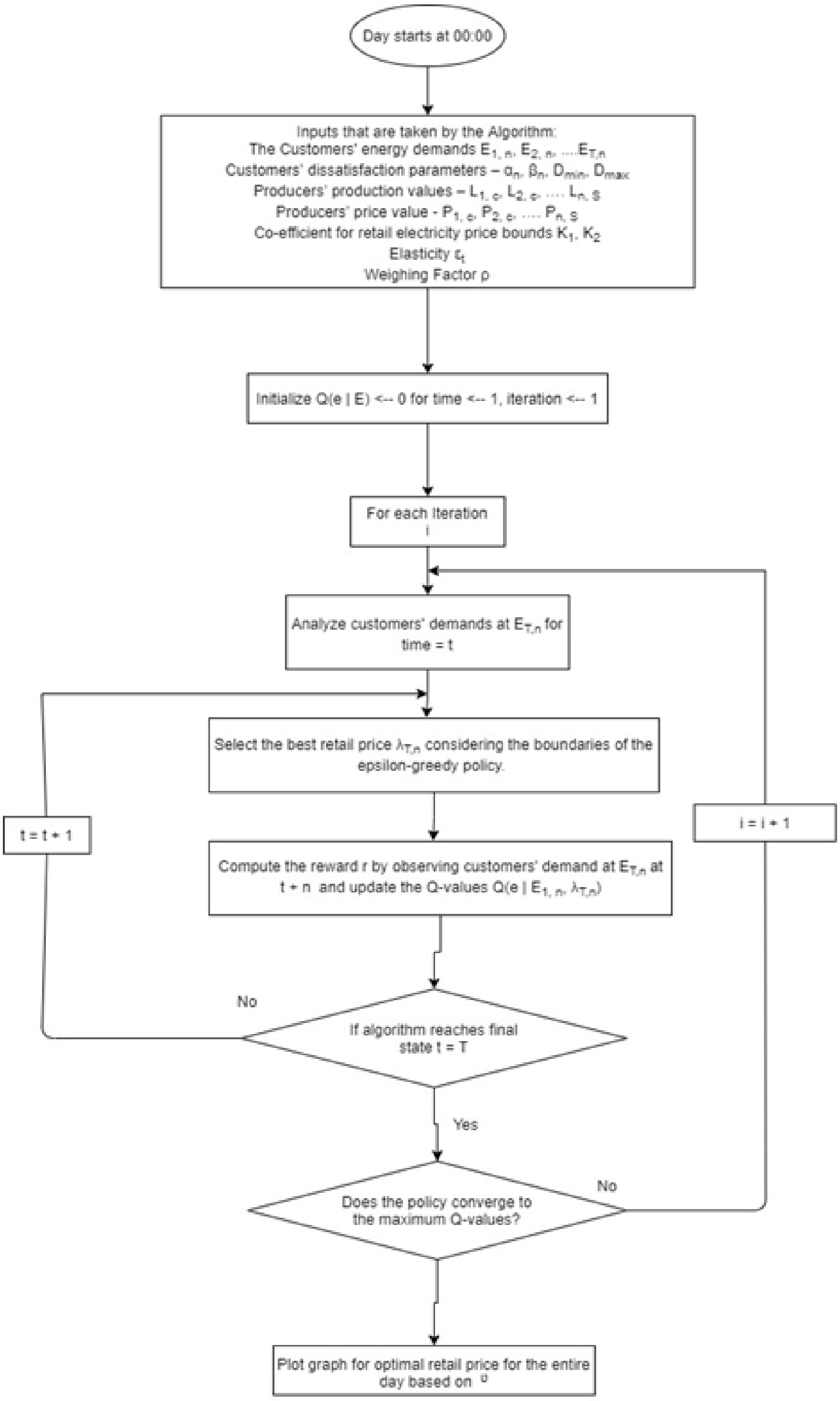

Figure 5.1: Flowchart for implementing the Q-learning process for figuring out the optimal price.

Since RL excels at making sequential decisions in an unknown environment, it can adapt to the policy that is required in real-time and learning from past experiences. Therefore, here, Q-learning is the best RL-method to find the optimal pricing strategy. Here, the flowchart shows that the algorithm begins at 00:00 and ends after 24 hours. The inputs that are taken are the prices from producers following the time slot T in hourly fashion. There are also the coefficients of the price bounds from the third-party providers, and all other parameters in the flowchart shown above. After the algorithm runs the inputs, it initializes the Q-value $Q(ec|ed)$ to 0, the time to the beginning of the day. The algorithm then finds the optimal prices at each hour of the day using epsilon-greedy policy, abiding by the price bounds. To be efficient, the epsilon-greedy policy selects an action with uniform distribution from a set of available actions.
Using this policy, a random action with a probability that is between 0 and 1 from the 1-values from each state while iterating. In short, the agent here is randomly selecting a retail price at each state and scans them amongst the already stored values from the previous states to select the highest value and then choose the retail price.
After the SP chooses the retail price, it will receive rewards accordingly from the reward equation. In the same time, the SP also observes the customers demands for the following time slot and updates the Q-values using the Q-learning process. Lastly, if the Q-value does not converge to the maximum Q-value, the system then moves on to the next iteration until it finally does converge. This is the termination rule, where the difference between the present and previous Q-value is less than δ. After converging, the SP will then get the final optimal retail prices for the hourly slots of the day.

Chapter 6

Implementation and Analysis

6.1 Implementation of Blockchain

In this section, an Ethereum based private blockchain system is introduced that ensures a safe and secure electricity marketplace between the CU and SP. Ethereum provides an open-source blockchain platform that runs on the basis of smart contracts, and the currency it runs with is 'Ether'. It is mainly a DPL that can record, verify and deploy transactions that occur within the network. In order to create business, financial, and entertainment applications, the Ethereum network is an excellent choice as it provides complete immutability and security without any disruption. These applications are known as 'DApp' [50] and are very common in the market industry nowadays. The application users pay a particular fee that is 'Gas' to run the functions, and it depends on the amount of computational power needed to compute the whole process. Smart contracts stored on the blockchain interact with DApps and integrate the user interface with the smart contract that stores data to the blockchain on the back end in different batches or blocks. It usually works on the 'Proof of Work' consensus mechanism where all the connected computers are called 'nodes', and they use their computational resource to add information of transactions to the blocks. There is a canonical computer in the Ethereum technology(EVM) whose state is followed by the other nodes. When a command is run in a particular node, every other node must also verify and execute the same command and make a newer version of the block in the blockchain. It brings a change in the state of the EVM where smart contracts are stored. Users call out programs into the EVM storage using particular parameters to perform actions.

6.2 Experimental Setup

6.2.1 Blockchain Network

The core network of the system is Ganache which is a personal or private blockchain for DApp development. [49] Its primary purpose is to design, launch, and test the DApp securely and predictably without costing any real Ether. It provides a local dummy server that holds ten(10) accounts containing a balance of fake 100 Ether to perform the transactions. Each account holds a public key and a private key that is used to conduct the transactions. Using these wallets, we can deploy, test, perform actions. While doing the transactions, the 'Gas' amount is given, which is a core part of the transaction as it requires Ether to run the transactions, and provided amount of Gas is limited. The server accepts only RPC connection on host HTTP://127.0.0.1: 7545. Here 7545 is the port number through which the blockchain can connect with the smart contract. In figure 6.1, 10 fake accounts are displayed where each has a balance of 100 Ether. And using the private key showed in figure 6.2, we can connect that particular account with the browser to do transactions.

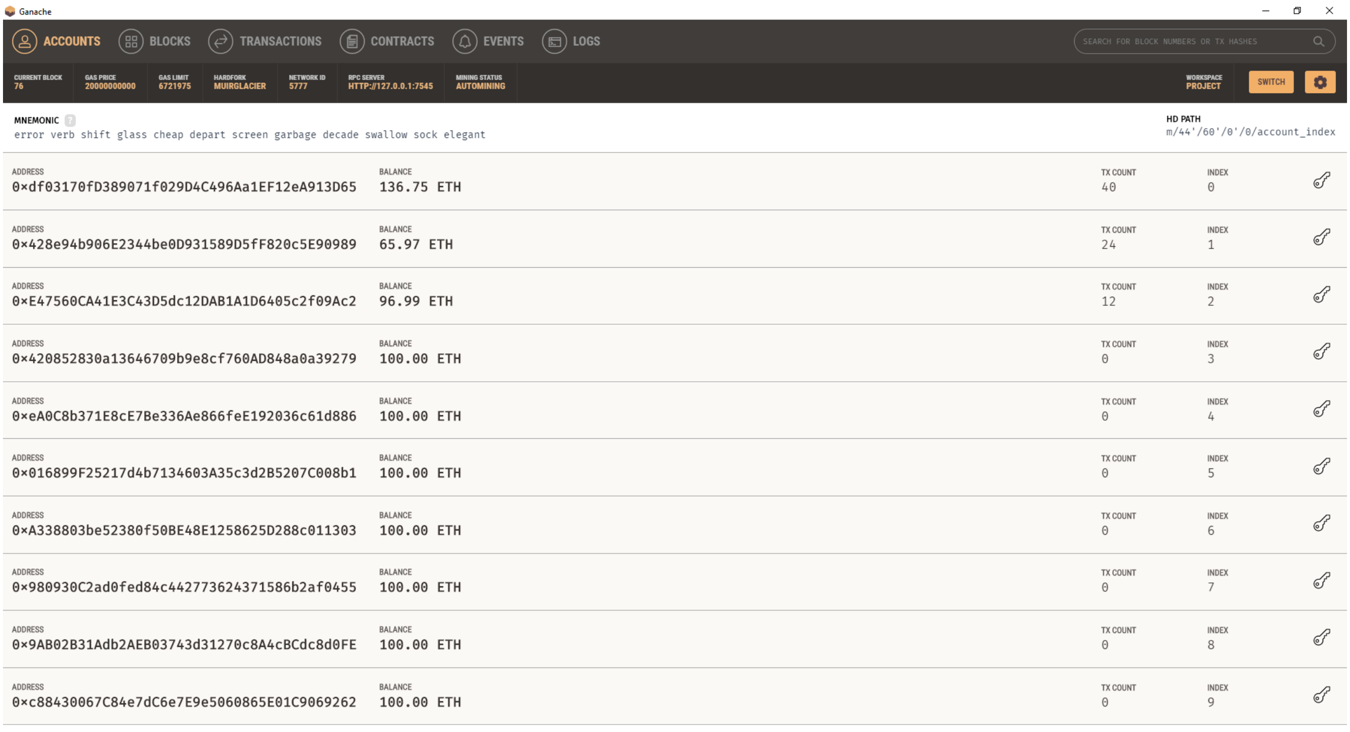

Figure 6.1: Ganache UI Blockchain Server.

Figure 6.2: Account Address and Private Key.

6.2.2 Smart Contract Deploy

Solidity language is used to write the smart contract is created, and the version of the Solidity language is >=0.4.21 <0.6.0. It holds the main functions and events of the whole system. As the smart contract contains the main structure of the performed actions, we must provide particular requirements to correct and realistic the supplied parameters. After deploying the smart contract, it will hold an address of its own that must be provided to the blockchain while connecting them. Usually, smart contracts have a balance and can send transactions across the network. However, they are not controlled by a user; instead, they are deployed to the network and run according to a set of instructions. The smart contract is written in Visual Studio Code, and it is named Marketplace.sol. In figure 6.3, a contract is initiated that holds records of the id, name, price, owner's address, and purchase history. It also consists of the events that can run and the variables it can hold. At first, the contract is compiled and then it is deployed in the network.

Next, In figure 6.4, the functions are stated, which are the main actions that are performed. The functions are createelectricity(name,price) and purchaseelectricity(id). These functions need some requirements to ensure the provided values in the parameters are accurate and correct, and at last, the events must be emitted. While the purchase is made, the owner's address must be swapped, and the buyer's address as the buyer will be considered the new owner of that particular amount of power.

In figure 6.5, the address of the generated smart contract and transactionHash is displayed.

Marketplace.sol M ×

src > contracts > Marketplace.sol

```
pragma solidity >=0.4.21 <0.6.0;
contract Marketplace {
    string public name;
    uint public electricityCount = 0;
    mapping(uint => electricity) public electricitys;

    struct electricity {
        uint id;
        string name;
        uint price;
        address payable owner;
        bool purchased;
    }

    event electricityCreated(
        uint id,
        string name,
        uint price,
        address payable owner,
        bool purchased
    );

    event electricityPurchased(
        uint id,
        string name,
        uint price,
        address payable owner,
        bool purchased
    );

```

Figure 6.3: Smart Contract Struct and Events

```
Marketplace.sol M ×
src > contracts > Marketplace.sol
    function createelectricity(string memory _name, uint _price) public {
        require(bytes(_name).length > 0);
        require(_price > 0);
        electricityCount ++;
        electricitys[electricityCount] = electricity(electricityCount, _name, _price, msg.sender, fals
        emit electricityCreated(electricityCount, _name, _price, msg.sender, false);
    }

    function purchaseelectricity(uint _id) public payable {
        electricity memory _electricity = electricitys[_id];
        address payable _seller = _electricity.owner;
        require(_electricity.id > 0 && _electricity.id <= electricityCount);
        require(msg.value >= _electricity.price);
        require(!_electricity.purchased);
        require(_seller != msg.sender);
        _electricity.owner = msg.sender;
        _electricity.purchased = true;
        electricitys[_id] = _electricity;
        address(_seller).transfer(msg.value);
        emit electricityPurchased(electricityCount, _electricity.name, _electricity.price, msg.sender,
    }
}
```

Figure 6.4: Smart Contract Functions.


```
"networks": {
  "5777": {
    "events": {},
    "links": {},
    "address": "0x1044b20c80864C2820749B81A93F3e5315A2779C",
    "transactionHash": "0x520b2082545f31bd8fee5f4c1750ba9a0ca978a1d5952477fbaabb192447021b"
  }
```


Figure 6.5: Smart Contract Address and transactionHash.

6.2.3 Web3.js

web3.js is a set of libraries that allows a system to communicate with an Ethereum node, either locally or remotely, using an HTTP or IPC connection. It establishes a connection between the smart contract and the blockchain. At first, node.js is installed as it will include all the needed libraries. Node.js is a run-time environment that provides every function needed to run a JavaScript program. The Ethereum blockchain is accessed with the web3 JavaScript library. It can retrieve user accounts, send transactions, and communicate with smart contracts, among other things. Several utility functions are also provided by using the web3.js, which makes development easier. In figure 6.6, the web3 is loaded that connects the Ethereum node to the smart contract.

In figure 6.7, the blockchain is loaded where the smart contract is stored. It loads all

```
async loadWeb3() {
  if (window.ethereum) {
    window.web3 = new Web3(window.ethereum)
    await window.ethereum.enable()
  }
  else if (window.web3) {
    window.web3 = new Web3(window.web3.currentProvider)
  }
  else {
    window.alert('Non-Ethereum browser detected. You should consider trying MetaMask!')
  }
}
```

Figure 6.6: Loading web3.js.

the accounts that are needed to perform the transactions. The Marketplace.abi holds the fundamentals of the smart contract and it is used to store the smart contract in the blockchain. As well as the address of the smart contract is also needed to connect them. Then the functions are also stored in the blockchain, where they are kept secure.

```
async loadBlockchainData() {
  const web3 = window.web3
  // Load account
  const accounts = await web3.eth.getAccounts()
  this.setState({ account: accounts[0] })
  const networkId = await web3.eth.net.getId()
  const networkData = Marketplace.networks[networkId]
  if(networkData) {
    const marketplace = web3.eth.Contract(Marketplace.abi, networkData.address)
    this.setState({ marketplace })
    const electricityCount = await marketplace.methods.electricityCount().call()
    this.setState({ electricityCount })
    // Load electricitys
    for (var i = 1; i <= electricityCount; i++) {
      const electricity = await marketplace.methods.electricitys(i).call()
      this.setState({
        electricitys: [...this.state.electricitys, electricity]
      })
    }
    this.setState({ loading: false})
  } else {
    window.alert('Marketplace contract not deployed to detected network.')
  }
}
```

Figure 6.7: Loading Smart Contract in Blockchain.

6.2.4 MetaMask Wallet

MetaMask Wallet is a cryptocurrency wallet that permits users to communicate with a blockchain with the help of a browser extension. The wallets provided by Ganache must be imported into the MetaMask Wallet using the private key of the accounts. Custom RPC must be created, which holds the IP HTTP://127.0.0.1: 7545 same as the Ganache RPC. Using these accounts, we can confirm the transactions and pay for the gas price and the bought energy. One of the imported accounts is named 'Seller', which can add selling items to the list and another account is named 'Buyer', which can buy the listed items and be the owner of that particular product. In figure 6.8, the first transaction is made to add the item to the list, and a minimum amount of gas value is needed to confirm it. The 'seller does the process,' and after that, the 'buyer' can buy the product using Ether from its account and it will also cost some gas to confirm the transaction.

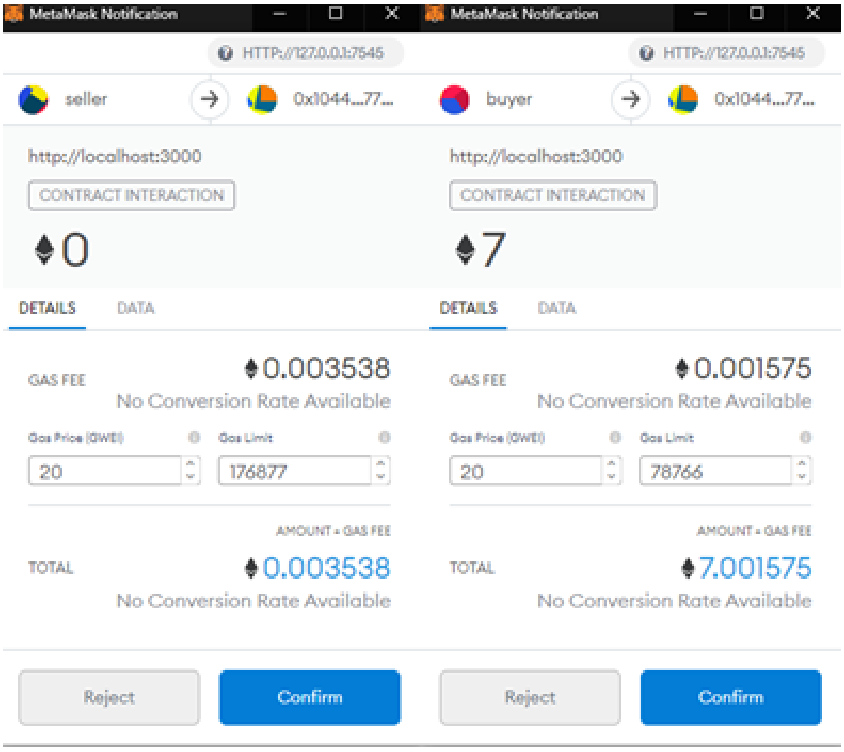

Figure 6.8: MetaMask Wallet for Seller and Buyer.

Table 6.1 Gas Used for every Block.

Block No	Mined Date	Gas Used
82	2021-06-02 3:53:33	52511
81	2021-06-02 3:53:01	117930
80	2021-06-02 3:49:43	117954
79	2021-06-02 3:48:47	117918
78	2021-06-02 3:48:17	117906
77	2021-06-02 3:48:02	132894
76	2021-06-02 3:45:42	745906
75	2021-06-02 3:45:42	244636

6.2.5 React.js

The front end of the application is made with React.js. React is a front-end JavaScript library for creating user interfaces and UI designs. It ensures fast rendering, as well as the maintenance functions, are excellent. The front end connects to the blockchain using the web3 functions. Every time the seller adds an item to the list by providing the parameters (name, price), an item is added to the ”Buy Electricity’ list. It will also show the owner of the item, which will be the seller while the items are listed. Then the buyer can buy those particular items by pressing the ’Buy’ button and confirming the transactions with MetaMask Wallet. By this, the Ether will be transferred to the seller from buyer and the ownership of the product will be swapped where the buyer’s address will be presented now. In figure 6.9, the whole front end is portrayed where we can access the functions.

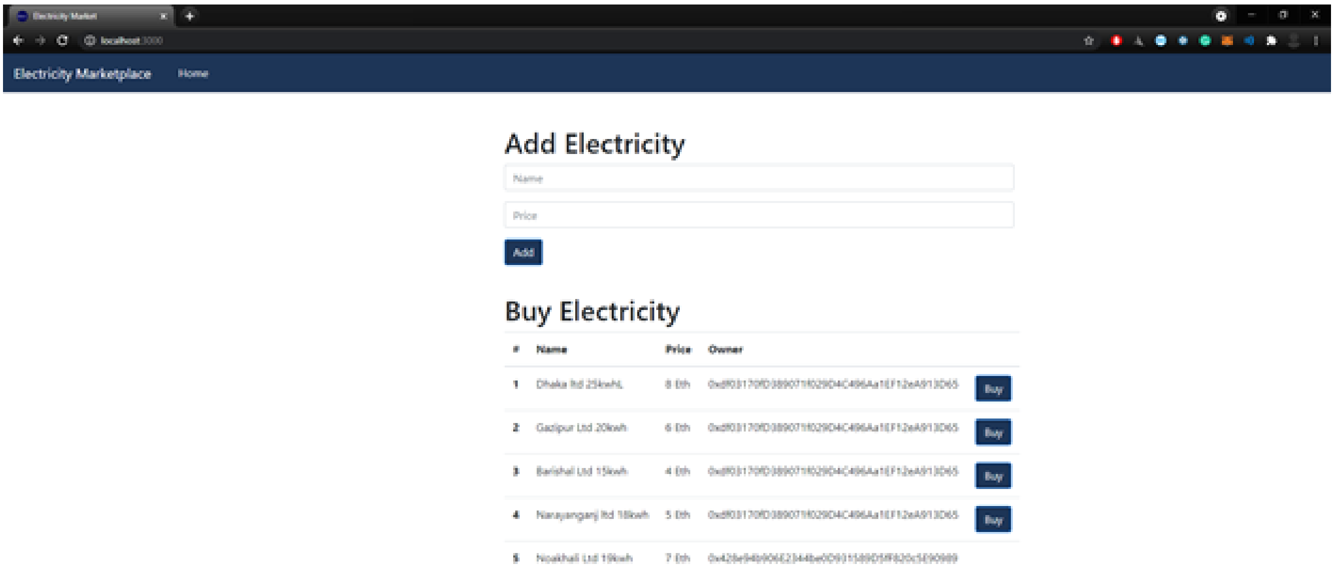

Figure 6.9: Front end application using React.js.

To conclude the implementation, a complete business DApp was demonstrated. In this system, the smart contract was created and deployed in the blockchain system as well as the front end, and the transactions were confirmed and stored in the updated blockchain network.Here is the work plan of the implementation system.

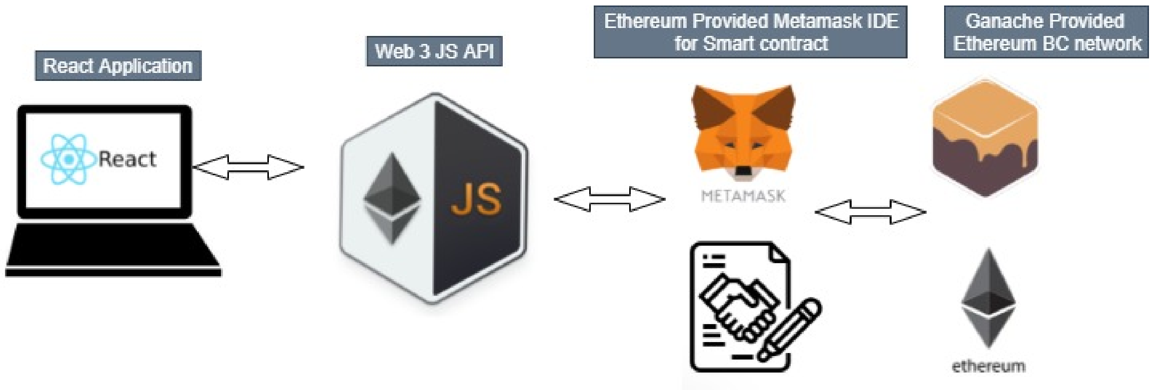

Figure 6.10: Workplan of Blockchain implementation.

6.3 Input Data

6.3.1 Producer Input

Part of the input has five different electricity producers in the form of Coal, Nuclear, Wind, Water, and Air. Since each of the producers produce different quantity of electricity in different times of the day, and prices them accordingly; the simulation uses 0/1 Knapsack as the algorithm to choose the best producer with the best price for retailers. The algorithm takes the maximum capacity of weight as W, the list that contains the electricity production weight in list W [C1, C2, C3, ...], and the prices for production in values V []. The algorithm filters through the list through brute force recursion and it calculates the total weight and value of all the subsets. Moreover, it will only consider the subsets whose total weight is smaller than the maximum capacity W.

6.3.2 Customer Input

The other inputs that the algorithm takes are the dissatisfaction parameters – dmul, alphan, and betan. Moreover, the customers curtailable demand and critical demand are also taken for determining the optimal retail price at specific times of the day given the high demand during peak hours and the low price at off peak hours.

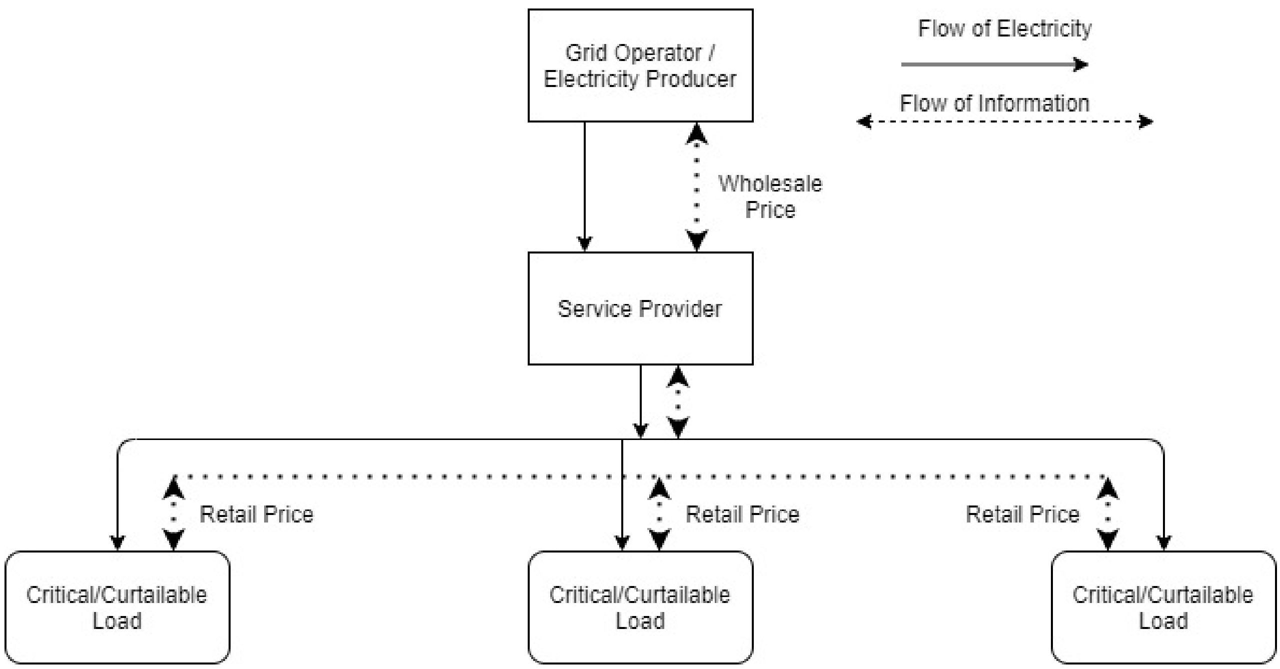

Figure 6.11: Producer, Provider and Consumer hierarchy

6.4 Numerical Simulation Results

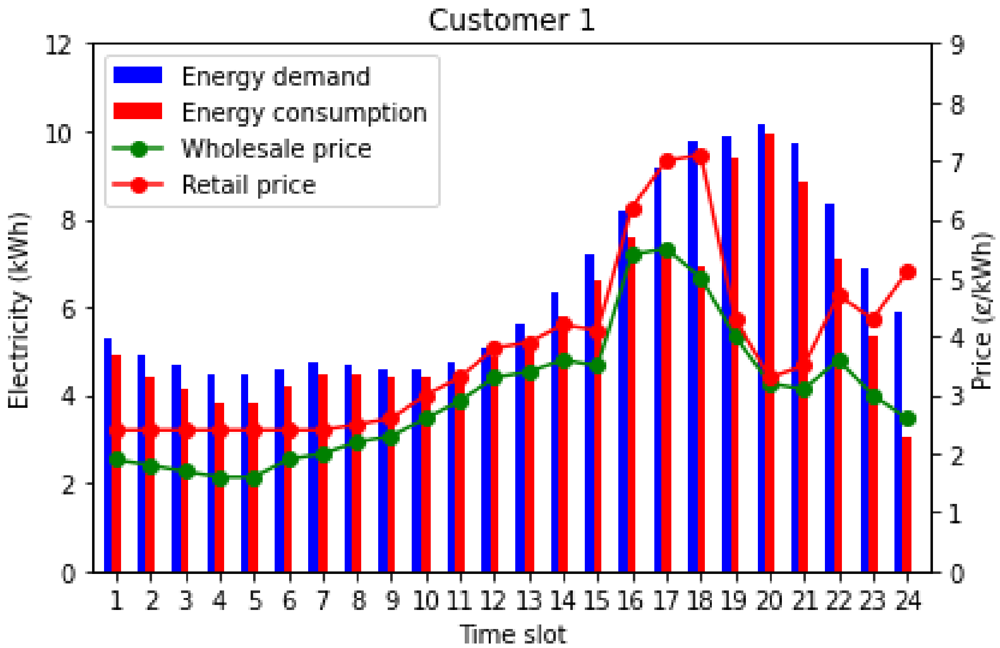

Figure 6.12: Energy, Demand consumption with optimal retail price for customer 1.

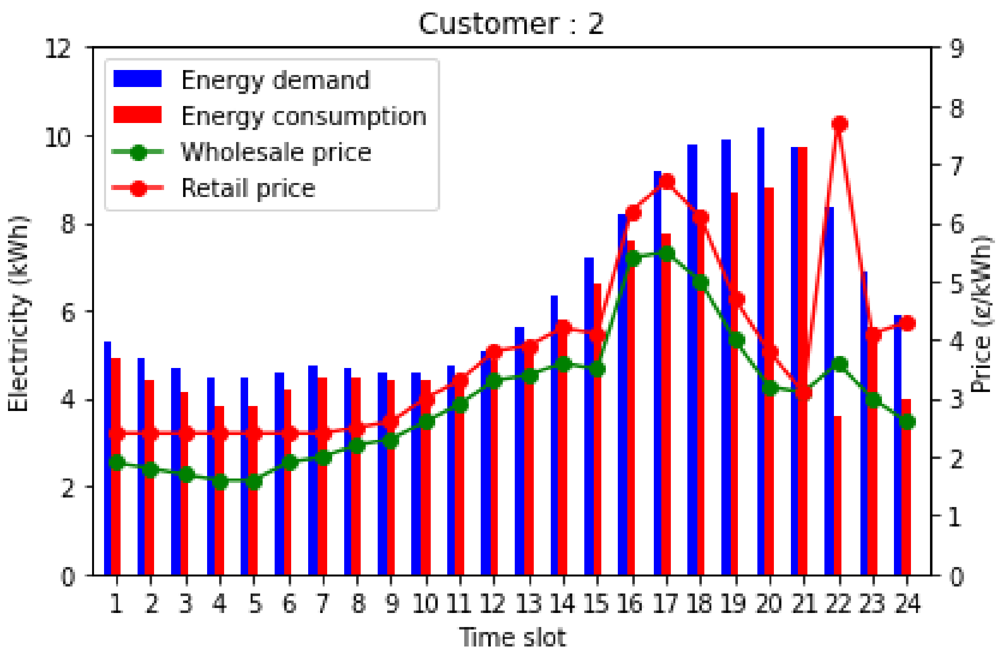

Figure 6.13: Energy, Demand consumption with optimal retail price for customer 2.

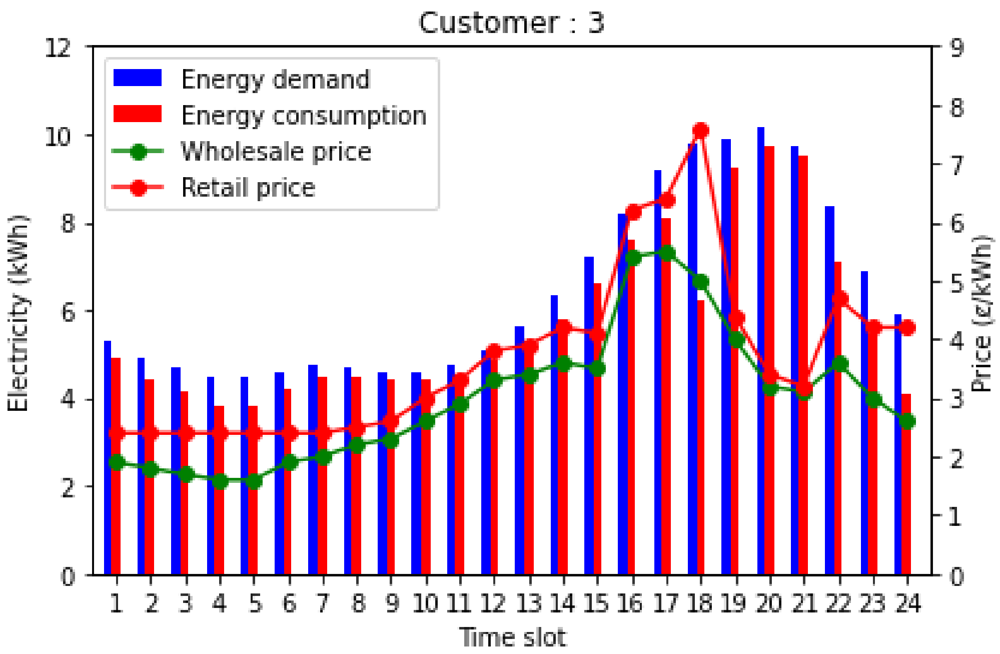

Figure 6.14: Energy, Demand consumption with optimal retail price for customer 3.

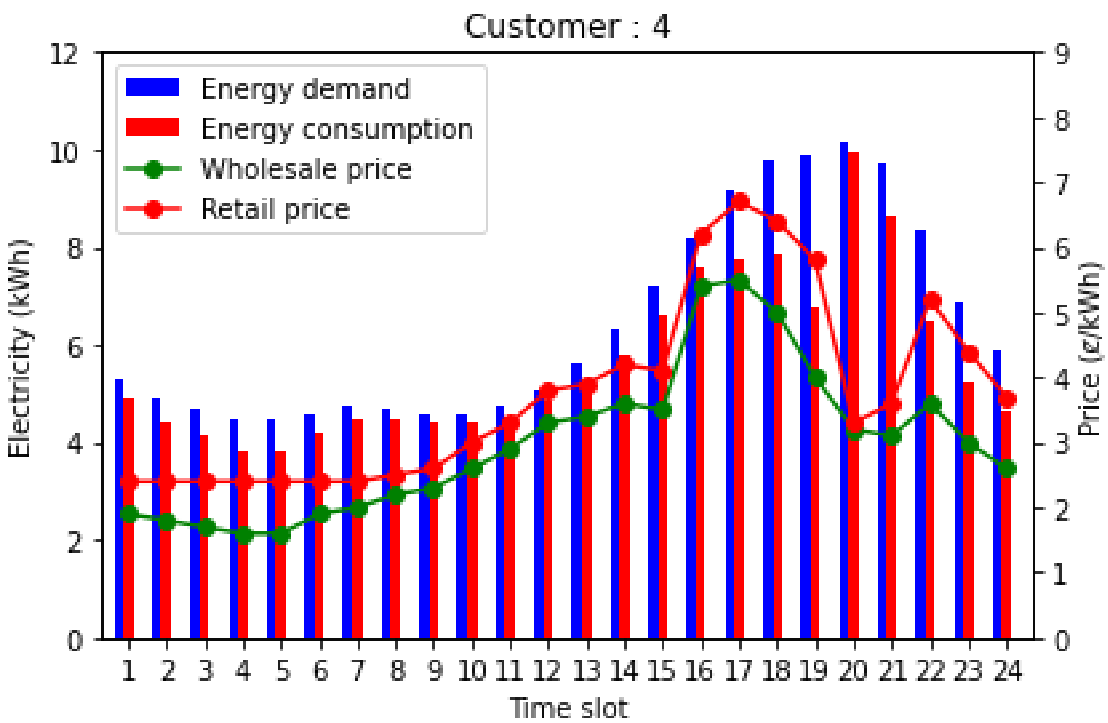

Figure 6.15: Energy, Demand consumption with optimal retail price for customer 4.

Each of the simulations above shows the Energy demand, consumption, the wholesale price, and the retail price per hour for an entire day. Each of the time slots represents the value of T in the sections above. Here, all the parameter values are specific and they can be changed as per the characteristics of the market and the providers, producers, and the customers. However, they do not change the overall performance of the simulation in a negative way.
From the graphs shown above, it can be seen that the agent at the beginning does not know the best actions that results in the higher Q-values, however, as the algorithm iterates, the agent learns from the environment gradually, and then finally converges to the maximum Q-value.

6.4.1 Optimal retail prices

The simulation results in the optimal retail prices for each of the customers in question. Each of the figures show the optimal retail price along with the prices from the providers for each time slot T. Moreover, the curtailable load and energy consumption is also shown for the customers. The graph also shows that the retail price shows very similar pattern to the wholesale price but never exceeds it due to the price bounds and dissatisfaction parameters. The price elasticity of the entire day shows a continuing increase in demand as well as the increase of retail price then a gradual decrease of both. This is because during peak hours, the electricity demand is maximum and more elastic; therefore, it results in a higher energy consumption.

Chapter 7

Conclusion and Future Research

7.1 Conclusion

In this paper, the major drawbacks of traditional grids as well as conventional electricity pricing strategy implementations which were based on abstract models are discussed. This paper proposes a dynamic DP algorithm which will not only benefit the SP profit but also minimizing CU costs. This model-free approach was done using RL where retail prices will be adaptive in nature and change based on the investigation of CUs' and dissatisfaction level and demand profile. This dynamic pricing problem of electricity is approached by converting it to a finite discrete MDP and then formulate the decision-making using Q-Learning. As a result, a SP does not need any specific model of the customers' energy consumption to learn about future retail rates. Rather, it is more convenient as the SPs will learn about CUs' satisfaction and dissatisfaction levels through on-line dynamic interaction. This approach is feasible as with the change of environment which in this case in the variation of CUs and their demand profile, the dynamic pricing model will improvise through learning. Thus, it works beneficiary for both SP and CU; ensuring a win-win strategy. Furthermore, a secured blockchain transaction is paramount for the security of both the SP and CU which is also discussed in this paper. In this paper, a blockchain based transaction DApp has been created where it works as a virtual electricity market place for CU. SPs will list their selected prices for a particular amount of load in the virtual market and CUs can buy it using Ether. As the transaction is decentralized and done in cryptocurrency, maximum security of the user's transaction activity is secured. The blockchain environment is created using a private blockchain named Ganache. The frontend is constructed using react.js and the backend with smart contracts which is a cumulation of transaction functions. Web3.js is used to connect the backend and front end. Metamask works as an online wallet to connect blockchain with browser.

7.2 Limitations

Although the blockchain transaction is highly secured, the PoW needed for authentication is relatively energy consuming. It takes a lot of computational power so in cases where fast transaction is needed, efficiency can sometimes be an issue. This efficiency can be highly negligible in most cases given the secured, reliable transaction method blockchain system provides.

7.3 Future Work

In the future, the current dynamic pricing model will be furthered improved upon by analyzing deeply the weighting factor and further scrutinize the optimal between the provider, producer and customers. The blockchain system can be improved by implementing advanced storage options like InterPlenatary File Sharing System (IPFS) which acts as a decentralized online storage to store information of users. User data in IPFS can have encryption facility for more security and convenience.

References

[1] M. R. Blouin and R. Serrano, “A decentralized market with common values uncertainty: Non-steady states,” en, *Rev. Econ. Stud.*, vol. 68, no. 2, pp. 323–346, 2001.

[2] Q. Qdr, “Benefits of demand response in electricity markets and recommendations for achieving them,” *US Dept. Energy, Washington, DC, USA, Tech. Rep*, vol. 2006, 2006.

[3] J.-W. Lee and D.-H. Lee, “Residential electricity load scheduling for multi-class appliances with Time-of-Use pricing,” in *2011 IEEE GLOBECOM Workshops (GC Wkshps)*, IEEE, 2011, pp. 1194–1198.

[4] R. P. Nair, “Proposed system for a smart grid implementation at oklahoma state university,” Ph.D. dissertation, Jul. 2011.

[5] Z. Zhou, F. Zhao, and J. Wang, “Agent-based electricity market simulation with demand response from commercial buildings,” *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 580–588, 2011.

[6] F.-D. Li, M. Wu, Y. He, and X. Chen, “Optimal control in microgrid using multi-agent reinforcement learning,” en, *ISA Trans.*, vol. 51, no. 6, pp. 743–751, 2012.

[7] J. Torriti, “Price-based demand side management: Assessing the impacts of time-of-use tariffs on residential electricity demand and peak shifting in northern italy,” en, *Energy (Oxf.)*, vol. 44, no. 1, pp. 576–583, 2012.

[8] E. Kuznetsova, Y.-F. Li, C. Ruiz, E. Zio, G. Ault, and K. Bell, “Reinforcement learning for microgrid energy management,” en, *Energy (Oxf.)*, vol. 59, pp. 133–146, 2013.

[9] R. Sutton, *Reinforcement learning, second edition : An introduction : An introduction*, en. London, England: MIT Press, 2013.

[10] P. Yang, G. Tang, and A. Nehorai, “A game-theoretic approach for optimal time-of-use electricity pricing,” *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 884–892, 2013.

[11] Y. M. Ding, S. H. Hong, and X. H. Li, “A demand response energy management scheme for industrial facilities in smart grid,” *IEEE Trans. Industr. Inform.*, vol. 10, no. 4, pp. 2257–2269, 2014.

[12] X. H. Li and S. H. Hong, “User-expected price-based demand response algorithm for a home-to-grid system,” *Energy (Oxf.)*, vol. 64, pp. 437–449, 2014.

[13] R. Rana and F. S. Oliveira, “Real-time dynamic pricing in a non-stationary environment using model-free reinforcement learning,” en, *Omega*, vol. 47, pp. 116–126, 2014.

[14] D.-C. Gao, Y. Sun, and Y. Lu, “A robust demand response control of commercial buildings for smart grid under load prediction uncertainty,” *Energy (Oxf.)*, vol. 93, pp. 275–283, 2015.

[15] B. Jiang and Y. Fei, “Smart home in smart microgrid: A cost-effective energy ecosystem with intelligent hierarchical agents,” *IEEE Trans. Smart Grid*, vol. 6, no. 1, pp. 3–13, 2015.

[16] A. Lahouar and J. Ben Hadj Slama, “Day-ahead load forecast using random forest and expert input selection,” en, *Energy Convers. Manag.*, vol. 103, pp. 1040–1051, 2015.

[17] S. Vandael, B. Claessens, D. Ernst, T. Holvoet, and G. Deconinck, “Reinforcement learning of heuristic EV fleet charging in a day-ahead electricity market,” *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 1795–1805, 2015.

[18] K. Vanthournout, B. Dupont, W. Foubert, C. Stuckens, and S. Claessens, “An automated residential demand response pilot experiment, based on day-ahead dynamic pricing,” en, *Appl. Energy*, vol. 155, pp. 195–203, 2015.

[19] A. Chis, J. Lunden, and V. Koivunen, “Reinforcement learning-based plug-in electric vehicle charging with forecasted price,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 5, pp. 1–1, 2016.

[20] Z. Luo, S.-H. Hong, and J.-B. Kim, “A price-based demand response scheme for discrete manufacturing in smart grids,” en, *Energies*, vol. 9, no. 8, p. 650, 2016.

[21] M. Yu and S. H. Hong, “Supply–demand balancing for power management in smart grid: A stackelberg game approach,” en, *Appl. Energy*, vol. 164, pp. 702–710, 2016.

[22] ——, “Supply–demand balancing for power management in smart grid: A stackelberg game approach,” en, *Appl. Energy*, vol. 164, pp. 702–710, 2016.

[23] E. U. Haq, H. Xu, L. Pan, and M. I. Khattak, “Smart grid security: Threats and solutions,” in *2017 13th International Conference on Semantics, Knowledge and Grids (SKG)*, IEEE, 2017, pp. 188–193.

[24] Y.-C. Li and S. H. Hong, “Real-time demand bidding for energy management in discrete manufacturing facilities,” *IEEE Trans. Ind. Electron.*, vol. 64, no. 1, pp. 739–749, 2017.

[25] C. Sillaber and B. Waltl, “Life cycle of smart contracts in blockchain ecosystems,” en, *Datenschutz Datensicherheit - DuD*, vol. 41, no. 8, pp. 497–500, 2017.

[26] H.-Z. Wang, G.-Q. Li, G.-B. Wang, J.-C. Peng, H. Jiang, and Y.-T. Liu, “Deep learning based ensemble approach for probabilistic wind power forecasting,” en, *Appl. Energy*, vol. 188, pp. 56–70, 2017.

[27] M. Yu and S. H. Hong, “Incentive-based demand response considering hierarchical electricity market: A stackelberg game approach,” en, *Appl. Energy*, vol. 203, pp. 267–279, 2017.

[28] J. Chiu and T. V. Koeppl, “Blockchain-based settlement for asset trading,” en, *SSRN Electron. J.*, 2018.

[29] M. Jin, W. Feng, C. Marnay, and C. Spanos, “Microgrid to enable optimal distributed energy retail and end-user demand response,” en, *Appl. Energy*, vol. 210, pp. 1321–1335, 2018.

[30] ——, "Microgrid to enable optimal distributed energy retail and end-user demand response," en, *Appl. Energy*, vol. 210, pp. 1321–1335, 2018.

[31] P. Kofinas, G. Vouros, and A. I. Dounis, "Energy management in solar microgrid via reinforcement learning using fuzzy reward," *Adv. Build. Energy Res.*, vol. 12, no. 1, pp. 97–115, 2018.

[32] P.-H. Kuo and C.-J. Huang, "An electricity price forecasting model by hybrid structured deep neural networks," en, *Sustainability*, vol. 10, no. 4, p. 1280, 2018.

[33] J. Lago, F. De Ridder, and B. De Schutter, "Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms," en, *Appl. Energy*, vol. 221, pp. 386–405, 2018.

[34] U. Ugurlu, I. Oksuz, and O. Tas, "Electricity price forecasting using recurrent neural networks," en, *Energies*, vol. 11, no. 5, p. 1255, 2018.

[35] *What are public keys and private keys?* en, https://www.ledger.com/academy/blockchain/what-are-public-keys-and-private-keys, Accessed: 2021-6-2, Oct. 2019.

[36] A. A. G. Agung and R. Handayani, "Blockchain for smart grid," en, *J. King Saud Univ. - Comput. Inf. Sci.*, 2020.

[37] S. Bhatnagar, G. Nahar, V. Maurya, R. Mathur, B. Student, and P. Scholar, "Smart grid -for smart cities," Aug. 2020.

[38] G. Iredale, *Blockchain technology history: Ultimate guide*, en, https://101blockchains.com/history-of-blockchain-timeline, Accessed: 2021-6-2, Nov. 2020.

[39] *Secured taxation operation using transaction functionalities of blockchain*, en, https://www.dpublication.com/abstract-of-10th-rstconf/10-20119/, Accessed: 2021-6-2, Dec. 2020.

[40] P. J. Taylor, T. Dargahi, A. Dehghantanha, R. M. Parizi, and K.-K. R. Choo, "A systematic literature review of blockchain cyber security," en, *Digit. Commun. Netw.*, vol. 6, no. 2, pp. 147–156, 2020.

[41] M. Zahid, I. Ali, R. J. U. H. Khan, Z. Noshad, A. Javaid, and N. Javaid, "Blockchain based balancing of electricity demand and supply," in *Lecture Notes in Networks and Systems*, Cham: Springer International Publishing, 2020, pp. 185–198.

[42] L. Conway, *Blockchain explained*, https://www.investopedia.com/terms/b/blockchain.asp, Accessed: 2021-6-2, Jun. 2021.

[43] D. Floyd, *How bitcoin works*, https://www.investopedia.com/news/how-bitcoin-works/, Accessed: 2021-6-2, May 2021.

[44] https://www.investopedia.com/news/how-bitcoin-works/, Accessed: 2021-6-2.

[45] https://stormgain.c, Accessed: 2021-6-2.

[46] http://incompleteideas.net/book/first/ebook/node12.html, Accessed: 2021-6-2.

[47] *Benefits of blockchain*, https://www.ibm.com/topics/benefits-of-blockchain, Accessed: 2021-6-2.

[48] S. Bhatt, *5 things you need to know about reinforcement learning - KDnuggets*, https://www.kdnuggets.com/2018/03/5-things-reinforcement-learning.html, Accessed: 2021-6-2.

[49] *Ganache*, https://www.trufflesuite.com/docs/ganache/overview, Accessed: 2021-6-2.

[50] *Intro to ethereum*, https://ethereum.org/en/developers/docs/intro-to-ethereum/, Accessed: 2021-6-2.

[51] *Public and private keys: What are they?* https://www.gemini.com/cryptopedia/public-private-keys-cryptography, Accessed: 2021-6-2.