Republic of Iraq
Ministry of Higher Education
and Scientific Research
University of Babylon
College of Engineering
Department of Electrical
Engineering

Intelligence Cloud Computing System for Home Load Pattern Larceny Detection and Power Factor Correction

A Thesis

Submitted to the College of Engineering / University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master in Engineering /Electrical Engineering / Electronic

By

Saja Abdul-Hamza Yas Khudhier

Supervised by

Prof. Dr. Laith Ali Abdul-Rahaim

Assist. Prof. Dr. Sarmad Khaleel Ibrahim

2023A.D *1445A.H*

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

نَرْفَعُ دَرَجَاتٍ مَّن نَّشَآءُ

وَفَوْقَ كُلِّ ذِى عِلْمٍ

عَلِيمٌ

صَدَقَ ٱللَّهُ ٱلْعَظِيمُ

سورة يوسف (76)

Dedication

To the one who has been my support and aid through all the trials in my life, to the most remarkable man in my life... My dearest father.

To the generous heart and caring source... My beloved mother.

To those whom God has blessed me with as my support, and they were the best of helpers... My family.

To my friends and colleagues, my companions in this journey.

To the esteemed supervisors who extended their helping hand.

I dedicate this work

Saja

October, 2023

Acknowledgment

In the name of Allah, Most Gracious, Most Merciful

I begin by praising Allah, the almighty, for granting me the opportunity and ability to undertake this thesis.

*I express my utmost appreciation to my thesis supervisors, **Prof. Dr. Laith A. Abdul-Rahaim** and **Assist. Prof. Dr. Sarmad K. Ibrahim** , for their unwavering help, encouragement, guidance, patience, and support throughout this research.*

My heartfelt thanks go to my friends and colleagues for their assistance, collaboration, and friendship. I am also grateful to the faculty and staff of the Department of Electrical Engineering for their kind support.

I express my deepest appreciation to all those who offered me a helping hand during this study.

Lastly, I would like to extend my profound gratitude to my family for their patience and unwavering support throughout my studies. Their love and encouragement have enabled me to complete this work.

Saja

October, 2023

Abstract:

Access to reliable electricity is vital for powering various sectors, including residential, industrial, and medical facilities. However, challenges such as energy larceny, excessive energy bills, and energy loss, hindering efficient and sustainable energy consumption. This thesis introduces an intelligent cloud system designed to tackle these challenges by detecting electricity larceny in real-time and identifying abnormal load patterns with power factor correction.

The proposed system utilizes a cloud server for data storage and analysis. It adopts a hybrid-oriented approach that combines both hardware and data oriented approaches to detect electricity larceny. The hardware-oriented approach involves installing sensors on the distribution network to identify illegal electricity consumers. Simultaneously, the data-oriented approach utilizes deep learning techniques to train a model that capable of detecting suspicious load patterns associated with theft.

The hardware-oriented approach has undergone testing in various conditions and has demonstrated its effectiveness and efficiency in accurately detecting electricity larceny. The data-oriented approach, utilizing Convolutional Neural Networks (CNN), has been selected for its ability to extract relevant features from electrical data. In this approach, a dataset called the "Smart Grid Dataset" provided by the State Grid Corporation of China (SGCC) was utilized. To address imbalanced data, several data preprocessing techniques were tested, including Random Under Sampling (RUS), Random Over Sampling (ROS), Synthetic Minority Over-sampling Technique (SMOTE), SMOTETomek, and Adaptive Synthetic Sampling (ADASYN). Among these techniques, the Adaptive Synthetic (ADASYN) technique was chosen to handle the data imbalance. The evaluation of the combined CNN model and ADASYN demonstrated its effectiveness in accurately detecting larceny, achieving an impressive accuracy of 97.22%, precision of 97%, and recall of 99.9%.

By integrating both approaches, the system significantly improves the identification and prevention of electricity theft, leading to enhanced system stability, reliability, and efficiency. Additionally, the system promotes responsible energy consumption, ensuring equitable access to electricity while reducing the financial burdens associated with excessive energy bills.

Cloud computing techniques have greatly facilitated the implementation and dissemination of these approaches. Cloud servers provide the necessary storage capacity and computational resources to execute the models, analyze data, and present results to the authorities involved in electricity theft detection. Cloud servers have simplified the distribution and utilization of this hybrid approach in the context of electricity theft detection.

In addition, we suggest utilizing machine learning techniques, specifically a random forest algorithm, to select the optimum capacitance for power factor correction, which is a metric indicating the efficiency of the electricity distribution system. By enhancing the power factor, we can effectively minimize energy wastage and decrease the overall cost of electricity provision.

The performance of the proposed approach was evaluated using various metrics. The model achieved an accuracy of 97.93% reflects the model's generalization ability to unseen data, suggesting its effectiveness in real-world scenarios. Moreover, the precision of 97.98% implies that the majority of choosing capacitance values is correct, while the recall of 97.93% suggests that a high percentage of actual capacitance values are correctly identified by the model. Overall, these results confirm the effectiveness and reliability of the random forest algorithm in correcting the power factor and optimizing the efficiency of the electricity distribution system. When comparing the results achieved by our system to those of others, it is evident that our system outperforms the alternative methods.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Definition
1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
AC	Alternative Current
Acc	Accuracy
ADASYN	Adaptive Synthetic Sampling
AMI	Advanced measurement infrastructure
APFC	Automatic Power Factor Corrector
API	Application Programming Interface
AUC	Area Under the Curve
C	Current
CC	Cloud Computing
CM	Consumer Meters
CNN	Convolutional Neural Network
Conv1D	One-Dimensional Convolutional Neural Network
CPM	Checkpoint Meter
CPU	Central Processing Unit
CRUD	Create, Read, Update, And Delete
CT	Current Transformer
DBRNN	Deep Bidirectional Recurrent Neural Networks
DE-RUSBoost	Differential Evaluation Random Under Sampling Boosting
DL	Deep Learning
E	Energy

EF	Entity Framework
ELM	Extreme Learning Machine
EM	Measurement Errors
F	Frequency
FC	Fully Connected Layer
FN	False Negatives
FP	False Positives
GPS	Global Positioning System
GPS	Global Positioning System
GSM	Global System For Mobile Communications
GUI	Graphical User Interface
HDD	Hard Disk Drive
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IaaS	Infrastructure As A Service
IBM	International Business Machines Corporation
IFTTT	If This, Then That
IIS	Internet Information Services
IoT	Internet Of Things
IP	Internet Protocol
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
IR	Infrared Sensor
IT	Information Technology
KNN	K-Nearest Neighbors
LSTM	Long Short-Term Memory,

ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
MVC	Model-View-Controller
NTL	Non-Technical Losses
ORM	Object-Relational Mapping
P	Power
PaaS	Platform As A Service
PF	Power Factor
PFC	Power Factor Correction
RAM	Random Access Memory
ReLU	Rectified Linear Unit
RF	Random Forest
RNN	Recurrent Neural Networks
ROS	Random Oversampling Technique
RUS	Random Undersampling
RUSBoost	Random Under Sampling Boosting
SaaS	Software As A Service
SD card	Secure Digital card
SG	Smart Grid
SGCC	State Grid Corporation Of China
SMOTE	Synthetic Minority Over-Sampling Technique
SMOTETomek	SMOTE And Tomek Links
SQL	Structured Query Language
SSD	Solid State Drive
SVM	Support Vector Machine
TCP	Transmission Control Protocol

TL	Technical Losses
TTL	Transistor-Transistor Logic
TN	True Negatives
TP	True Positives
V	Voltage
VAE-GAN	Variational Autoencoder-Generative Adversarial Network
VT	Voltage Transformer
Wi-Fi	Wireless Fidelity
NaN	Not a Number
PWM	Pulse Width Modulation

List of symbols

Item	Description	Unit
f_i	The Activation Function Used For The Convolutional Layer.	Unitless
x_i	The Input To The Convolutional Layer.	Unitless
F_1	F1 Score	Unitless
PF_1	The Current Power Factor Of The System.	Unitless
PF_2	The Targeted Power Factor For PFC.	Unitless
PT_k	Power Theft In The $\boldsymbol{k^{th}}$Sector	Unitless
PT_1	Power Theft In The First Sector	Unitless
PT_2	Power Theft In The Second Sector	Unitless
Pf_d	The Desired Power Factor.	Unitless
Pf_r	Power Factor For Real C	Unitless
P_i	Power Readings At Individual Consumers With Identification Number i	W
b_i	The Offset Vector Or Bias Associated With The Convolutional Layer.	Unitless
m_i	i^{th} Element Of The Dataset (Data Point).	Unitless
m_{i+1}	$(i+1)^{th}$ Element Of The Dataset (The Next Data Point).	Unitless
m_{i-1}	$(i-1)^{th}$Element Of The Dataset (The Previous Data Point).	Unitless
w_i	The Weights Associated With The Convolutional Layer.	Unitless
x_{max}	The Maximum Value In That Column.	Unitless
x_{min}	The Minimum Value In That Column.	Unitless
$x_1, x_2, \dots, x_n$	Individual Data Points In The Row	Unitless
y_i	The Output Of Each Convolutional And Max-Pooling Layer In A Convolutional Neural Network (CNN).	Unitless

Acc	Accuracy	Unitless
Nan	Not a Number	Unitless
$\text{Max}(y_{i,j})$	The Maximum Value Among The Outputs $y_{i,j}$ Obtained From Different Convolutional Layers.	Unitless
$C\ calc$	The Calculated Capacitance.	F
$C\ real$	The Real or Actual Capacitance.	F
EM	Measurement Errors	Unitless
FN	False Negatives	Unitless
FP	False Positives	Unitless
P	Precision	Unitless
PF	Power Factor	Unitless
PT	Power Theft	Unitless
R	Recall	Unitless
S	*Apparent Power*	Volt Ampere
TL	Technical Losses	W
TN	True Negatives	Unitless
TP	True Positives	Unitless
V	Voltage	Volt
VAR	The Amount Of Reactive Power Required For Power Factor Correction (PFC) In An Electrical System.	VAR
c	Capacitance Value	Farad
$c(k)$	Power Readings at Checkpoint Meter With Identification Number k	W
f	Frequency	Hz
$f(m_i)$	The Approximate Value	Unitless

$f(x)$	The normalized value of the data point x, scaled between 0 and 1 using min-max scaling.	Unitless
m	Mean Of The Data Points In A Row.	Unitless
n	Total Number Of Consumers	Unitless
n	Number Of Data Points In The Row.	Unitless
sd	Standard Deviation Of The Data Points In a Row.	Unitless
x	A Data Point In a Column Of The Smart Grid Dataset.	Unitless
y'	The Final Output Obtained After Applying Max-Pooling To The Outputs $y_{i,j}$From All Convolutional Layers.	Unitless
$\theta1, \theta2$	The Angles Whose Tangent Values Correspond To PF_1And PF_2 Respectively.	Radians or Degrees

1

CHAPTER ONE

INTRODUCTION

Chapter One

Introduction

1.1 Introduction

Increasing the electricity demand has resulted in increased electrical losses, encompassing both technical (TL) and non-technical losses (NTL). TLs are attributed to inefficiencies in electricity transmission and distribution, whereas NTLs arise from larceny, non-payment, faulty and outdated meters, and illegal connections [1]. Our system is specifically devised to minimize both TLs and NTLs, where TLs can be mitigated by power factor correction, while NTLs can be mitigated by the detection of the electrical larceny.

The power factor is a measure of how effectively the load converts the delivered electrical energy into useful energy. A low power factor indicates that a load is not utilizing the available power effectively, resulting in energy waste, increased energy costs, and reduced system efficiency [2]. Power factor correction has become increasingly important in recent years, as energy costs have raised and environmental concerns have grown. Improving the power factor helps to reduce carbon emissions and other harmful pollutants associated with energy generation [3].

Electricity larceny is a significant problem for utility companies globally, as it makes up a significant percentage of their overall losses. The annual losses due to NTLs amount to approximately 89.3 billion U.S. dollars for electricity companies worldwide [4]. Electricity larceny can take many forms, from tampering with the electricity meter to bypassing it altogether. This can cause power outages and damage to electrical equipment. Electricity larceny can also have serious economic consequences, as it leads to revenue losses for utility companies and higher electricity prices for consumers [5].

The electric power system is divided into three main parts: generation, transmission, and distribution [6]. NTLs primarily occur in the distribution part, as this part is closest to the end-users, making violations easy to carry out. Therefore, it is crucial to take this issue seriously by developing and implementing a smart system that can effectively address non-technical losses in a low-cost and efficient manner[18].

The smart grid (SG) concept has been developed to tackle the challenges of efficiency, reliability, and security in the power grid by incorporating the activities of all users through the use of sensors [7]. To increase the effectiveness of the electric grid, several sensors are being added to it. These sensors allow communication between home appliances and power generators. However, as the smart power grid generates a large amount of data through its embedded sensors, an important question arises as to where and how this data is stored, processed, and analyzed. One solution that has emerged is utilized of cloud computing [8].

Cloud computing (CC) has revolutionized information technology infrastructure and provided users with unprecedented flexibility by delivering on-demand computing resources over the network. These resources may include processors, storage, software, network, etc. With CC, users can leverage computing resources for the required time on a "pay-as-you-go" model, making it cost-effective and flexible. This technology also enables users to access resources from anywhere, unlike traditional computer systems where one needs to be physically present where the resource is located [9]. CC has been widely adopted by various organizations, governments, and businesses across the world, given its scalability, robustness, availability, and cost-effectiveness [10].

To optimize the performance of the SG system, it is essential to expand communication networks and ensure decentralized data storage for efficient data management. Recent research indicates that integrating cloud computing into the smart grid system can significantly enhance its overall performance. By incorporating cloud computing, the SG can benefit from remote access, real-time data control, monitoring, and regardless of location and time [11].

Another approach that can be integrated with SG technology is the utilization of Machine Learning (ML) and Deep Learning (DL), which play a crucial role in identifying patterns and generating valuable insights [12]. ML and DL have the capacity to process massive amounts of data and uncover relationships and patterns that may be difficult for humans to detect [13]. Where Cloud-based technologies enable the deployment of ML and DL models, thereby enhancing the capabilities of the SG. DL is a powerful form of ML and has proven its effectiveness in various domains such as image classification and language processing. Recently, DL has gained significant attention for its potential applications in the SG field, including load prediction, fault identification, and security [14].

By integrating cloud computing, ML, and DL technique, the SG system becomes more effective in detecting and mitigating TLs and NTLs.

1.2 Problem Statement

The proliferation of home devices and appliances has resulted in a growing need for a system that can efficiently monitor and optimize power consumption in order to reduce energy costs. Consequently, the problem statement encompasses three key objectives:

1. The development of advanced monitoring systems that is smarter, more resilient, and less dependent on human intervention to ensure accurate and efficient monitoring of power consumption.
2. The design and implementation of an intelligent CC system that employs sophisticated algorithms, such as DL, to analyze historical consumption data

and identify anomalies in-home load patterns to help reduce NTLs and ensure the efficiency of SG.

3. The system provides power factor correction and energy optimization solutions for smart homes to help reduce TLs and overall costs.

Addressing these problems is essential to enable the efficient and sustainable use of energy in smart homes, which is becoming increasingly important in the face of rising energy costs.

1.3 Motivation

The electricity demand has been steadily increasing over the years due to population growth, economic development, and an increase in the use of electrical appliances [15]. Figure 1.1 illustrates an increase in electricity generation over the years for Iraq.

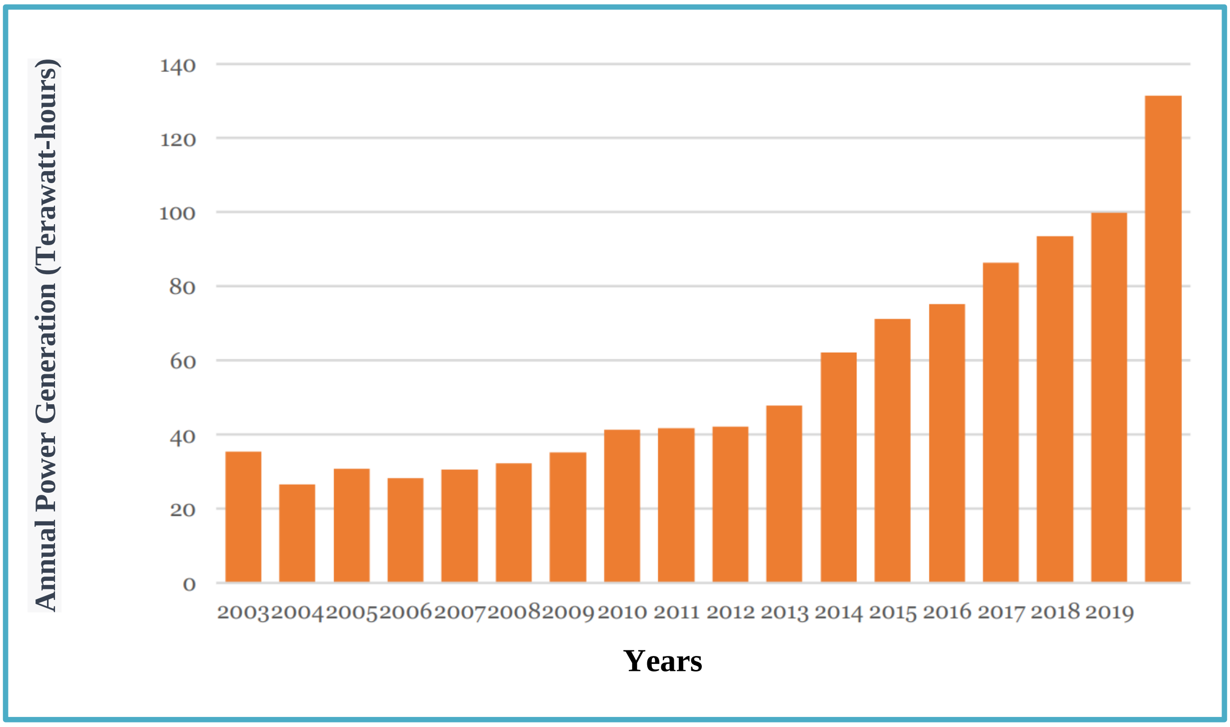


Figure 1.1: Increasing in electricity generation over the years for Iraq [15].

However, the country faces significant challenges in meeting this demand, with losses in transmission and distribution ranging between 40-50%, with up to 90%

of these losses attributed to distribution losses, including TLs and NTLs such as larceny, non-payment, faulty and outdated meters, and illegal connections. Moreover, the lack of maintenance and inefficient network in Iraq also contributes to TLs, while the global average for transmission and distribution losses is around 8% [15]. The transmission and distribution losses in the region are shown in Figure 1.2.

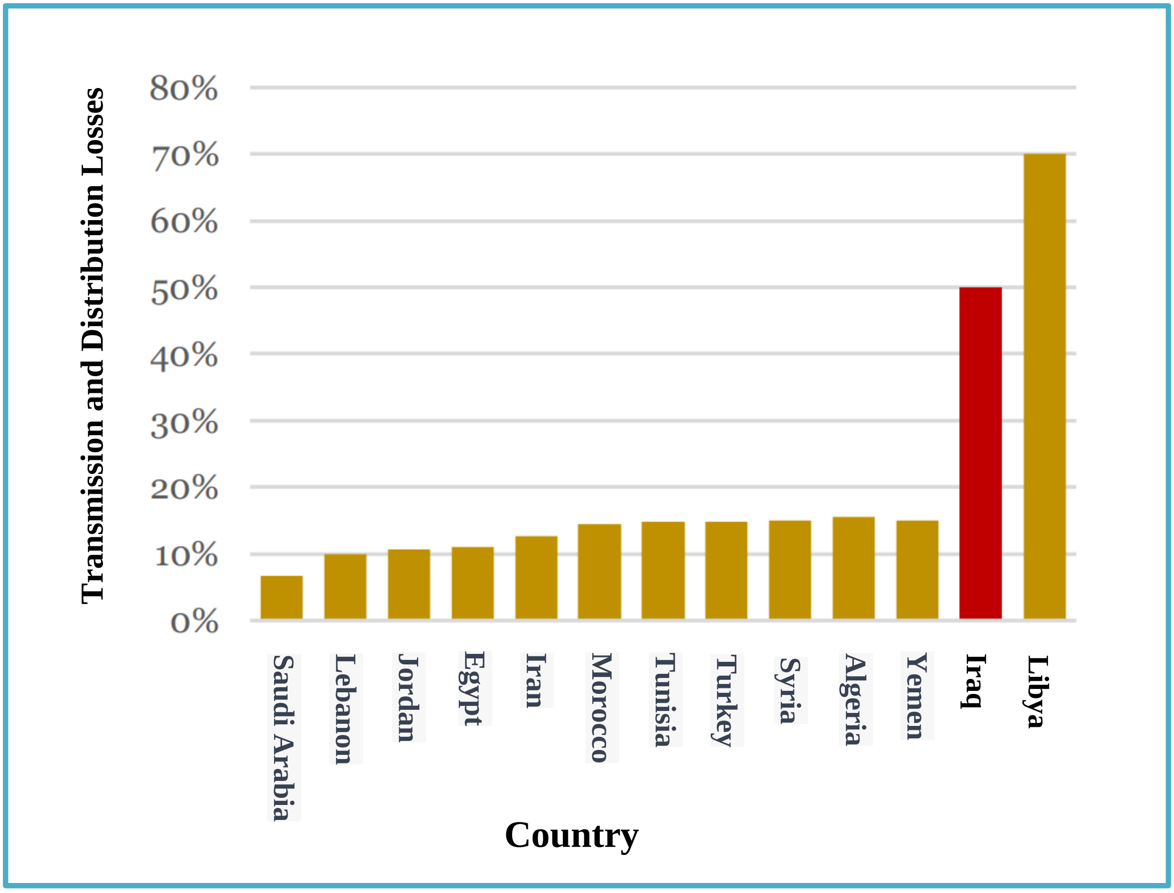


Figure 1.2: Transmission and distribution losses in the region [15].

Approximately two-thirds of energy generated is unpaid, with 23% lost due to illegal connections. Outdated and faulty meters are also prevalent, with around 80% of meters being over 30 years old and many not calibrated at all. Even those who pay their bills only cover around 10% of the true cost of electricity provision. Therefore, Iraq needs to invest in modernizing and upgrading its electricity infrastructure, reducing losses, and improving efficiency in order to meet the increasing demand for electricity and ensure sustainable economic growth [15].

The development of smart metering infrastructure is a crucial component of modernizing and upgrading Iraq's electricity infrastructure. It can contribute to solving electricity-related challenges by providing accurate and timely data on energy consumption, enabling utilities to optimize their operations and reduce losses.

In addition, developing a cloud computing system for home load pattern larceny detection and power factor correction can also contribute to solving the electricity-related challenges in Iraq. Detecting and preventing electricity larceny can reduce NTLs, therefore, reducing energy waste. Moreover, it reduces the demand on the system, leading to lower levels of fuel consumption. Additionally, the power factor correction can optimize energy use and reduce energy waste, leading to lower TLs, and lower costs for utilities and customers.

Ultimately, these measures can enhance customer satisfaction by providing more reliable and consistent service to all customers, and ensure sustainable economic growth.

1.4 Literature Review

The section covers the relevant literature on the subject matter, including the various techniques and methodologies used in designing such systems.

1.4.1 Homes Electric Energy Larceny Detection

In the field of electrical larceny detection, researchers employ various approaches and techniques to identify electrical larceny effectively. These approaches typically involve the use of two methodologies which are:

1.4.1.1 Hardware-based solution

R. E. Ogu et al (2016) [16] utilized an Arduino Wi-Fi Shield with an Arduino Mega 2560 board for networking and control purposes. Sensing and actuating were managed using an Infrared Sensor and Relay. Through this setup, the meter can communicate its GPS location to the distribution company's via cloud-based storage, ensuring data preservation. The system continuously monitors the meter for any tampering attempts, and if any are detected, it immediately disconnects the load attached to the meter from the distribution grid.

N. Pranau et al (2017) [17] developed a tampering detection method based on the summation of power provided by the transformer and the power consumed by clients. By comparing these values, any tampering attempts can be identified. The system retrieves electricity usage and distribution data from the relevant meters, which are then uploaded to the electrical board's monitoring portal and stored in the cloud for analysis. If any theft is detected, the system immediately notifies the server page, triggering an alarm installed near the server. This rapid response ensures that the theft can be promptly prevented or addressed.

R. Meenal et al (**2019**) [19] develop a system to track and detect electricity larceny by transforming data from the home Electricity Board, utilizing multiple sensors (Current and Voltage sensors) that transmit real-time data. The Raspberry Pi detects energy theft and communicates the data to the Electricity Board via the GSM module. Increased sensor sensitivity allows for the detection of unauthorized activities in the transmission lines.

In July 2020, M. J. Jeffin. et al [1] used a linear regression method to detect energy theft. The Internet of Things technology was included in the meter to send data to the server/database for analysis/storage respectively. Also, they developed an Android app to monitor consumption information and send notifications when a theft has occurred.

R. Aswini et al (2020) [20] designed a system based on fuzzy logic to track and quantify the number of units consumed daily by homes. And send the consumption bill automatically to the client. Also, enter the infrared sensor to identify any illegal change and send an alert message to the customer.

K. Kumaran et al (2021) [21] designed a system that monitors current withdrawal from a transformer in the current line to detect electrical power theft. If there is overloading, the sensor readings will increase above a threshold value, so the power supply will cut off automatically, and a message will be sent to officials by GSM via the Internet of Things. Use the bell to alert officials about power theft. Using the Message Queuing Telemetry Transport (MQTT) broker to transfer measured data from sensors to the Think-Speak cloud where load consumption information is stored and energy theft detection.

In December 2021, Mafaz et al [22] detected electrical fraud like overriding meters and direct connectivity based on cloud computing (a platform as a service). They used a raspberry pi as a centralized cloud unit to implement a theft detection algorithm and monitor massive distribution networks. This method reduced the devices used because it implements one algorithm in one central cloud unit for multi-home detection. It also supports advanced metering and offers knowledge for the power providers that need to expand in electrical networks.

S. Saadhavi et al (2022) [23] designed a system to monitor electrical theft, which constantly measures the amount of power distributed and consumed and compares the two readings. Based on the difference, determine if the theft occurred. They also developed a desktop app to enable the manager to monitor consumption data and take appropriate action when needed. In addition, they developed a website to allow the consumer to track his energy consumption data. The system stores the measured data from the electricity distribution system in GoDaddy's database.

V. Saritha et al (2022) [24] designed a smart electricity meter that automatically measures and creates bills. He used AllThingsTalkMaker as a centralized cloud to monitor and save data received from the meter. The meter sends an overload indicator and information about the meter tampering to the sub-power station, so it helps detect theft.

Table 1.1 provides a comprehensive comparison of the key characteristics of various hardware-based electricity larceny detection systems described in the literature survey, including the core hardware devices, communication schemes, sensors, user interfaces (UI), and the methods used for detection and system type.

Table 1.1: Comparison of Characteristics of hardware approach Energy Larceny Detection Systems in Literature Survey with Our System

Ref	Core hardware device	Com-node	Com. Scheme	Sensors	UI	Detection way and system type
[16]	Arduino	Arduino Wi-Fi Shield	Wi-Fi	IR sensor	Web page	tampering sensing / Cloud storage
[17]	Local server	ESP8266	Wi-Fi	MOD bus with meter	Web page	Power difference / Cloud storage
[19]	Arduino	GSM Module	GSM	CT, VT	LCD	Current sensing with threshold/ IoT
[1]	ATmega328P	Mobile app	Wi-Fi	CT, VT, IR sensor	Mobile app	Leaner Regression / cloud storage
[20]	PIC16F877 microcontroller	GSM Module	GSM	CT, VT	web app + LCD	Fuzzy logic/cloud administration data
[21]	Arduino Uno	GSM Module	GSM	CT, VT	web app + LCD	Current difference /cloud storage
[22]	Raspberry Pi	Raspberry Pi	Wi-Fi	PZEM-004T	Web page	Current difference /cloud storage
[23]	Arduino	NodeMCU	Wi-Fi	CT	Desktop app + web app	Power difference /cloud storage
[24]	Arduino	NodeMCU	Wi-Fi	CT, VT	LCD + web page	Current sensing with threshold/cloud storage
Our system	Arduino mega Wi-Fi	Arduino mega Wi-Fi	Wi-Fi	PZEM-004T	Web page + LCD	Power difference / Cloud storage and analysis

1.4.1.2 Data-based solution

Jain et al (2016) [25] proposed a hybrid approach using Support Vector Machine (SVM) and Extreme Learning Machine (ELM) to detect electricity theft by suspected consumers. They collected data from Maharashtra Government electricity utility and trained the system with established rules. The approach yielded a list of suspected customers engaged in fraudulent activities, with an accuracy of 94.19%, and outperformed the K-Nearest Neighbors (KNN) algorithm in terms of speed and efficiency.

S. Chatterjee et al (2017) [26] proposed an Artificial Intelligence (AI)-based solution to detect power theft and non-technical losses. They used Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) units to analyze half-hourly power consumption data and identify abnormalities. The model achieved a test accuracy of 72.93% on specific households and 65.3% on localities.

Z. Zheng et al (2018) [27] proposed electricity-theft detection method based on a Wide & Deep Convolutional Neural Network (CNN) model for smart grids. The model consists of two components: the Wide component and the Deep CNN component. The Deep CNN component accurately identifies the no periodicity of electricity theft and the periodicity of normal electricity usage using 2-D electricity consumption data. The Wide component captures global features from 1-D electricity consumption data. The model achieved an AUC (Area Under the Curve) of 0.7815.

Hasan et al (2019) [28] proposed a system for detecting electricity theft utilizi1ng a hybrid CNN-LSTM deep learning model. This model combines the strengths of CNN and LSTM networks. CNN is responsible for extracting features, while LSTM is specifically designed to handle time-series data. To handling of class imbalance problems, Synthetic Minority Over-sampling

Technique (SMOTE) was employed. The model achieved an impressive overall accuracy of 89%.

Adil et al (2020) [29] introduced a model that combines LSTM and bat-based Random Under Sampling Boosting (RUSBoost) techniques for the effective handling of class imbalance problems. The RUSBoost technique specifically addresses the issue of imbalanced data by undersampling the majority class while boosting the minority class. On the other hand, the bat algorithm is employed for parameter tuning to optimize the performance. The evaluation metrics of the proposed model demonstrate its effectiveness, with an F1-score of 96.1%, a precision of 88.9%, a recall of 91.09%, and a ROC-AUC of 87.9%. While the proposed model outperformed alternative techniques, the paper highlights its sensitivity to changes in input data. This suggests that the model's performance may vary when applied to different datasets or when the data undergoes significant changes.

Mujeeb et al (2020) [30] presented an enhanced model for electricity theft detection that incorporates the Differential Evaluation Random Under Sampling Boosting (DE-RUSBoost) classifier. This classifier is optimized using the Differential Evaluation (DE) meta-heuristic optimization algorithm. The proposed method achieved a high accuracy of 96%, and a false detection rate of 0.004.

Chen et al (2020) [31] introduced a novel approach for detecting electricity theft. This approach utilizes deep bidirectional recurrent neural networks (DBRNN) to effectively analyze time-series data. By combining the strengths of both DRNN and Bi-RNN, the proposed method captures both the internal characteristics and external correlations within the data. The DBRNN model achieves impressive performance metrics. The accuracy of the model is recorded as 97.44%, a precision of 95.70%, a recall of 99.74%, and the F1-Score 92.09%.

Syed et al (2020) [32] developed an LSTM-based electricity theft detection model. The model is capable of capturing long-term dependencies in data sequences. The proposed model achieves an accuracy of 92.69%. The authors conclude that their proposed methodology is simpler and can be applied to various SG applications.

Pereira et al (2021) [33] used CNN for detecting instances of electricity theft. The researchers addressed the challenge of unbalanced class distributions in the dataset by employing several techniques. These techniques include Random Undersampling, Cost-Sensitive Learning, Random Oversampling, Synthetic Minority Oversampling Technique, K-medoids-based Undersampling, and Cluster-based Oversampling. The study concludes that the Random Oversampling technique (ROS) performs the best in terms of Area Under the Curve (AUC), with a value of 0.6714.

J. Huang et al (2021) [34] developed an energy theft detection method using the power consumption information acquired from power enterprises. The method involves clustering and training a neural network named DWMCNN at a centralized data center. DWMCNN is an advanced CNN framework designed to extract features from users' electricity consumption data with a focus on daily, weekly, and monthly patterns. Then, at the edge data center, the extracted features are classified using the random forest (RF) algorithm. The CNN feature extractor effectively identified periodic patterns in the data. The DWMCNN-RF combination model demonstrated good accuracy. The proposed method offers advantages in terms of computing time, data privacy, and accuracy, making it suitable for deployment in edge data centers.

Ouamane (2021) [35] conducted research on detecting fraudulent behavior in power usage. The study involved the development of deep learning-based models utilizing both 1D-CNNs and 2D-CNNs. The 1D-CNNs model achieves an accuracy of 94.52%, and an AUC value of 77.66%. On the other hand, the 2D-CNNs model did not perform as well as the 1D-CNNs model. The accuracy of the 2D-CNNs model was 93.14%.

Lucas et al (2022) [36] employed a BiGRU-CNN artificial neural network, which combines the strengths of bidirectional gated recurrent unit (Bi-GRU) and CNN architectures. This hybrid approach allows the model to extract long-term temporal correlations through the Bi-GRU layer and capture local trends using the CNN layer. The model was evaluated utilizing various metrics. The accuracy of the model is reported as 0.929, the precision as 0.885, the recall as 0.801 the F1-Score 0.841, and the ROC AUC of 0.966.

Youngghyu et al (2023) [37] present a model that tackles the challenge of imbalanced data by generating synthetic electricity theft data with the same characteristics as real data utilizing VAE-GAN (variational autoencoder-generative adversarial network). Once the balanced dataset is obtained, a CNN model is employed as the detector. The reported model's performance values for precision, true positive rate, and F1-score are 0.925, 0.909, and 0.905 respectively.

Table 1.2 provides a summary of various techniques used in data-based electricity theft detection, highlighting their performance metrics and dataset sources. Notably, some techniques achieve high accuracy, precision, recall, and AUC, indicating their effectiveness in addressing this critical issue. Additionally, the proposed system stands out as one of the top-performing approaches in the comparison, with impressive accuracy and recall values. Our system uses a CNN model with the ADASYN data balancing technique, achieving an accuracy of 97.22%, precision of 96.8%, recall of 99.9%, and an AUC of 97.22%, making it one of the top-performing techniques.

Table 1.2: comparison of electricity data approach larceny detection techniques in the Literature Survey

Ref	Techniques applied	Data Balancing Techniques	Accuracy	Precision	Recall	AUC	Dataset
[25]	SVM and ELM	-	94.19%	-	-	-	Maharashtra Govt. Elec Utility Data
[26]	RNN and LSTM	-	72.93%	-	-	-	Collected data
[27]	Wide & Deep CNN	-	-	-	-	78.15%	SGCC
[28]	CNN-LSTM	SMOTE	89%	94.04%	87%	-	SGCC
[29]	LSTM	RUSBoost	87.9%	88.9%	91.09%	-	SGCC
[30]	DE-RUSBoost	RUSBoost	96%	90.2%	-	-	SGCC
[31]	DBRNN	-	97.44%	95.70%	99.74%	-	Irish Smart Energy Trials
[32]	LSTM	-	92.6941	-	-	-	SGCC
[33]	CNN	ROS	78.61%	-	-	-	SGCC
[34]	DWMCNN-RF	K-mean clustering	-	97%	97%	97%	SGCC
[35]	1D-CNN 2D-CNN	-	94.52% 93.29%	-	-	77.66% 72.72%	SGCC
[36]	BiGRU-CNN	-	92.9%	88.5%	80.1%	96.6%	SGCC
[37]	CNN	VAE-GAN	94%	92.5%	90.9%	-	SGCC
Our system	**CNN**	**ADASYN**	**97.22%**	**96.8%**	**99.9%**	**97.22%**	**SGCC**

1.4.2 Power Factor Correction and Monitoring Systems

In November 2018, Gunawan et al [38] conducted experiments to enhance the accuracy of measurement in the power factor meter. They achieved this by incorporating a Secure Digital (SD) card to store the collected data, which were then sent to MATLAB's client interface for observation. Additionally, they utilized a smart meter with an Internet of Things (IoT) framework, featuring local storage. The system facilitated automatic power factor improvement via parallel capacitors.

Taye (2018) [39] developed and simulated an automatic power factor correction prototype using an Arduino as a microcontroller. This system reduced costs and proved beneficial for consumers as it improved the industry power factor from 0.66 to 0.92.

In July 2019, Bhagavathy et al [40] developed an Automatic Power Factor Corrector Panel (APFC) using a low-cost microcontroller named PIC 16F877A. If the capacitor bank failings to compensate for reactive power, the user will be alert through developed Android mobile apps. Utilizing the Internet of Things (IoT), the clients can monitor the current state of the panel through Android apps. It is possible to predict future electricity consumption by using cloud analytics.

A. Barhate et al (2019) [41] built the automated power factor control (APFC) device, which can automatically boost a system's power factor. The design included IoT technology for the surveillance of energy usage. The APFC device determines the capacitor value required by an inductive load in the system for compensating lagging power factor utilizing the capacitor bank.

In February 2020, Teddy et al [42] designed a power factor meter utilizing IoT. The system provides data logging, monitoring, and power factor adjustment. Wi-Fi module and Arduino allowed access to the device and transmitted the data to a server for logging, storage, and more analysis. Also, the data is recorded on an SD card and is accessible via computer using the Matlab graphical user interface GUI.

In October 2021, Mafaz et al [43] designed a monitoring system that automatically corrects the power factor using cloud computing (platform as a service) and neural networks. He uses a private centralized cloud processing core (Raspberry Pi) to execute and manage the algorithms. It also helps in hosting, accessible on request anytime and anywhere while the internet is available. The neural network algorithm was employed to calculate the appropriate capacitance value to correct the power factor. This design achieved accurate results and reduced the cost of the bill, in addition reducing the number of devices used, because it supports several home improvements in one centralized core (Raspberry Pi).

In September 2021, Md. Abdullah et al [3] designed a system that continuously analyzes several aspects of an induction motor and updates the data on a webpage using the IoT concept. Additionally, they built an Automatic Power Factor Corrector (APFC) unit utilizing an Arduino as a microcontroller. The power factor component improved from 0.76 to 0.97 under the testing load condition. The average power consumption savings were around 1.7% of the intended load.

Nugroho et al (2021) [44] designed a system that uses neural networks to measure/improve power factors automatically. Based on IoT, clients can monitor electrical parameters online through a web page. He enhanced the power factor by 97.8% for the trained load and 94.8% for the untrained load. The use of the MQTT protocol can improve data transfer efficiency.

M. Pampalle et al (2022) [2] built an automated power factor correction device (APFC) utilizing Arduino-Uno. Based on the Internet of Things (IoT), the APFC unit constantly measures and compensates for power factors automatically. If the APFC unit fails to correct the power factor, an alert message will be sent to the industry manager via the IoT using the If This, Then That (IFTTT) server and the Wi-Fi module to do the necessary actions.

Madhiarasan (2023) [45] develop an automatic power factor correction (APFC) system that monitors energy consumption and improves power factor automatically. They implemented an open-source energy monitoring library and conducted hardware experiments using a capacitor bank and IoT technology for energy monitoring and power factor correction. They also created a mobile application for convenient power monitoring. The developed Raspberry Pi-based system successfully improved the power factor without human intervention by switching the capacitor bank. The proposed design is compact, simple, and easy to implement.

Table 1.3 provides a comparison of Power Factor Correction (PFC) systems from various studies and understand their key characteristics, including the hardware components, communication methods, sensors, user interfaces, PFC techniques, and system types used in each study.

Table 1.3: Comparison of PFC Systems Characteristics in Literature Review with Our System

Ref	Core hardware device	Com-board	Com. Scheme	Sensors	UI	PFC method	System type	Accuracy
[38]	Arduino	Arduino	Wi-Fi	CT, VT	LCD	Computational	Cloud storage	-
[39]	Arduino	Local	Wi-Fi	CT, VT	LCD	Computational	IoT	-
[40]	PIC 16F877A	Wi-Fi module	Wi-Fi	CT, VT	android app + LCD	Computational	IoT	-
[41]	Arduino Uno	ESP8266	Wi-Fi	CT, VT	Web page	Computational	Cloud server	-
[42]	Arduino	Wi-Fi module	Wi-Fi	CT, VT	LCD	Computational	Cloud storage	-
[43]	Raspberry Pi	NodeMCU	Wi-Fi	PZEM-004T	Web page	Neural network	Cloud host and storage	-
[3]	Arduino	ESP8266	Wi-Fi	CT, VT	Desktop app + web app	Computational	IoT	-
[44]	STM32	ESP8266	Wi-Fi	PZEM-004T	Web page	Neural network	Local server	94.8%
[2]	Arduino Uno	ESP8266	Wi-Fi	CT, VT, IR sensor	LCD + Mobile app	Computational	IoT	-
[45]	Raspberry Pi	Raspberry Pi	Wi-Fi	CT, VT	Mobile app	Computational	IoT	-
Our system	**Arduino mega Wi-Fi**	**Arduino mega Wi-Fi**	**Wi-Fi**	**PZEM-004T**	**Web app + LCD**	**Random Forest algorithm**	**Cloud server**	**97.93%**

1.5 Thesis Objectives

The objectives of designing this system are:

1. The system aims to enable real-time monitoring and accurately identify instances of electricity theft or unauthorized activities to reduce losses.
2. The system seeks to implement techniques for optimizing power factors to improve power quality, reduce energy wastage, and enhance overall system efficiency.
3. The systems are validated for effectiveness and efficiency by implementing them on a prototype Advanced Metering Infrastructure (AMI) system with smart meters and evaluating their performance through real-world testing.

1.6 Thesis Contributions

Developing a cloud computing system for electricity larceny detection and power factor correction contributes to:

1. Enhanced Efficiency: The system helps identify instances of electricity larceny, reducing NTLs and leading to improve grid performance.
2. Improved Power Quality: Power factor correction minimizes reactive power, optimizing energy utilization and reducing TLs leading to reduce power wastage.
3. Real-time Monitoring: Cloud-based solutions allow for remote monitoring of the grid, enabling timely detection of issues.
4. Data Analysis: The system facilitates efficient data collection and analysis, in addition to enabling the utilization of advanced analytics techniques, such as machine learning and deep learning, to extract valuable information from vast amounts of data.

5. Cost Savings: By detecting electricity larceny and optimizing the power factor, the system helps reduce financial losses and promotes efficient energy usage.

1.7 Thesis Outlines

The Outlines for this thesis includes the following chapters:

- Chapter two
  - It explained the theoretical knowledge pertaining to designed systems for power factor correction and electric energy larceny with cloud computing.
- Chapter three
  - It described the design architecture and operating aspects of all system components, with the appearance of schematics of the block systems.
- Chapter four
  - It detailed the most significant results of the system and discussed them comprehensively.
- Chapter five
  - It offered the findings, conclusions, and recommendations for future research that can be conducted under the opinion of the researcher.

2

CHAPTER TWO

THEORETICAL BACKGROUN

Chapter Two

Theoretical Background

2.1 Introduction

This chapter explores the theoretical foundations and essential concepts associated with cloud computing systems for data collection and processing. It delves into understanding fundamental principles and theoretical concepts that form the basis of the proposed techniques for power monitoring, load pattern analysis for larceny detection, and power factor correction with intelligence cloud computing. It will discuss the hardware and software components necessary for the operation of such systems.

2.2 Advanced Metering Infrastructure (AMI)

Advanced Metering infrastructure (AMI) is the more significant part of the smart grid infrastructure. AMI consists of three components: Smart Meters, Communication Networks, and Power Management Systems. AMI allowss a bidirectional flow of information for providers and clients and remote collection of data on energy usage, which improve the efficiency and reliability of the power grid and also provides more detailed information about electricity consumption. AMI technology can help utilities better manage distribution networks, detect and respond to outages, and reduce costs associated with meter readings and customer service [46],[47]. Figure 2.1 illustrates the AMI components.

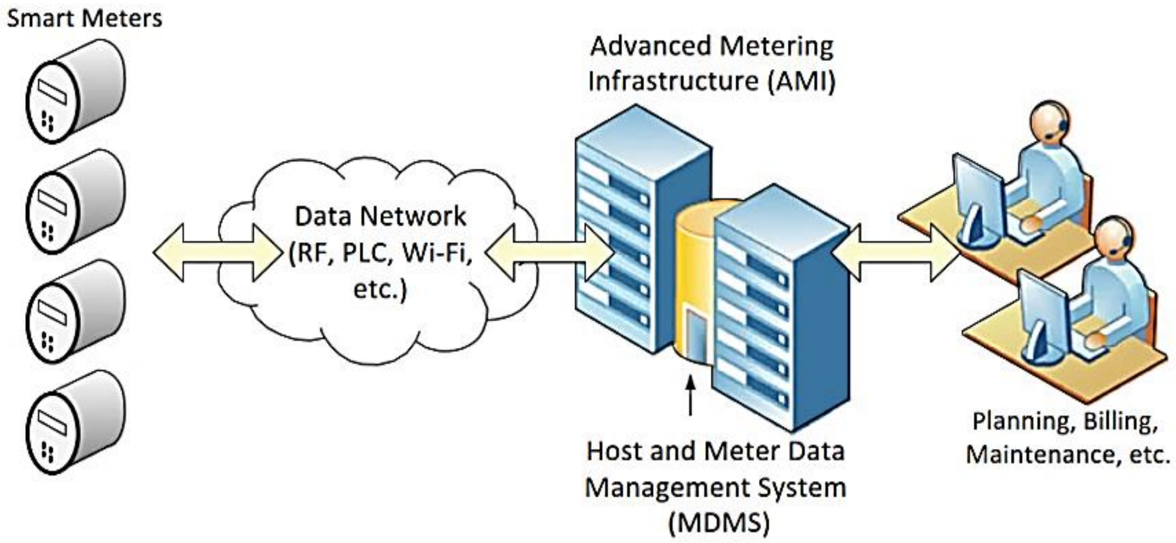

Smart Meters
Advanced Metering Infrastructure (AMI)
Data Network (RF, PLC, Wi-Fi, etc.)
Host and Meter Data Management System (MDMS)
Planning, Billing, Maintenance, etc.

Figure 2.1: AMI components[48].

2.3 Cloud computing

In recent years, the power systems industry has undergone a transformation from offline to online procedures, with cloud computing (CC) emerging as a viable solution to meet the increasing demands for resource usage and storage capacity [49]. CC provides a scalable model that integrates computer resources and virtualization, allowing for the internal networking of power system resources, often referred to as 'clouding' [50]. This integration of CC offers a wide range of applications in power system analysis, including power flow computation, system monitoring, scheduling, and reliability analysis. Moreover, CC provides customers with diverse capabilities, encompassing hardware, databases, storage, networks, and software applications [49].

Given the large volume of data obtained from sensors, CC presents an ideal solution for processing big data, making it particularly suitable for condition monitoring systems and data-driven and model-based applications [51]. Additionally, the cloud platform can utilize various communication protocols and techniques to gather diverse measurements from the grid [52].

Cloud computing (CC) has an important advantage in that it can combine different types of computing resources, such as physical servers, storage devices, networking equipment, and virtual machines, due to the transformative nature of virtualization. This has revolutionized the approach to delivering IT services, reducing infrastructure demands and simplifying the introduction of IT services [53]. CC services are web-based and can be easily launched with minimal effort, as they are designed for easy maintenance and cost-effectiveness [54].

Beyond the power systems industry, CC offers significant benefits to other fields such as healthcare organizations, home automation, and innovative home systems [55].

2.3.1 Cloud Computing Services

CC services refer to the various types of services and resources that are provided over the internet by cloud service providers. These services enable users to access and utilize computing resources, without the need for on-premises infrastructure. The main types of CC services are[56] :

2.3.1.1 SaaS (Software as a Service)

SaaS is the first layer of Cloud Computing service models. It allows multiple users to access and use software applications and data through a platform without needing to install additional hardware or software on personal computers[57]. Instead, it is accessed and provided through a vendor or service provider, typically via the Internet. SaaS is often associated with a pay-as-you-go subscription model [56]. An example of this would be Google Docs.

2.3.1.2 PaaS (Platform as a Service)

PaaS is the middleware of cloud computing models that offers a platform for users to develop, run, and manage web apps and services without having to worry about the infrastructure. Developers can concentrate on designing and implementing their apps while the provider handles the underlying infrastructure, including servers, storage devices, and networking equipment [57]. PaaS can help organizations avoid the costs and complexity associated with scaling their infrastructure [56]. An example of a PaaS model is Structured Query Language (SQL) database.

2.3.1.3 IaaS (Infrastructure as a Service)

IaaS is the third and most comprehensive cloud computing model. In this model, IaaS provides virtualized computing resources servers, storage devices, networking equipment, and software applications. Some providers offer additional services, such as load balancing, monitoring, and security. These services can be accessed through Application Programming Interfaces (APIs), web portals, or command-line interfaces. Organizations can choose the resources they need, such as CPU, memory, and storage, and only pay for what they use. This can help organizations save money [57]. An example of an IaaS model is Amazon Web Services.

2.3.2 Cloud Computing Deployment

Deployment models in (CC) are used to determine how cloud services are deployed and delivered to users. The most common deployment models in CC are [58]:

2.3.2.1 Public Cloud Deployment Model

This deployment model is a widely common model in CC. In this model, the providers offer a range of computing resources to users over the Internet. Users can access and utilize these resources based on their specific needs and are charged on a pay-per-use basis. Unlike private or hybrid clouds, users do not need to own or manage the underlying infrastructure. Some examples of this model providers are Amazon Web Services and Microsoft Azure. The advantages of this model include scalability, flexibility, and cost-effectiveness [10].

2.3.2.2 Private Cloud Deployment Model

This deployment model is used by organizations that need more control over their computing resources. In this model, computing resources are not shared with other organizations and are exclusively allocated to a single organization. This deployment model can be hosted by a third-party provider or be on-premises. Private cloud deployments are often used in industries where security requirements are strict. The advantages of this model include improved security, privacy, and control over resources [9].

2.3.2.3 Hybrid Cloud Deployment Model

This deployment model combines the public and private cloud models. In this model, an organization can use both public and private clouds, depending on their specific needs and workload requirements. Hybrid cloud deployments can be integrated through a single management platform, allowing for seamless data transfer and workload portability. The advantages of this model include flexibility, cost-effectiveness, and improved security [9].

2.3.2.4 Community Cloud Deployment Model

This deployment model is specifically designed for organizations that share similar requirements and objectives, such as research institutions or government agencies. In this model, cloud infrastructure is established and shared among multiple organizations. This model provides the advantages of both public and private clouds, such as scalability and control, while also fostering collaboration and resource sharing. The advantages of this model include improved collaboration, cost-sharing, and increased security [10]. Figure 2.2 showed the CC Deployment Models and Services.

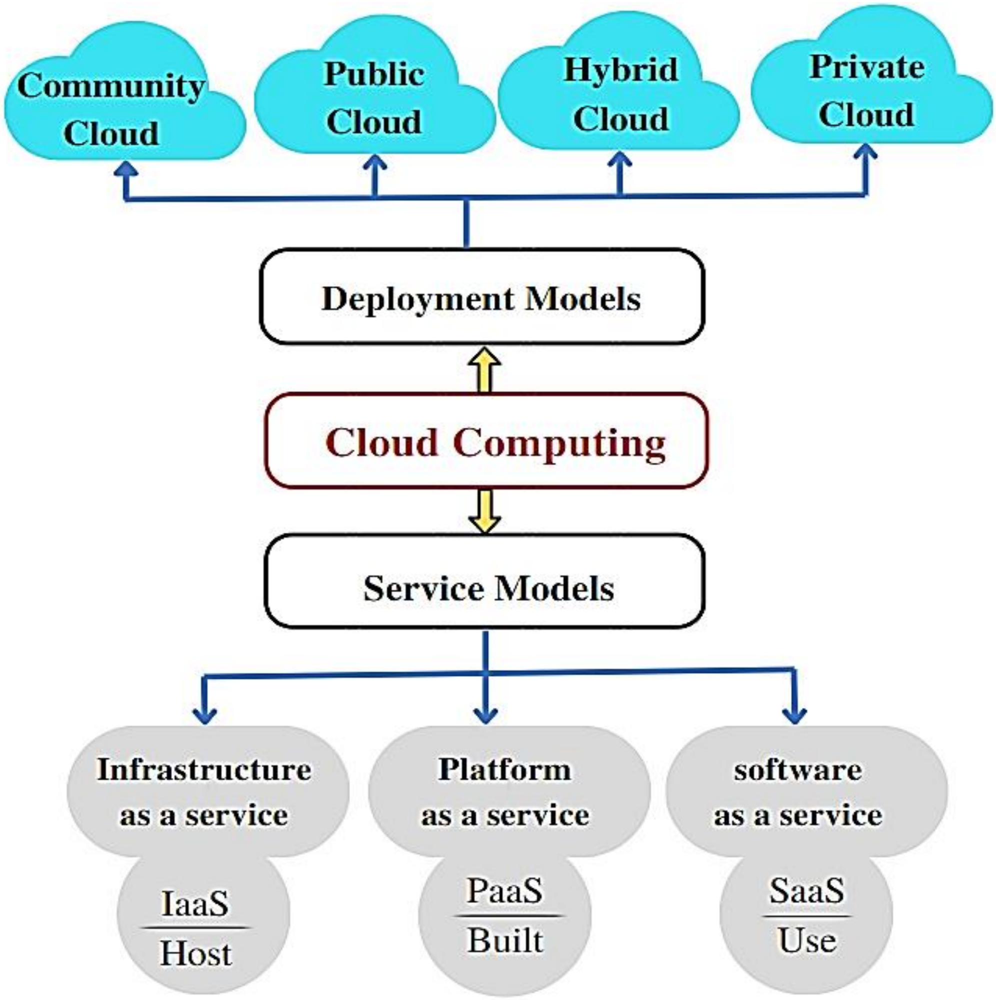


Figure 2.2: Cloud computing deployment models and services.

2.3.5 IoT and Cloud Computing

The Internet of Things (IoT) has revolutionized the way we connect and interact with devices and sensors, opening up new possibilities for innovation. To support IoT applications, a reliable, flexible, and agile platform is required. One such platform that supports IoT is cloud computing [60]. Cloud computing can be defined as a framework that provides users with subscription-based access to computer resources and private network storage space. This allows users to access their personal information from anywhere and at any time, eliminating the need for physical presence at data storage devices [61].

The Cloud-IoT architecture involves three layers: the sensing layer, the network layer, and the application layer, which are interconnected. The sensing layer comprises objects that can read and collect data through various IoT systems, while the network layer facilitates the communication between these objects and the Cloud. The application layer is responsible for sensing data and sending requests to the Cloud for analyses and determining sensor data results[62]. The Cloud-IoT architecture is depicted in Figure 2.3.

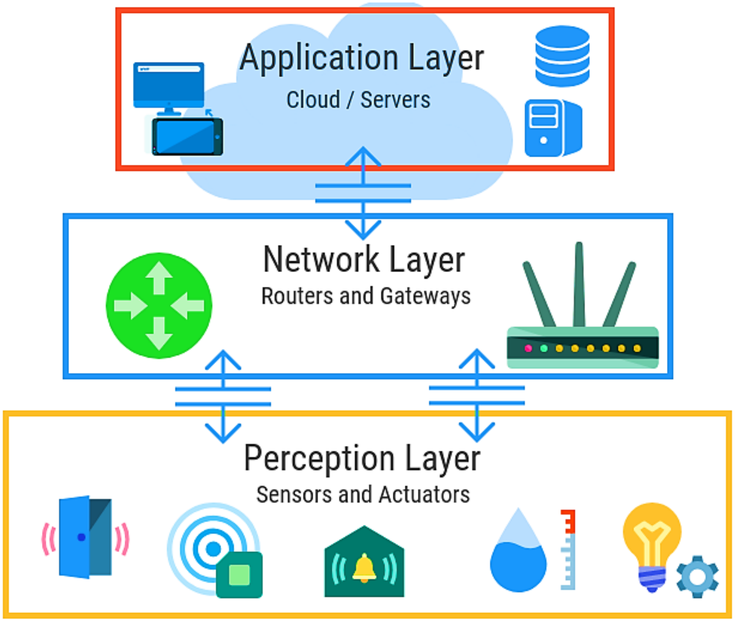


Figure 2.3: The Architecture of Cloud-IoT [62].

2.3.6 Communication Protocols for Cloud

Several communication protocols can be used in cloud computing to ensure efficient and secure communication between devices, servers, and applications [63]. Some of the most commonly used protocols for cloud communication are:

2.3.6.1 HTTP (Hypertext Transfer Protocol) and HTTPS

HTTP is a protocol used for communication between clients (web browsers) and servers in web architecture. It follows a request-response model in which, a client sends a request to a server, which processes the request and sends a response back to the client. This exchange of requests and responses occurs over a network [64], as depicted in Figure 2.4, representing the HTTP request/response model.

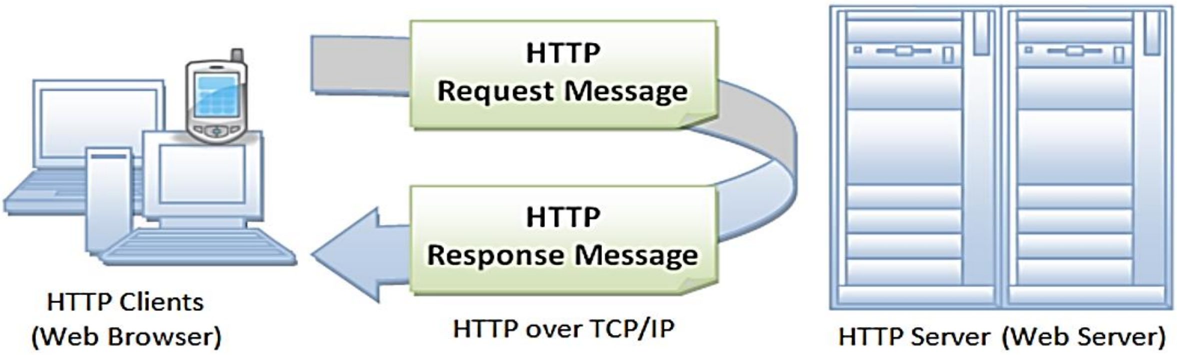


Figure 2.4: HTTP request/response model [66].

HTTPS, or HTTP over SSL/TLS, is a secure communication protocol that builds upon HTTP. It adds an extra layer of encryption to ensure the confidentiality and integrity of data transmitted between a web server and a client. SSL (Secure Socket Layer) is used as a sublayer in HTTPS to establish an encrypted connection. By encrypting the data, HTTPS prevents unauthorized access and eavesdropping. Certificates are utilized in HTTPS to verify the security level of a web application and ensure the authenticity of the server [64]. Figure 2.5. shows the difference between HTTP & HTTPS.

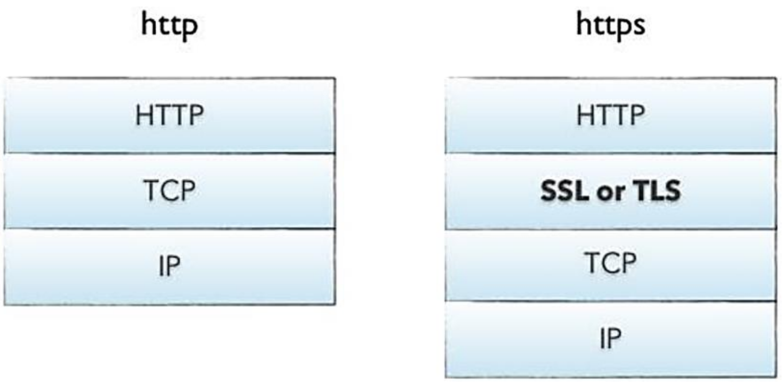


Figure 2.5: Difference between HTTP & HTTPS [64].

2.3.6.2 MQTT

MQTT (Message Queuing Telemetry Transport) is a protocol specifically designed for communication between devices with limited resources and operating in challenging network conditions, such as low bandwidth and high latency. It addresses the need for efficient and reliable messaging in IoT applications. In the MQTT protocol, communication follows the publish-subscribe paradigm. MQTT was developed by IBM and involves three main components: publishers, subscribers, and brokers (or servers).

In MQTT, clients can act as publishers, responsible for sending messages, or subscribers, which receive messages. Publishers send messages to a specific topic, and subscribers can choose to receive messages from specific topics they are interested in. Multiple clients can subscribe to the same topic, and whenever a new message is published on that topic, the broker distributes it to all subscribed clients. Figure 2.6 represents the MQTT protocol model, illustrating the interaction between publishers, subscribers, and brokers.

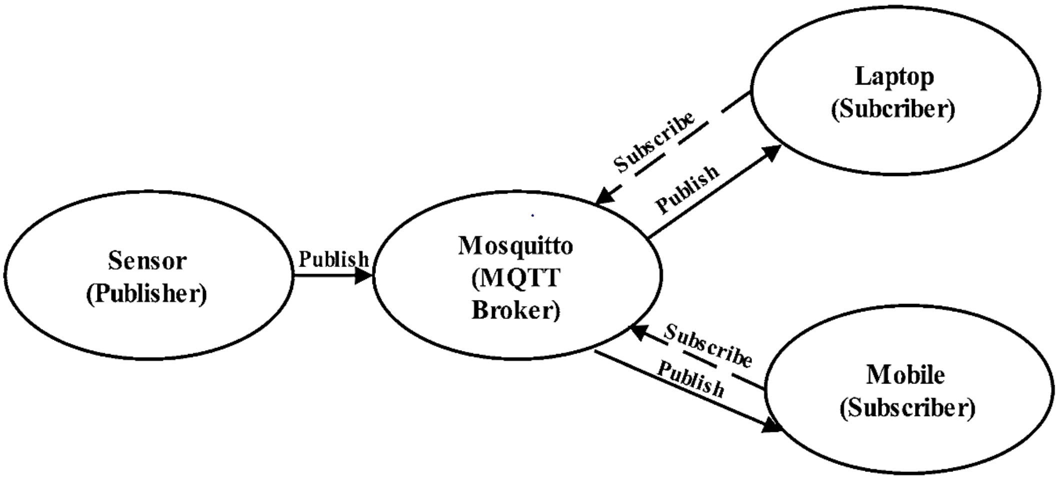


Figure 2.6: Publisher/subscriber model for MQTT protocol [65].

2.3.7 Cloud Server

A cloud server refers to a strong infrastructure, either physical or virtual, that enables the delivery of applications, processes information, and provides data storage. Some Cloud servers utilize virtualization software to divide a single physical server into multiple virtual servers. Cloud service providers follow an infrastructure as a service (IaaS) model to offer customers access to virtual or bare metal servers [66].

2.3.7.2 Why choose cloud servers?

1. **Cost-effectiveness:** Organizations can optimize costs by paying only for the resources they need, eliminating the expense of maintaining physical server hardware.
2. **Scalability:** Cloud servers offer the ability to scale computing and storage resources based on fluctuating demands, making them ideal for organizations with varying workloads.
3. **Integration:** Cloud servers ensure seamless communication and swift deployment through networked connections, offering comprehensive control via a unified interface [66].

2.4 Tools for Smart Environment Technologies

Several technologies related to the smart environment, including the following [67]:

2.4.1 Internet Protocol (IP)

An IP address is a unique identifier assigned to networking devices that use it for communication. It serves two main purposes: host or network identification and geographic location. There are two versions of IP addresses: IPv4 and IPv6. Each version has its own set of addresses, which are different from one another [67]. IPv4 is the original version of IP and has been in use for several decades. However, as the number of internet-connected devices has grown exponentially, the available IPv4 address space has been exhausted. IPv6 was developed as a replacement for IPv4 **[68].**

2.4.2 Wireless Fidelity (Wi-Fi)

Wi-Fi is a wireless technology that enables devices to connect to a network without the need for physical cables. Wi-Fi networks use radio waves to transmit data over the air and can be set up in a variety of configurations. Wi-Fi networks operate according to specific IEEE 802.11 standards and can be secured using a variety of security protocols. The most common frequencies used in Wi-Fi are 2.4 GHz and 5 GHz [67].

2.4.3 The Transmission Control Protocol (TCP)

TCP is a reliable, connection-oriented protocol for error-free and ordered data transmission between two computers. It divides the data stream into segments, assigning a unique sequence number to ensure correct ordering and reliability. Upon receiving the segments in the correct order, the receiver sends a cumulative acknowledgment back to the sender, confirming receipt of all data up to that sequence number.

If a segment arrives out of order, the receiver sends an acknowledgment of the expected sequence number of the missing segment. If no acknowledgment is received within a certain time, the sender retransmits the unacknowledged segments. This ensures a robust and error-free communication channel between the two computers [69]. Figure 2.7. Relationship between layers, TCP,and IP.

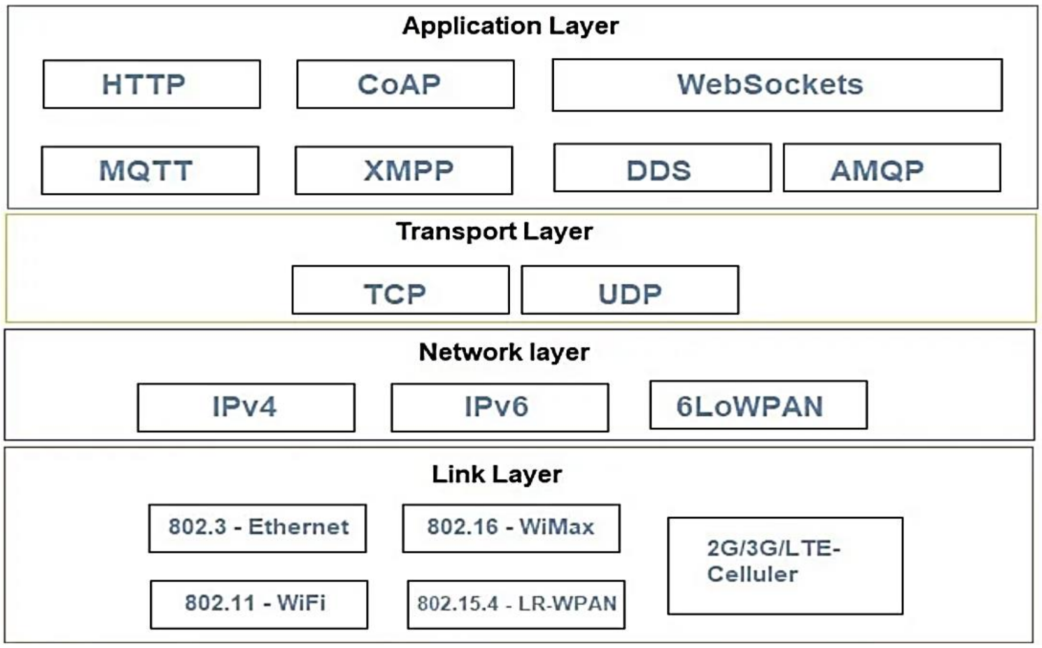


Figure 2.7: TCP/IP Layer [67].

2.4.4 Sensor Networks

Sensor networks are an important component of the Internet of Things (IoT) ecosystem. They are made up of a large number of small, low-cost sensors that are connected to each other and to the Internet. These sensors are able to collect data on a wide range of environmental conditions, including temperature, humidity, light levels, and air quality. One of the key benefits of sensor networks is their ability to provide real-time data on environmental conditions.

This data can be used to monitor and manage a wide range of systems and processes [67].

2.5 Smart Metering

Smart metering is a modern technology designed to improve the traditional metering process. It involves the use of advanced metering infrastructure (AMI) technology, which allows for two-way communication between the meter and the energy provider. Smart meters are connected to a communication network, such as Bluetooth or ZigBee, and have a chip that contains software to determine the functions, applications, and scope of the meter [70].

The main advantage of smart metering is the large amount of data that can be collected and managed by the Data Management System. This data can be used to improve the distribution of energy, reduce energy consumption, and provide more accurate billing. Smart meters can also provide users with a wide range of features, such as real-time energy consumption data and remote control of energy usage [71]. Figure 2.8 comprises the framework for metering and communication

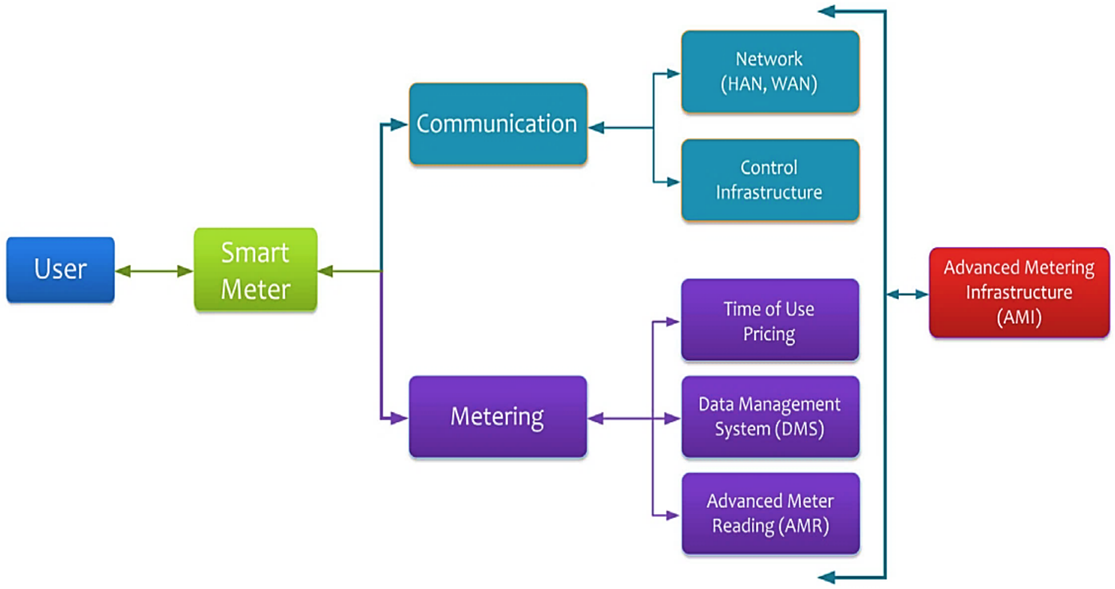


Figure 2.8: Metering and Communication Framework [70]

2.6 Smart Meter Energy Sensors and Wireless

Smart meters equipped with sensors and wireless communication technologies are critical components of modern energy grids. They provide real-time data on energy consumption, enable more efficient energy management, and increase grid reliability. As the technology behind smart meters continues to evolve, we can expect to see even more innovative and useful features added to these essential devices. Smart meters may have current and voltage, accelerometers, Hall sensors, anisotropic magnetoresistance sensors, and passive infrared sensors. These sensors enable the meter to add new functions [72]. Some of the essential tools and sensors for the prototype smart meter are

2.6.1 PZEM-004T Module

The energy sensor (PZEM-004T-100A) is used to measure AC voltage (V), AC current (I), active power (P), power factor (Pf), frequency (f), and active energy (E). This sensor has the ability to measure and store the above measurements and display data through the interfacing Transistor-Transistor Logic (TTL) because this sensor has no display unit. This sensor uses an alternative current (AC) power source and measures the above parameters by passing the load wire through the current transformer (CT).

The reasons for using this sensor are its high accuracy, low cost, ease of programmable, and low power consumption which means long battery life for the whole system and it is easy to communicate with serial communication by using only two terminals (Tx and Rx) [72]. Figure 2.9. shows the shapes of the energy sensor. [73].

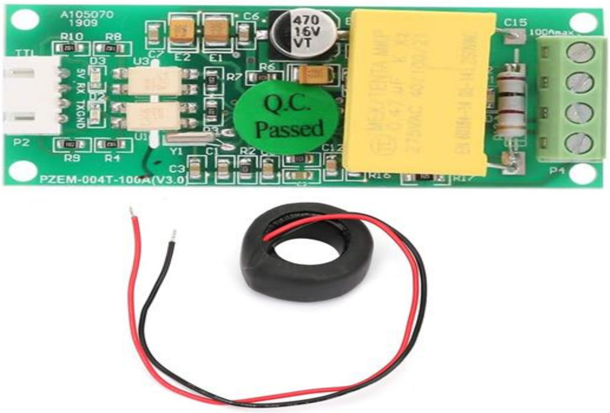


Figure 2.9: PZEM-004T Module with Current Transformer [73]

2.6.2 Arduino Mega WiFi R3 (ATMEGA2560 + ESP8266)

The ARDUINO MEGA Wi-Fi R3 is a powerful microcontroller board that is equipped with an ATMEGA2560 processor and an ESP8266 Wi-Fi module, offering 32 MB of memory. These two modules can be used together or independently, as they have their separate pin headers. It has 54 digital input/output pins (of which 14 can be used as Pulse Width Modulation (PWM) outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. This board is designed to provide a convenient solution for developing new projects that require the capabilities of both the Mega R3 ATmega2560 and WiFi ESP8266 modules [73]. Figure 2.10. shown the shape of Arduino mega Wi-Fi R3 board

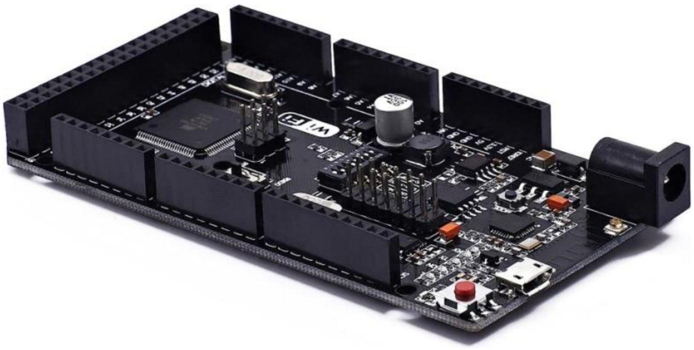


Figure 2.10: Arduino Mega Wi-Fi R3 board [73].

2.7 Technical and Non-Technical Electric Losses

Electric losses refer to the energy that is dissipated or lost during the transmission, distribution, and utilization of electricity. These losses occur in various places of the electrical system such as transformers, transmission lines, and other equipment, and they represent a significant economic and environmental impact. Electric losses in power systems can be categorized following their source: technical or non-technical losses [74].

2.7.1 Technical Losses (TLs)

TLs in power systems are a common occurrence and are caused by several factors. These include resistive losses due to internal electrical resistance, losses in transformers and transmission lines, and losses in measurement systems. These losses are caused by the flow of current in the electrical network and result in power dissipation.

1. Copper losses: these losses are due to the resistance of the conductors and result in heat generation.
2. Dielectric losses occur due to the heat generated by the dielectric material located between the conductors.
3. Induction and radiation losses occur due to the electromagnetic fields generated around conductors.

If the power system consists of known loads, it is possible to calculate and minimize Technical Losses (TLs). Various techniques can be utilized to decrease TLs, including the use of efficient transformers and conductors, load balancing, voltage regulation, and correcting the power factor [74].

2.7.2 Non-Technical Losses (NTLs)

NTLs in the power system refer to electricity losses that occur due to theft, meter tampering, billing and collection inefficiencies, and errors in energy accounting. These losses are often caused by human factors, including dishonesty, corruption, and lack of awareness of the importance of paying for electricity [74].

1. **Meter tampering:** Consumers bypass or manipulate the meter to reduce or avoid paying for their electricity consumption.
2. **Illegal connections:** This involves unauthorized connections to the electricity supply.
3. **Billing and collection inefficiencies:** Electricity bills may be inaccurate or not delivered on time.
4. **unpaid bills:** Some consumers may intentionally delay or avoid paying their bills.

2.8 Power Factor Correction (PFC)

PFC is a technique used in electrical power systems to enhance the power factor by the system. The power factor indicates how efficiently the electrical power is being utilized. A low power factor indicates there is a high quantity of reactive power (Q) in the circuit. Reactive power, often measured in volt-amperes reactive (VAR), represents the power that oscillates without performing useful work in the circuit. It is essential for devices like motors and transformers but doesn't contribute to actual work, making the power system less efficient. This condition can lead to increased energy consumption, higher electricity bills, and reduced efficiency [40].

A power factor of 1 means that all the power being supplied to the system is effectively used, while a power factor below 1 indicates some of the power is lost. This causes increased bills and less efficiency [42]. Power factor correction is achieved by adding power factor correction devices, such as most common synchronous condensers, phase advancers, and static capacitor banks, to the electrical system. These devices help to reduce the amount of reactive power in the circuit, which in turn improves the power factor of the system [44].

The PFC can be enhanced by adding capacitors to the electrical system. The amount of capacitors required for PFC depends on the load characteristics of the system

The power factor correction factor (PF) is defined as the ratio of real power (P) in watts to apparent power (S) in volt-amperes within the circuit [42], which is explained as:

$$PF = \frac{Real\ Power(kw)}{Apparent\ Power(kVA)} = \frac{P}{S} = \frac{VI\cos\theta}{VI} = \cos\theta \tag{2.1}$$

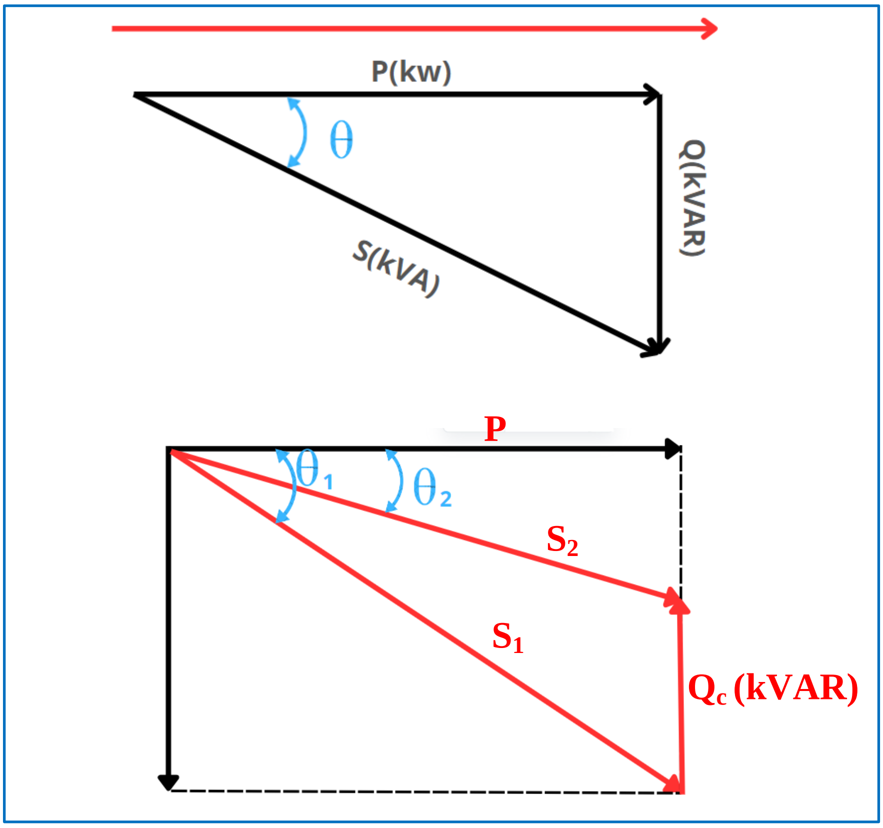


Figure 2.11: The power tringle

Equations (2.2), (2.3) and (2.4) can be used to calculate the amount of reactive power required for PFC:

$$P = S.\cos\theta\,, \qquad Q = S\,.\sin\theta, \qquad \tan\theta = \frac{Q}{P} \tag{2.2}$$

$$Q_c = P.\left[\tan\left(\cos^{-1}(PF_1)\right) - \tan\left(\cos^{-1}(PF_2)\right)\right] \tag{2.3}$$

$$Q_c = P.(\tan\theta_1 - \tan\theta_2) \tag{2.4}$$

Where Q_c represents the reactive power required for PFC, PF_1 and PF_2 are the initial and desired power factors, respectively. θ_1 and θ_2 are the corresponding phase angles [42].

The required capacitance size can then be calculated using equation (2.4):

$$c(F) = \frac{Q_c}{2\pi f v^2} \tag{2.4}$$

Where v is the voltage and f is the frequency [42]. The advantages of PFC in electrical systems include [43]:

1. **Improved efficiency:** PFC reduces the amount of reactive power supplied, reducing the load on electrical stations and reducing losses.
2. **Increased capacity:** By minimizing the requirement for reactive power supply, the capacity of the electrical system can increase.
3. **Compliance with regulations**: Many countries have regulations that require a minimum power factor for commercial and industrial electrical systems. PFC helps to avoid penalties and fines.
4. **Longer equipment lifespan**: PFC can reduce the stress on electrical equipment, leading to a longer lifespan.

2.9 Electric Energy Larceny

Electricity larceny is a critical issue that many power companies face, as it results in significant losses and unsustainable infrastructure. This problem arises when some consumers receive excessive bills, while others bypass their prepaid meters or directly tap into the distribution network, resulting in lost revenue for power companies [70]. Due to its multifaceted nature, power theft is a common occurrence in many countries, with a large amount of energy lost every year from low-voltage distribution networks. The sheer number of distributed endpoints in publicly accessible areas makes it challenging for electric distribution companies to prevent pilfering or fraud [71]. Therefore, targeted strategies are necessary to decrease energy losses on low-voltage grids, with AMI systems playing a critical role in combating non-technical losses associated with power theft. Implementing these measures is essential for improving the sustainability of the electricity industry and ensuring fair and equitable pricing for all consumers [72]. Electricity theft can occur in various forms involves [70]:

1. **fraud**: when a consumer deliberately deceives the utility company by energy meter tampering to display lower kWh consumption than the actual usage.
2. **stealing electricity:** where consumers bypass the electricity meter by connecting directly to supply points or distribution lines.
3. Billing irregularities: These irregularities can be unintentional or intentional. Unintentional billing irregularities occur due to system failures, which result in incorrect bills being issued to consumers. On the other hand, intentional irregularities may occur when utility company employees knowingly record incorrect meter readings.
4. **unpaid bills:** where consumers choose not to pay their bills and the outstanding debt is carried by the utility company.

To combat this issue, several technologies have been developed to detect and prevent electricity theft. These technologies can be divided into three categories: network-oriented, data-oriented, and hybrid-oriented.

The network-oriented method detects anomalies by measuring electrical parameters such as voltage, current, power, and frequency using sensors, meters, and other monitoring devices, and detects abnormalities that may indicate theft.

The data-oriented method analyzes consumption to identify abnormal load patterns or discrepancies that may suggest theft using statistical analysis, machine learning, and data mining techniques.

The hybrid-oriented method integrates both methods to improve accuracy and reliability. It combines monitoring devices with consumption to obtain a comprehensive view of the electricity network [74].

2.10 Random Forest (RF)

The RF Algorithm is a Supervised Classification method that aims to achieve higher prediction accuracy by constructing multiple classifiers. It operates by classifying data based on creating a set of classifiers, and the combination of their results predicts the class label for the given data set. This approach is particularly beneficial when dealing with a big quantity of data that may not be effectively classified by a single classifier, as it can lead to reduced accuracy in the results. RF are user-friendly and easily understandable for end-users without a statistical background. They do not need cross-verification and are not prone to overfitting issues [75]. The RF Algorithm involves the following steps for constructing decision trees[76]:

1. Determine the number of training data instances (N) in the sample and identify the total number of attributes (M) in the given input dataset.
2. Define 'm' as the parameter that governs the selection of the next attribute at each tree node. Note that 'm' should be less than M to ensure effective tree construction.

3. Extract training samples and create a tree for each sample, with the possibility of replacement.
4. At each tree node, arbitrarily select 'm' attributes from the dataset.
5. Compute the best split using the 'm' selected input attributes of the sample dataset.
6. Allow each tree to grow without pruning, ensuring full expansion.

The visual representation of the RF is depicted in Figure 2.12.

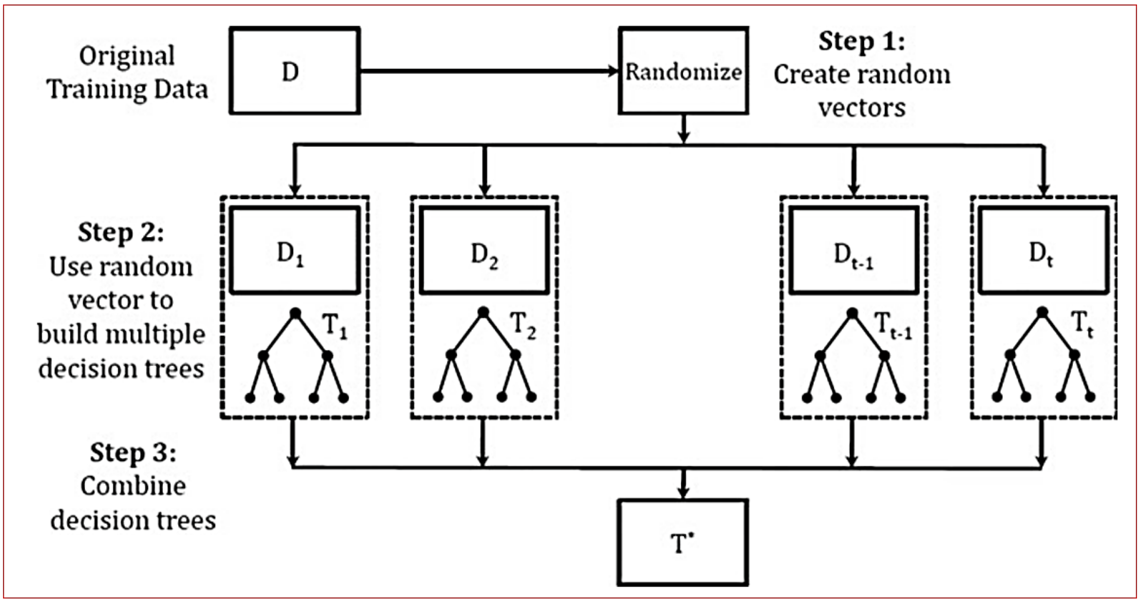

Original Training Data
D
Randomize
Step 1: Create random vectors
Step 2: Use random vector to build multiple decision trees
D_1
D_2
D_{t-1}
D_t
T_1
T_2
T_{t-1}
T_t
Step 3: Combine decision trees
T^*

Figure 2.12: Visual Representation of Random Forest [75]

Random Forest offers several advantages over traditional single classifiers [75].

- It achieves higher accuracy, efficiently handles large databases, and can handle thousands of input variables without compromising performance.
- It provides valuable information about the importance of variables in the classification process, helps identify significant factors, and offers methods to estimate missing data.
- Random Forest can handle missing data effectively, allowing for reliable analysis even in the presence of incomplete information.

- It utilizes prototypes to provide insights or metadata regarding the relationships between different variables, aiding in understanding the dataset structure.
- Random Forest enables the examination of variable interactions, revealing potential complex relationships between input variables.

2.11 Convolutional Neural Networks (CNNs)

CNNs are a specialized type of artificial neural network primarily used for processing data arranged in a grid-like structure, such as images or videos. These networks employ a mathematical operation called convolution, which combines two functions to generate a third function representing the modified version of one function by the other [28]. Compared to traditional classification methods, have the capability to capture and understand more intricate non-linear relationships in data. This allows them to achieve better generalization performance, meaning they can effectively classify new, unseen data accurately [77]. CNN networks' main goal is to extract features using a kernel, which is like a filter that slips over the input and performs a convolution process. This process results in the creation of a feature map. By using different kernels, we can obtain diverse feature maps, we can obtain diverse feature maps, which are then incorporated to create the convolution layer's output [79]. CNN architecture is made up of four types of layers: convolutional layers, pooling layers, a function of activation, and fully connected layers. Convolutional layers are particularly crucial for capturing the essential features of electrical theft patterns [77]. Figure.2.13 shows the general arrangement of these layers that are commonly used to construct a CNN.

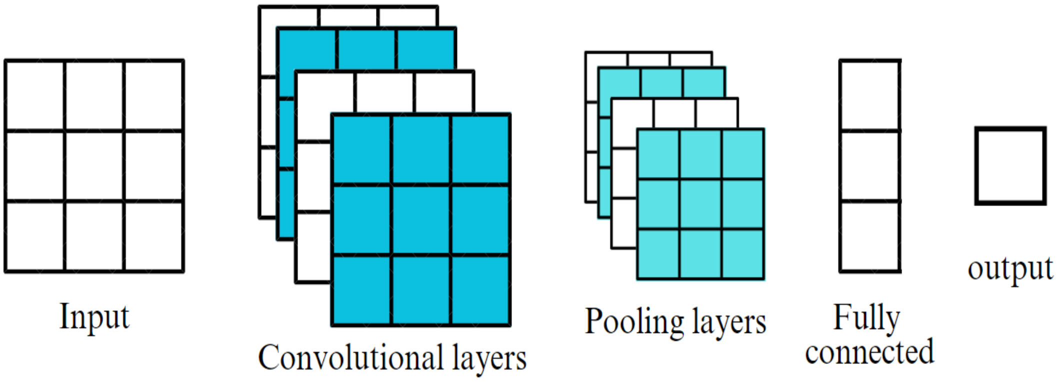


Figure 2.13: The CNN General Architecture

The behavior of the layers in the CNN can be described using equations (2.6) and (2.7) [36].

$$\mathrm{y}_i = f_i(x_i w_i + b_i) \tag{2.6}$$

$$\grave{y} = \max\left(y_{i,j}\right) \tag{2.7}$$

Where each convolution and max-pooling layer takes an input x_i and produces an output y_i and is associated with an activation function f_i. The behavior of the convolution layer is defined by the offset vector b_i and weights w_i , which are used to compute the output y_i through Equation (2.6). The max-pooling layer then takes the highest value of the outputs, producing the output y' through Equation (2.7) [36].

CNNs typically consist of several types of layers, each with a specific function in the network.

2.11.1 Convolutional Layer

In CNN, the convolutional layer is responsible for analyzing input data. It achieves this by applying filters, or kernels, to the input. These filters help extract important features from the input and generate a feature map.

The process begins by sliding the kernel over the input, examining small regions at a time. At each position, a dot product is computed between the kernel and the corresponding values in the input. This computation summarizes the relationship between the kernel and the input data, allowing the network to capture relevant information.

During the training phase of the CNN, the weights in the filters are adjusted gradually. This adjustment is an iterative process that aims to minimize a loss function. By modifying the weights, the CNN can learn to recognize meaningful patterns and features in the input data, leading to improve performance in tasks such as classification or object detection [79].

2.11.2 Pooling Layer

The pooling layer in CNNs plays a vital role in reducing the dimensionality of feature maps while preserving important information [35]. It achieves this by applying pooling operations, such as max pooling, to the input data using sliding filters. Max pooling divides the data into sub-regions and selects the maximum value within each region. This down-sampling process effectively reduces the complexity in the subsequent layers. There are various pooling techniques available, including average pooling, max pooling, average pooling, and min pooling [81].

2.11.3 Activation Functions Layer

Activation functions are an essential component of neural networks as they determine the output of each neuron in the network based on its input. They are used to introduce non-linearity to the output of a layer, enabling the network to learn more complex features. Activation functions take the output of the previous layer and apply a mathematical function to it, resulting in the output of the current layer [28]. Some types of activation functions:

- **The sigmoid activation function** is commonly used in binary classification problems. It maps any input value to a value between 0 and 1, allowing it to be interpreted as a probability or a binary decision [35].
- **The tanh (hyperbolic tangent) activation function** is similar to the sigmoid function but has an output range between -1 and 1. It is often used in regression problems where the target variable can have negative values [35].
- **ReLU (Rectified Linear Unit)** is a popular activation function, particularly in CNNs. It has a simple structure that sets all negative input values to 0 and leaves the positive values unchanged. ReLU is computationally efficient and helps address the vanishing gradient problem that can occur in deep neural networks [35].

Figure.2.14 shows the different types of activation functions.

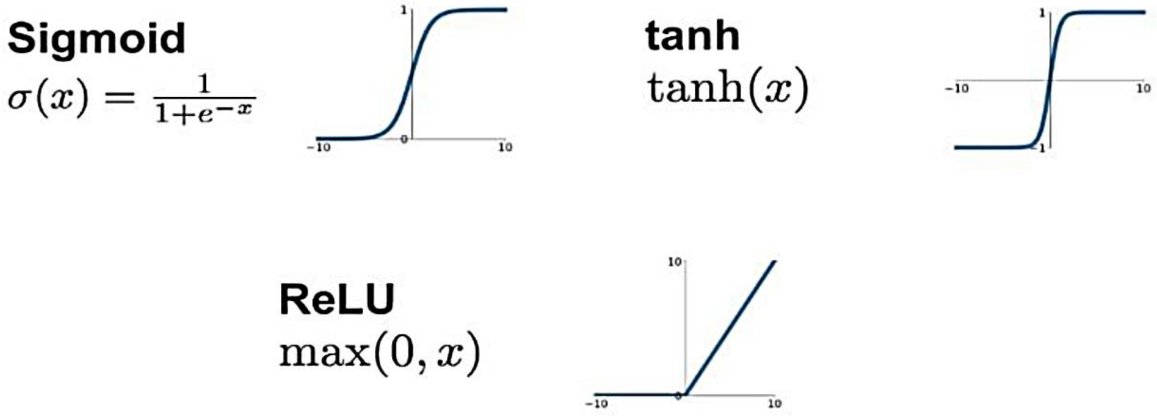


Figure.2.14: Activation Function Types

2.11.4 Fully Connected Layer (FC)

The fully connected (FC) layer in a CNN is the last layer that follows the convolutional and pooling layers. It is also referred to as the dense layer because each neuron in this layer is connected to every neuron in the previous layer. The FC layer takes the output from the previous layers and performs the classification task by learning high-level features and patterns. It combines the extracted features from the previous layers and maps them to the output classes, making the final decision on the classification task [79].

2.12 Data prepossessing

In ML, preprocessing is crucial for extracting hidden knowledge from data. It is necessary to prioritize the preprocessing phase before applying analysis algorithms. Preprocessing addresses various data issues such as inconsistencies, null values, extreme values, and noise. Incomplete data can result from missing attributes or erroneous information. Noise can occur due to data collection problems, transmission issues, or inconsistencies in code naming and assignment policies. Data preprocessing helps handle null values, identify outliers, and resolve inconsistencies. It is recommended to clean the data beforehand to avoid confusion during analysis [80]. Figure.2.15 shows different forms of data preprocessing.

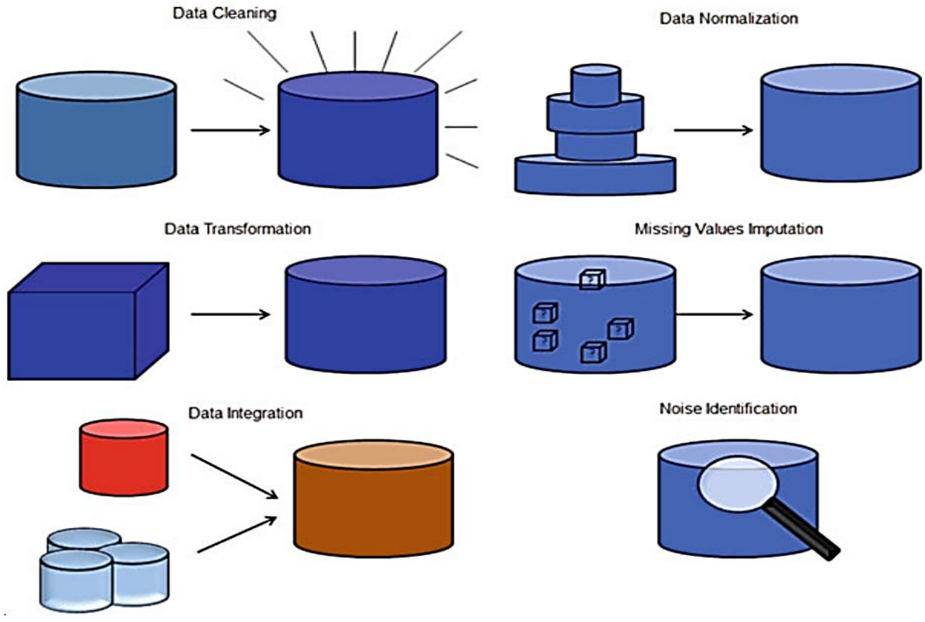


Figure.2.15: Data preprocessing forms [80].

In the context of SG (Smart Grid) technology, advanced sensors, and communication systems generate vast amounts of data about energy consumption. However, this data is often messy and contains errors, including duplicates or noise, which can distort analysis results and hinder interpretation [28].

2.13 Handling Imbalanced Data

refers to techniques and strategies used to address the problem of imbalanced data, where one class (the minority class) is significantly less compared to the other class(es) (the majority class(es)) in the dataset. This is a common problem in many machine learning applications, where the cost of misclassification of the minority class is often higher than that of the majority class [39].

2.13.1 Random Undersampling (RUS)

This technique arbitrarily deletes samples from the majority class until the dataset is balanced [33]. RUS can be effective when the majority class has numerous samples and removing some of them does not significantly impact the overall performance of the model. However, RUS can lead to a loss of information in the majority class [80].

2.13.2 Random Oversampling (ROS)

This technique creates new samples by randomly replicating available samples in the minority class until the number of instances for each class is balanced [33]. ROS is a simple and effective way to handle the problem of imbalanced datasets and can enhance the performance of machine learning models in detecting the minority class. However, it may result in overfitting and reduced generalization of the model if not used appropriately [80].

2.13.3 Synthetic Minority Oversampling Technique (SMOTE)

This technique is used to generate synthetic instances of the minority class by interpolating between existing samples. It starts by selecting a minority class instance and its k nearest neighbors. New samples are then created along the line segments connecting the selected minority sample with its neighbors. This process helps balance the class distribution in the dataset, which can improve the performance of machine learning algorithms [80].

An illustration of the SMOTE process is provided in Figure 2.14. The point $x_i$is chosen as the basis for generating new synthetic data points. Using a distance metric, multiple nearest neighbors (NNs) belonging to the same class (x_{i1} to x_{i4}) are selected from the training set. Subsequently, a randomized interpolation is performed to create new samples, denoted as r_1 to r_4 [80].

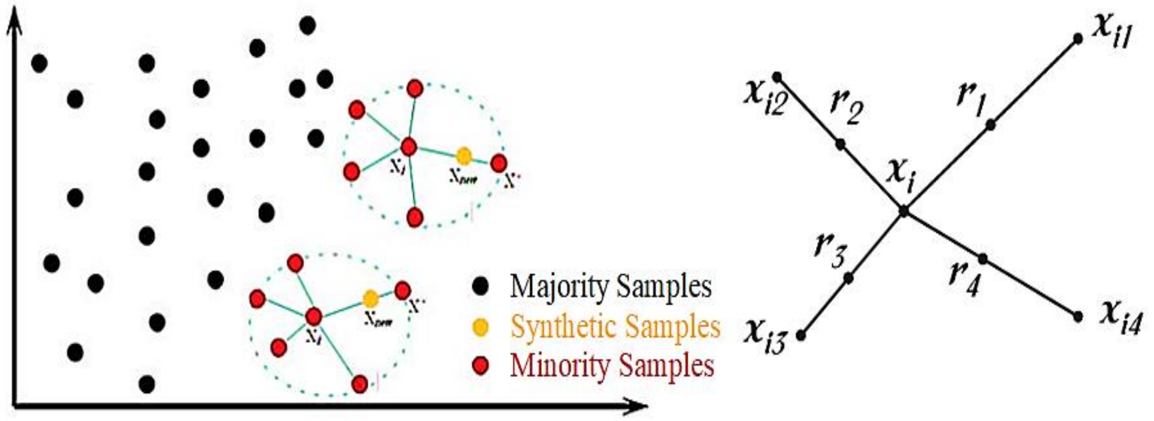


Figure 2.14: Synthetic data point generation by the SMOTE Algorithm [81],[80].

2.13.4 SMOTETomek

SMOTETomek is a hybrid method for dealing with imbalanced datasets that combines SMOTE and Tomek links. Tomek links are pairs of samples from different classes that are very close to each other. These pairs can act as noisy examples, and removing them can increase the margin between the two classes, making them easier to separate [82].

SMOTETomek first applies the SMOTE algorithm to oversample the minority class and then uses Tomek links to identify noisy examples and remove them. The resulting dataset has a balanced number of samples for each class and a larger margin between the two classes [82].

2.13.5 Adaptive Synthetic Sampling (ADASYN)

This technique generates synthetic samples of the minority class in regions of the feature space where the density of minority samples is low while keeping the density of majority samples unchanged. ADASYN is effective when the distribution of data is highly skewed and focuses more on those minority instances that are harder to learn[81]. An illustration of the ADASYN Algorithm is provided in Figure (2.15).

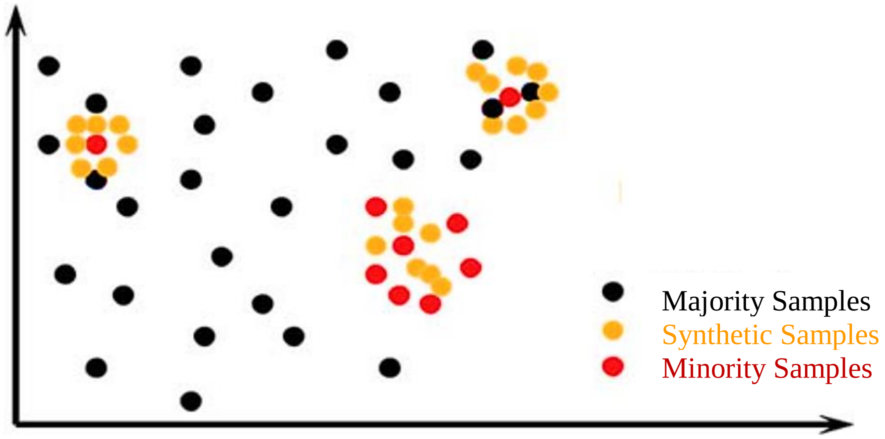


Figure 2.15: Synthetic data point generation by the ADASYN Algorithm [81].

2.14 Evaluation Metrics

Evaluation metrics are used to assess the performance of ML models. There are several evaluation metrics available, each of which is appropriate for different types of problems and use cases. The choice of metric depends on the specific problem being addressed.

2.14.1 Confusion Matrix

A confusion matrix is a representation of a model's predictions and the actual classes of a given dataset. It consists of two dimensions, representing the actual class (rows) and the predicted class (columns). The cells of the matrix are filled based on the classification outcomes: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) [28].

Actual value

		Positive	Negative
Predicated value	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Where:

- TP: The count of instances that are rightly identified as positive.
- FP: The count of instances that are wrongly identified as positive.
- TN: The count of instances that are rightly identified as negative.
- FN: The count of instances that are wrongly identified as negative.

2.14.2 Accuracy metrics (acc)

acc measures the ratio of right predictions over the total number of cases. It is calculated using equation (2.8).

$$\mathrm{acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FN} + \mathrm{FP}} \times 100\% \qquad (2.8)$$

It is worth noting that acc can be misleading in cases where the dataset is unbalanced. Where, a classifier that simply always predicts the majority class could achieve high accuracy, but this would not be a useful [38].

2.14.3 AUC metric

The term AUC stands for "Area Under the Curve," which is a measure of the degree of separability between the positive and negative classes. An AUC score of 1.0 indicates an ideal classifier, while an AUC score of 0.5 indicates a random classifier. Therefore, the higher the AUC value, the better the classifier's capability to accurately distinguish between the positive and negative classes [30].

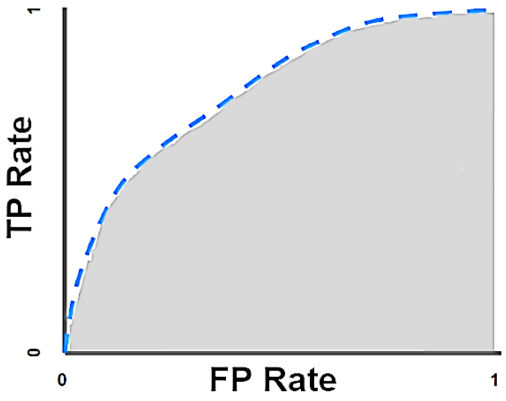


Figure.2.16: Area Under The ROC Curve [35].

2.14.4 F1 score metric

The F1 score is an important performance metric, especially when dealing with imbalanced datasets. The F1 score is computed as the harmonic mean of Precision (P) and Recall (R), taking into account both false positives and false negatives. To calculate the F1 score, Equation (2.9) is used.

$$F_1 = \frac{2\times \mathrm{P.R}}{\mathrm{P+R}} \qquad (2.9)$$

Where Precision (P) measures the accuracy of positive predictions. It is the ratio of true positive predictions to the total number of positive predictions, as shown in Equation (2.9):

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP + FP}} \qquad (2.10)$$

On the other hand, Recall (R) is defined as the ratio of true positive predictions to the total number of actual positive instances, as shown in Equation (2.11):

$$R = \frac{TP}{TP + FN} \tag{2.11}$$

The F1 score ranges from 0 to 1, with 1 being the best possible score [28].

3

CHAPTER THREE

DESIGN AND IMPLEMENTATION

OF

THE PROPOSED SYSTEM

Chapter Three

Design and Implementation of The Proposed System

3.1 Introduction

This chapter presents the design and execution of an intelligent cloud system able to detect electricity larceny in real time, identify anomalous load patterns, and correct the power factor. The system uses a cloud server for storage and analysis of the data collected by the smart meters that were developed. The system implements a hybrid-oriented approach to detect electricity larceny: A hardware-oriented approach and a data-oriented approach. The hardware-oriented approach involves installing sensors on the distribution network to detect illegal electricity consumers, while the data-based approach involves training a model to identify anomalous load patterns. By utilizing both of these approaches, the system can effectively identify and prevent electricity larceny, thereby improving system stability, reliability, and efficiency.

Moreover, we propose the use of the Random Forest algorithm in power factor correction, which is a measure of the efficiency of the electricity distribution system. By improving the power factor, we can reduce energy waste and lower the cost of electricity provision. Figure 3.1 depicts the overall design of the proposed system.

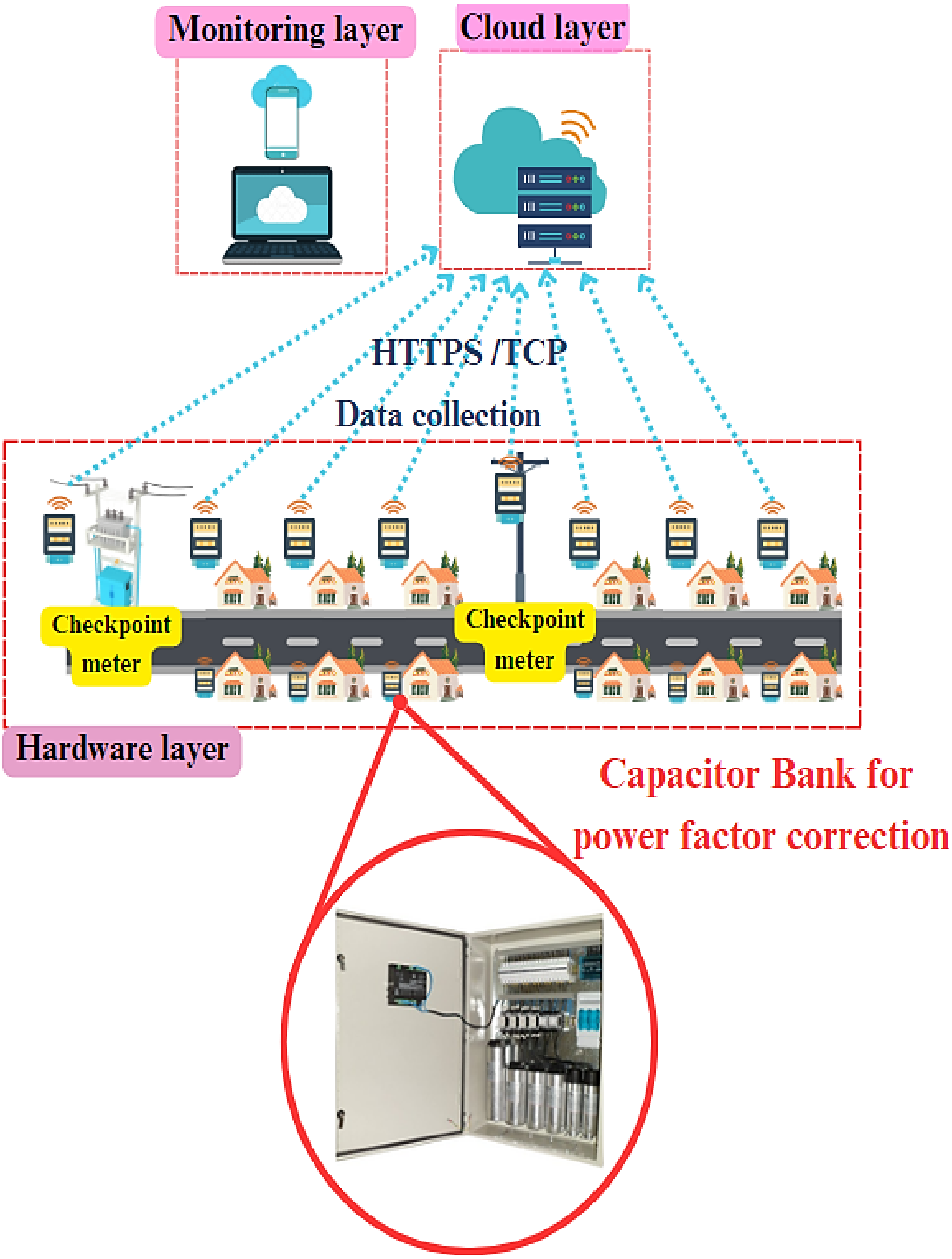


Figure 3.1: Overall Proposed System

3.2 Homes Electric Larceny Detection System

The main objective of this system is to utilize intelligent cloud computing to detect electricity larceny. Implementing this system requires the deployment of smart meters to collect data, which forms the device layer. The collected data is then transmitted to the cloud layer, where it undergoes storage and analysis. Within the cloud layer, a hybrid-oriented approach is employed to identify instances of larceny, which combine a hardware-oriented approach and a data-oriented approach.

The hardware-oriented approach involves the installation of smart meters at specific locations along the distribution lines to detect any difference in power drawn and power consumption. On the other hand, the data-oriented approach relies on deep learning algorithms, particularly convolutional neural networks, to learn from historical data and recognize abnormal load patterns that may indicate larceny. The results obtained from the analysis are then presented through the monitoring layer, which is represented by a web-based application.

By employing this system, it becomes possible to detect theft more efficiently compared to traditional methods. Also, this system places a high priority on loss reduction through the utilization of cloud computing to multi-processing of home data into a single system.

The utilization of a hybrid-oriented approach is essential due to the limitations of each approach. Hardware-oriented approaches are effective in detecting most types of electricity theft. However, when consumers tamper with meters or commits fraud, it becomes challenging for authorities to identify the responsible parties, given the large number of consumers. This is where the data-oriented approach becomes crucial. By analyzing the load patterns of individual consumers, abnormal load patterns can be identified, helping in the identification of fraud consumers. However, the data-oriented approach alone is insufficient in detecting unauthorized connections to the electrical network. Therefore, a combination of both approaches is necessary to create an effective theft detection system. Figure. 3.2 show the proposed system for the electric larceny detection block diagram.

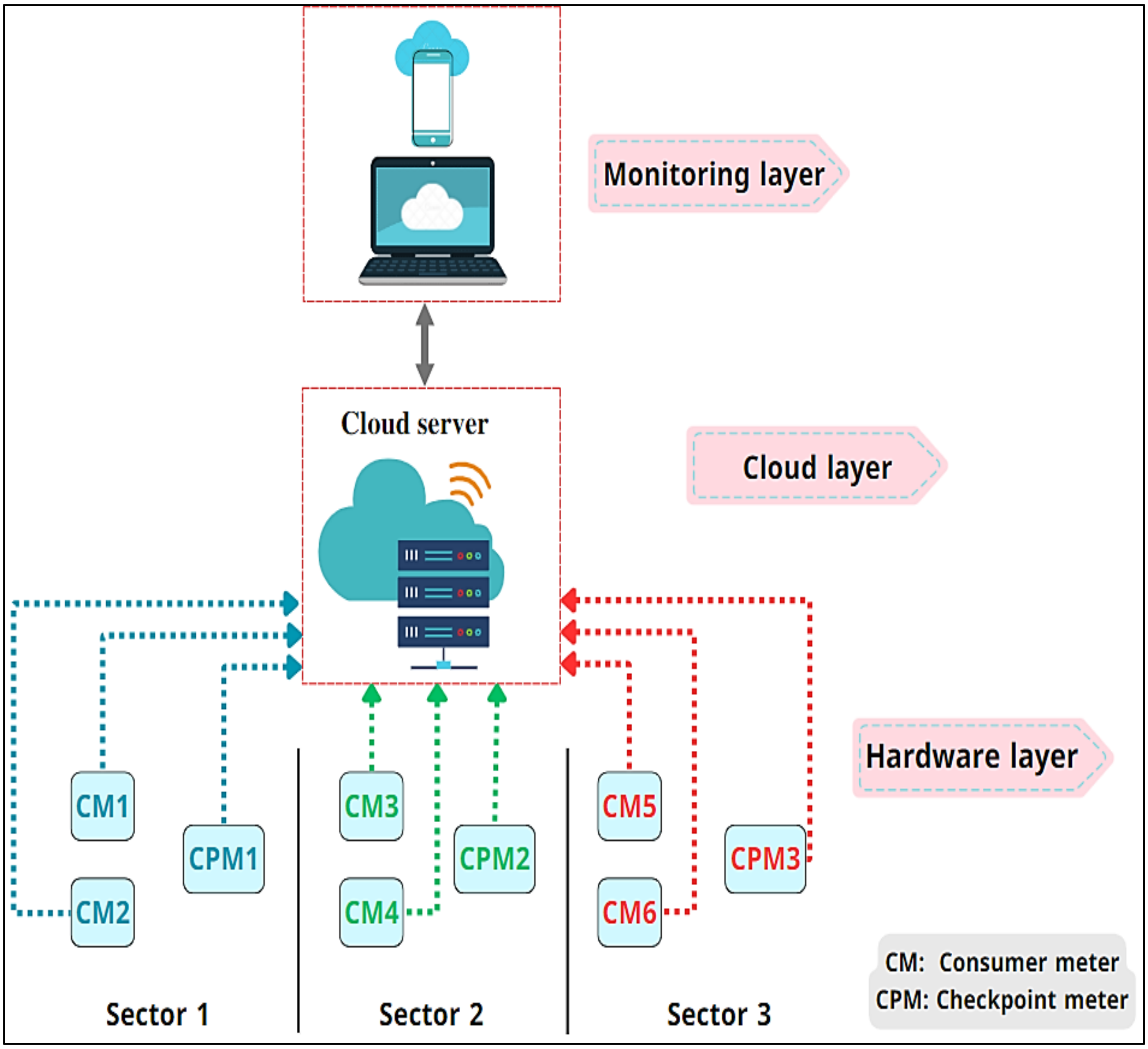


Figure 3.2: proposed electric larceny detection system block diagram.

3.2.1 Hardware layer

The hardware layer of the system is responsible for collecting data related to electrical parameters. This layer involves various components including the Arduino Mega Wi-Fi R3 module, PZEM-004T energy meter, and 20x4 LCD display.

The PZEM-004T energy meter is responsible for measuring various electrical parameters of the connected home, such as voltage, current, and power. Meanwhile, the 20x4 LCD display provides an interface for users to read and comprehend the measured data.

The Arduino Mega Wi-Fi plays a vital role in processing the data obtained by the PZEM-004T sensor. The data is sent to the ESP8266 module, which is an integrated chip within the Arduino Mega Wi-Fi R3 board. This communication occurs through a serial channel with a baud rate of 9600. Once connected to a network, the ESP8266 module obtains an IP address that allows it to transmit data to the cloud server using the HTTPS/TCP request protocol. Figure 3.3 illustrates the connection of the smart meter components.

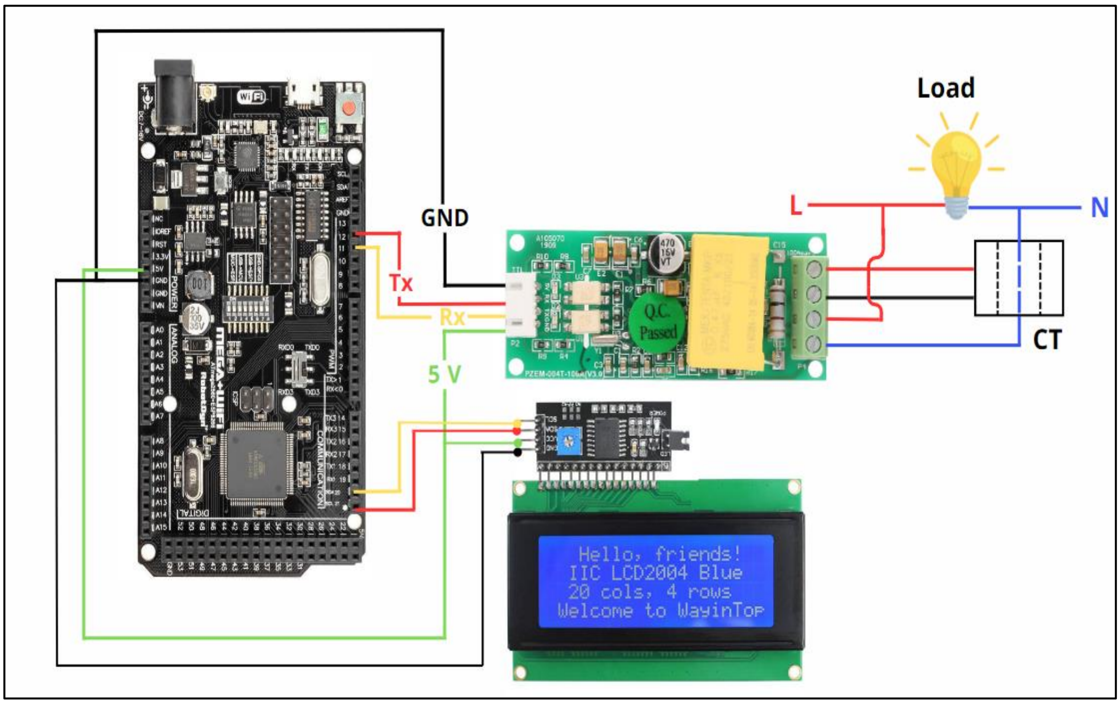


Figure 3.3: The connection of the smart meter components

Table 3.1 provide the pin wiring connections between the PZEM-004T sensor and the Arduino Mega Wi-Fi R3

Table 3.1: PZEM-004T Sensor Pin Wiring

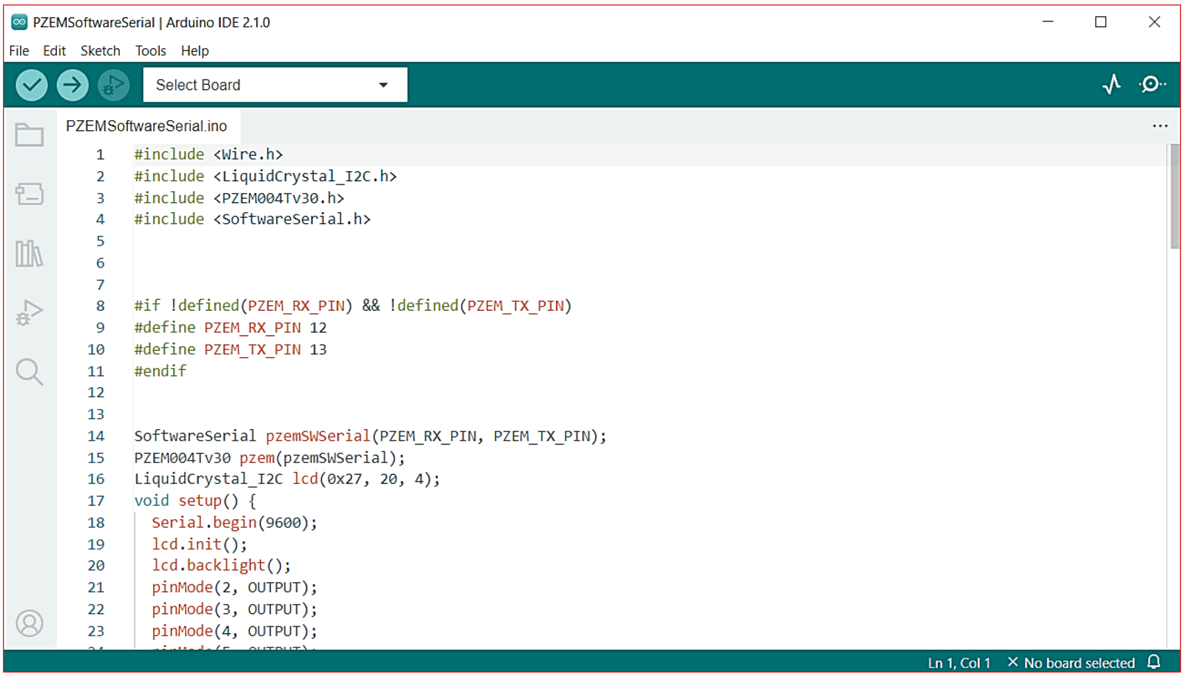
PZEM-004T Pin	Arduino Mega Wi-Fi R3 pin
5V	Connected to 5V pin on Arduino mega Wi-Fi R3
Rx	Connecting to 12 pin on Arduino mega Wi-Fi R3
Tx	Connected to 11 pin on Arduino mega Wi-Fi R3
GND	Connected to GND pin on Arduino mega Wi-Fi R3

Table 3.2 provide the pin wiring connections between the LCD display and the Arduino Mega Wi-Fi R3

Table 3.2: LCD Pinout for Arduino Mega Wi-Fi R3

LCD Pin	Arduino Mega Wi-Fi R3 pin
VCC	Connected to 5V pin on Arduino mega Wi-Fi R3
GND	Connected to GND pin on Arduino mega Wi-Fi R3
SDA	Connected to SDA (20) pin on Arduino mega Wi-Fi R3
SCL	Connected to SCL (21) pin on Arduino mega Wi-Fi R3

The software operation was programmed with Arduino IDE (Arduino Integrated Development Environment) and C++ languages (as shown in Figure 3.4) which are commonly used at the present time.



PZEMSoftwareSerial | Arduino IDE 2.1.0
File Edit Sketch Tools Help
Select Board
PZEMSoftwareSerial.ino
Ln 1, Col 1
No board selected

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <PZEM004Tv30.h>
#include <SoftwareSerial.h>



#if !defined(PZEM_RX_PIN) && !defined(PZEM_TX_PIN)
#define PZEM_RX_PIN 12
#define PZEM_TX_PIN 13
#endif


SoftwareSerial pzemSWSerial(PZEM_RX_PIN, PZEM_TX_PIN);
PZEM004Tv30 pzem(pzemSWSerial);
LiquidCrystal_I2C lcd(0x27, 20, 4);
void setup() {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
```

Figure. 3.4: Arduino IDE

3.2.2 Cloud layer

The cloud layer act as the brain of the proposed system, it is represented by a cloud server. This layer provides the necessary infrastructure for storing and analyzing the data collected from the device layer. The cloud server has a public IP address to enable data transmission between the Arduino and the server. The system allocates space in the cloud server for data storage in a centralized database. This database is designed to address large volumes of data. In this layer, the data analysis algorithms are implemented to detect instances of electricity. The outcomes of the data analysis are then presented on the monitoring layer.

3.2.3 Monitoring layer

The monitoring layer is the user-facing component of the system, represented by a web application for authorities to monitor the electricity distribution network in real time and to display the results of the analysis. This web app has been developed using ASP.NET Core MVC, which is a modern open-source framework developed by Microsoft.

3.3 Electric Larceny Detection Approaches

As mentioned earlier, the system utilizes a hybrid-oriented approach, which combines both hardware-oriented and data-oriented methods for theft detection. In this section, we will provide detailed information about each approach.

3.3.1 Hardware-based approach.

To establish our system, the initial stage involves the installation of smart meters in households (Consumer Meters CM). The distribution line is then divided into multiple zones, and a checkpoint meter (CPM) is installed after each group of houses, as shown in Figure 3.2, to identify the location of electricity theft. Real-time data from both Consumer meters and checkpoint meters is collected and transmitted via HTTPS protocol to a cloud server. Choosing a cloud server provides increased security, as it is a private server that is not shared with other users. This minimizes the possibility of data breaches and illegal access to sensitive data.

A mathematical model is deployed on the cloud server to identify electricity larceny in households. The model is based on the measurements of power, which were recorded by consumer meters and checkpoint meters.

The mathematical model works by comparing the power recorded at the checkpoint meter $(c(k))$ with the sum of power readings from individual consumers$(P_1 + P_2 + P_3 + P_4)$, where k and n are the identification numbers of the checkpoint meter and consumer meter, respectively. Technical losses (TL) and measurement errors (EM) are taken in account. The difference between $c(k)$ and the total power consumption of individual consumers is then calculated to identify the power theft, which is represented asPT.

The equation (3.1) for calculating power theft is given:

$$PT = c(k) - \sum_{i=1}^{n} P_i - (TL + EM) \tag{3.1}$$

If PT is greater than$TL + EM$, it is assumed that electric larceny has occurred.

Consider Figure (3.2), the inspection process begins with the first group, where the power readings of the consumers' meters in the group are compared to the reading of the first checkpoint meter. This is done to identify any instances of theft or discrepancies. Power theft in 1st sector PT_1 determined using equation (3.2):

$$PT_1 = c(1) - [(p_1 + p_2) - (TL + EM)_1] \tag{3.2}$$

Moving on to the second group, the total power readings of the consumers' meters in the second group, in addition to the readings of the first checkpoint meter, are calculated and compared to the designated checkpoint meter for the second group. The power theft in the second sector PT_2 determined using equation (3.3):

$$PT_2 = c(2) - [(p_3 + p_4) + c(1) - (TL + EM)_2] \tag{3.3}$$

This process is repeated for each subsequent group of houses. In general the power theft in k sector PT_k will be:

$$PT_k = c(k) - \left[\sum_{i=1}^{n} P_i + \sum_{j=1}^{k} c(j-1) - (TL + EM)_k\right] \tag{3.4}$$

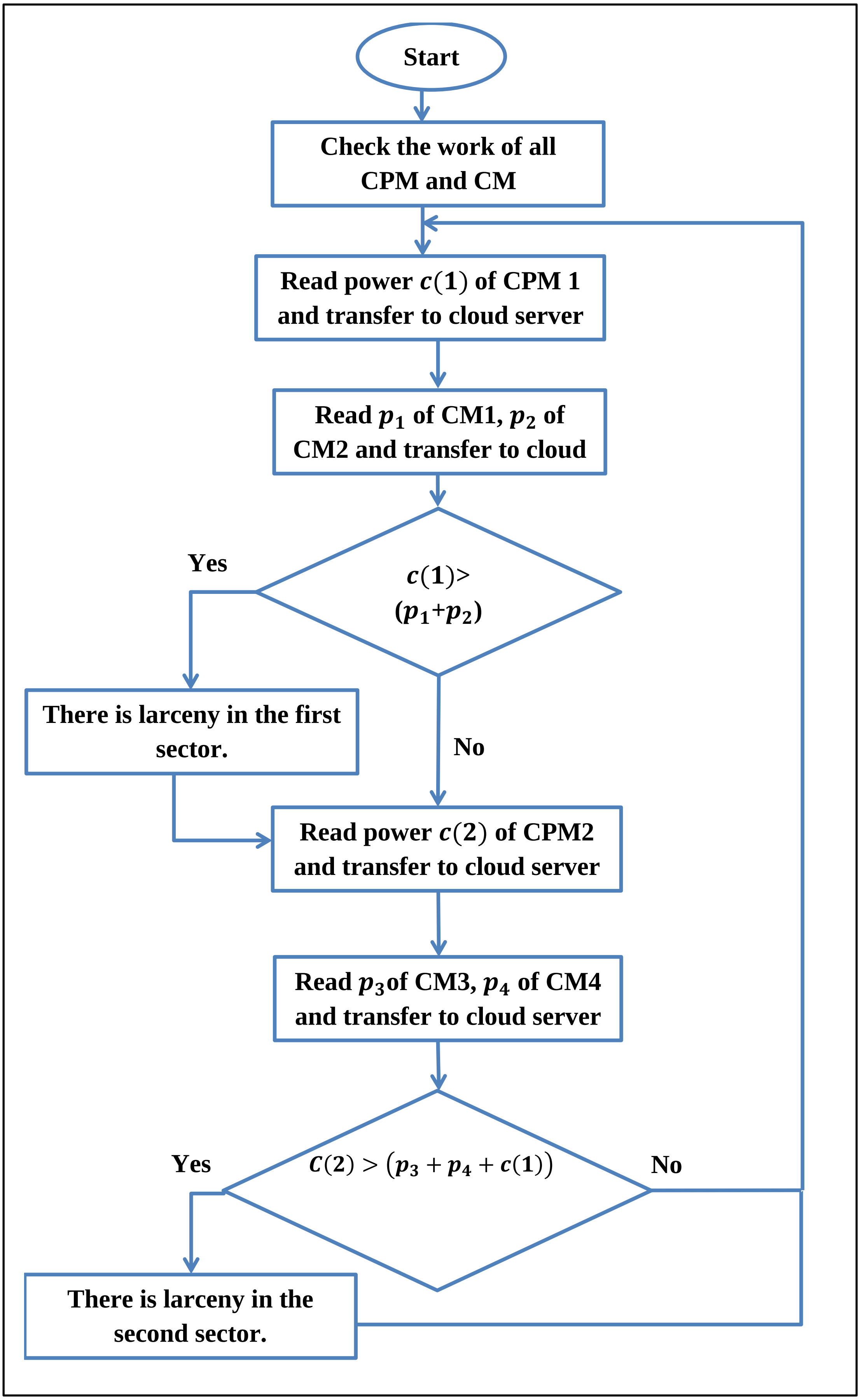
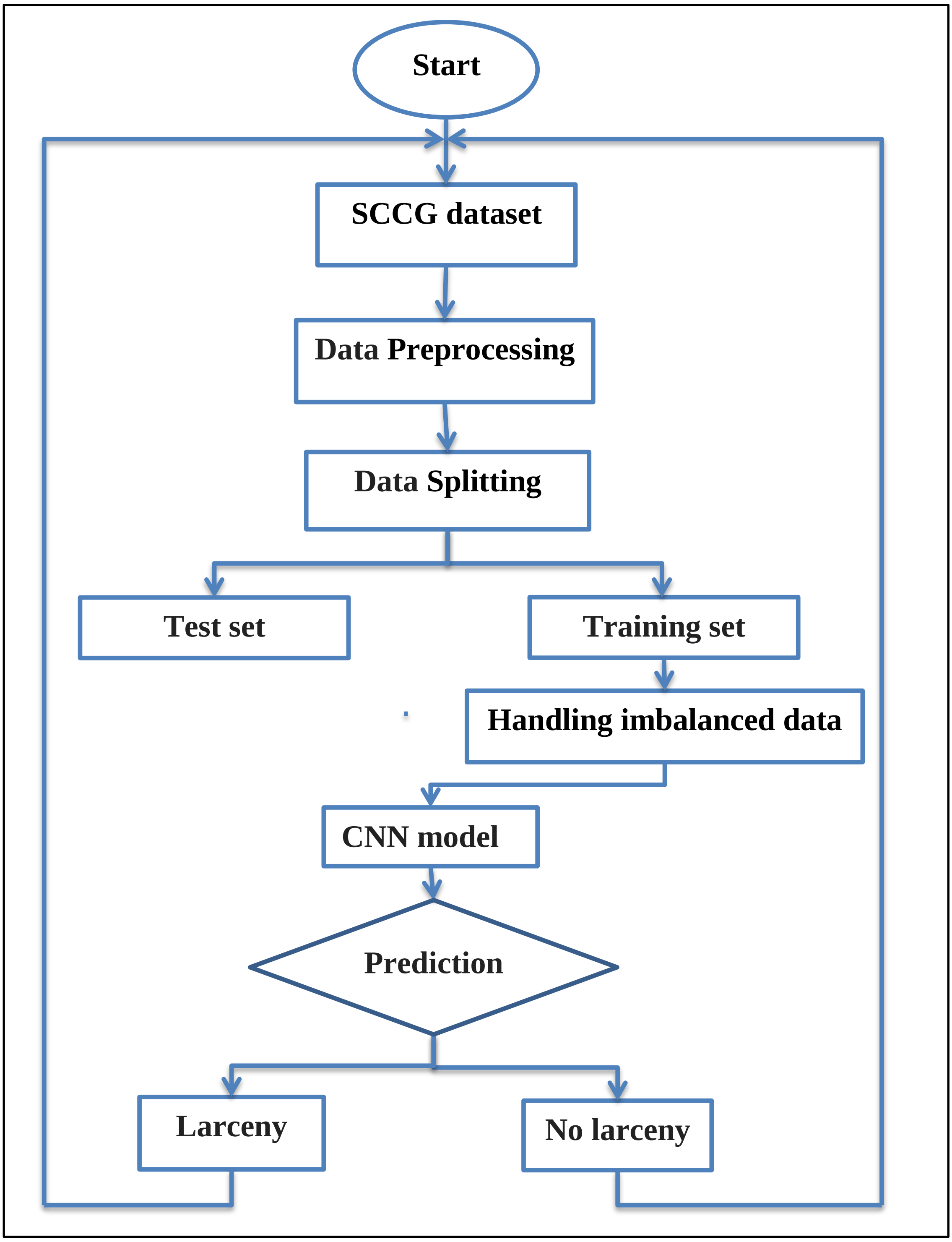

Start
Check the work of all CPM and CM
Read power $c(1)$ of CPM 1 and transfer to cloud server
Read p_1 of CM1, p_2 of CM2 and transfer to cloud
$c(1)> (p_1+p_2)$
Yes
There is larceny in the first sector.
No
Read power $c(2)$ of CPM2 and transfer to cloud server
Read $p_3$of CM3, p_4 of CM4 and transfer to cloud server
$C(2) > (p_3 + p_4 + c(1))$
Yes
No
There is larceny in the second sector.

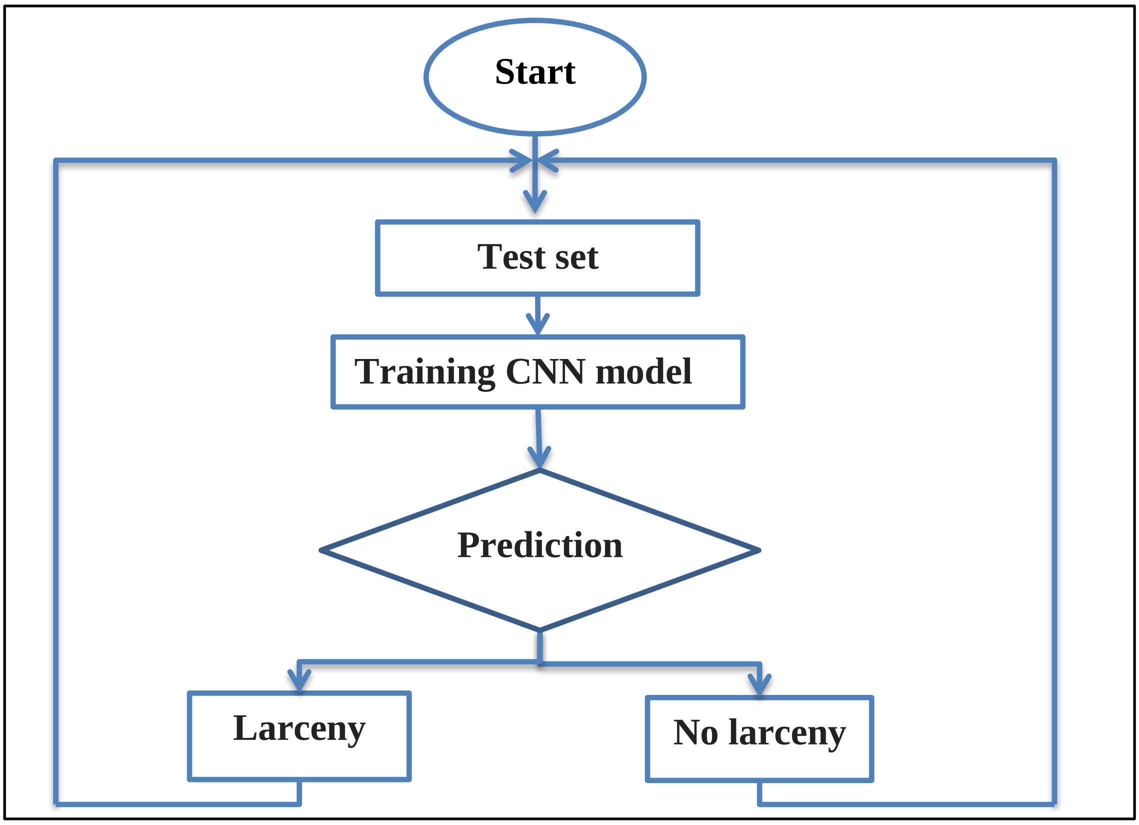
Figure 3.5: The flowchart of the Hardware-based approach.

3.3.2 Data-based approach.

In this approach, DL algorithms, specifically Convolutional Neural Networks (CNN), were employed to electricity larceny detection. The flowchart for this approach is shown in Figure 3.6.



(a)



(b)

Figure 3.6: The flowchart of the Data-based approach. **(a)** Training phase **(b)** Test phase.

The detail of this approach explains below.

3.3.2.1 Data Overview

To train the CNN model, we utilized a dataset called the "Smart Grid Dataset" obtained from Kaggle [83]. This dataset was provided by the State Grid Corporation of China (SGCC) and includes information about consumers' power usage during the period from January 1, 2014, to October 31, 2016.

The dataset consists of rows representing individual consumers and columns representing the amount of electricity consumed on specific dates. In total, there are records for 42,372 electricity customers. Among these customers, the first 3,615 have been identified as having committed electricity theft, while the remaining customers have been classified as honest consumers.

3.3.2.2 Data Preprocessing

Smart grid data requires preprocessing due to potential incompleteness or errors, which can significantly impact model accuracy. Applying preprocessing techniques on smart grid data leads to more accurate results. There are several steps involved in data preprocessing, including:

1. Removing duplicate rows, this can occur due to errors in data collection or storage. Duplicate rows can skew the analysis results, leading to inaccurate predictions.
2. Missing values (NaN) are filled using the 'linear interpolate' method, which estimates values based on surrounding data points. A 'limit' parameter is set to control the number of sequential missing values that can be filled, with a limit of '2' in our case. This prevents inaccurate predictions when there are too many consecutive missing values. After linear interpolation, any remaining missing values are filled using the 'fillna' function, replacing NaN values with '0'. This ensures the dataset is complete. The linear interpolation can be represented by equation (3.8): Consider a set of data $(m_1, m_2, \ldots, m_n)$ and we want to estimate m_i between m_{i-1} and m_{i+1}. We can approximate the value of $f(m_i)$ as follows:

$$f(m_i) = \begin{cases} \frac{m_{i-1} + m_{i+1}}{2}, & m_i \in NaN, \ (m_{i-1}) and (m_{i+1}) \notin NaN \\ m_{i-1,} & (m_i) and (m_{i+1}) \in NaN, \quad m_{i-1} \notin NaN \\ m_{i+1}, & (m_i) and (m_{i-1}) \in NaN, \quad m_{i+1} \notin NaN \\ m_i, & m_i \notin NaN \end{cases} \tag{3.8}$$

3. Outlier treatment is a crucial step in data preprocessing. Outliers refer to data points that deviate significantly from other points in a dataset and can affect the accuracy of models. There are different methods for handling

outliers, such as removing them or replacing them with more appropriate values.

One approach to treat outliers involves calculating the mean (m) and standard deviation (sd) for each row in the dataset. Then, any value in the row that exceeds the threshold of $(m + 3sd)$ is substituted with this threshold value. The mean (m), and standard deviation (sd) of a row can be calculated from equations (3.9), and (3.10) respectively.

$$m = \frac{(x_1 + x_2 + \ldots + x_n)}{n} \tag{3.9}$$

$$sd = \sqrt{\left(\frac{1}{n-1}\right)\left(\left((x_1 - m)^2 + (x_2 - m)^2 + \cdots + (x_n - m)^2\right)\right)} \tag{3.10}$$

Where the row consists of values $x_1, x_2, \ldots, x_n$ and the total number of values in the row is n.

4. Normalization is a data preprocessing technique used to transform data values into a similar range. It is particularly useful when features have widely varying ranges, as this can cause issues in some machine learning algorithms. The most commonly used normalization method is Min-Max scaling, which scales values to a range of 0 to 1.

   In the case of a smart grid dataset, Min-Max scaling is applied to ensure the equal contribution of all features in the analysis. This is achieved using the MinMaxScaler() function.

   Equation (3.11) for Min-Max scaling is:

$$f(x) = \frac{(x - x_{min})}{(x_{max} - x_{min})} \tag{3.11}$$

Where x : is a data point in a column, x_{min} : is the minimum value in that column, x_{max} : is the maximum value in that column, and $f(x)$: is the normalized value between 0 and 1.

3.3.2.3 Handling Imbalance Data

Our smart grid dataset shows class imbalance, with the majority class accounting for 92.14% of the dataset, while the minority class represents only 7.86%.

Class imbalance can introduce bias in machine learning models, making it challenging to accurately detect the minority class. To address this issue, several techniques have been tested, including:

- RUS involves randomly removing instances from the (consumer with no larceny) class to balance the class distribution.
- ROS works by randomly duplicating instances from the (consumer with larceny class) to increase its representation in the dataset.
- SMOTE generates synthetic instances for the (consumer with larceny) class by interpolating between existing instances. This technique creates synthetic data points, effectively increasing the size of the consumer with larceny class and addressing the imbalance.
- SMOTETomek is a hybrid technique that first applies SMOTE to oversample the (consumer with larceny class) and then uses the Edited Nearest Neighbors algorithm to remove noisy or misclassified instances from both the majority and minority classes. This helps to improve the overall quality of the dataset.
- ADASYN is an extension of SMOTE that introduces a higher density of synthetic instances for those (consumer with larceny) class instances that are more challenging to learn. This adaptive approach focuses on the regions of the feature space where the decision boundary is difficult to learn.

Implementing these techniques helps mitigate the impact of class imbalance and improves the performance of models in detecting electricity theft.

It is important to note that techniques for handling class imbalance are applied only to the training set, while the test set remains unprocessed. This is done to ensure that the imbalance handling techniques do not introduce any bias or affect the evaluation of the model's performance on unseen data.

3.3.2.4 Data preparing

Before training the CNN model, the dataset is partitioned into a training set and a test set. CNN model is trained, utilizing the training set. While the test set evaluates its performance on unseen data. Typically, a test size of 0.2 is chosen, reserving 20% of the data for testing and using the remaining 80% for training. Once the dataset is divided, techniques for handling imbalanced data can be applied specifically to the training set.

To prepare the dataset before being fed into the CNN model, it is converted to tensors without affecting the original data. This involves reshaping the dataset into three dimensions. The first dimension indicates the number of samples, whereas the second dimension indicates the number of input variables or features, and the third dimension is set to 1 for a single channel.

Reshaping the dataset in this way ensures it meets the input requirements of the CNN model, enabling effective training and evaluation of electricity consumption data.

3.3.2.5 Building and Training CNN Model

By implementing this CNN model architecture and training it with the specified parameters, we can effectively learn and classify instances of electricity theft and abnormal load patterns in the dataset. The model's architecture and training process contribute to its ability to extract relevant features and make accurate predictions. CNN model detail is described below:

1. CNN Model Architecture

The CNN model consists of multiple layers designed to process the data:

- The model includes two Conv1D layers with 64 filters each. Both layers use a kernel size of 7 and the activation function is ReLU.
- A MaxPooling1D layer having a pool size of 2, which decreases the dimensionality of the feature maps and captures important information.
- To prevent overfitting during training, a Dropout layer is utilized with a dropout rate of 0.2. This randomly ignores a portion of input units.
- A Flatten layer that converts the 3D output into a 1D input for the next fully connected layer.
- A Dense layer with 32 neurons and ReLU activation for feature extraction.
- An additional Dropout layer has a 0.2 dropout rate.
- The final output layer with a single neuron and sigmoid activation, produces a probability value between 0 and 1 for binary classification.
- The proposed CNN model is illustrated in Figure 3.7.

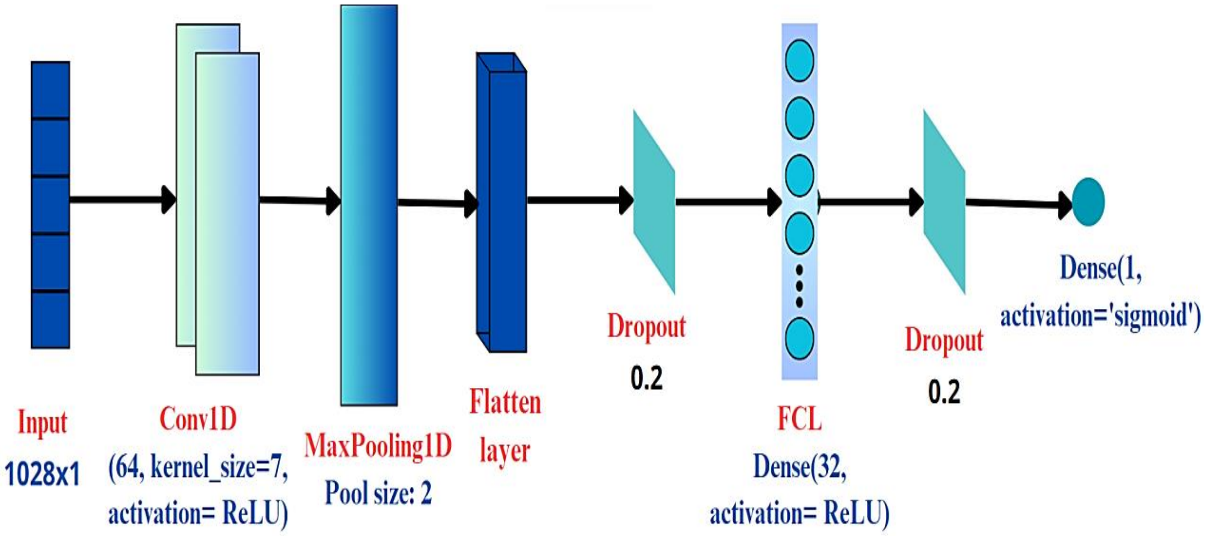


Figure 3.7: The CNN Model Architecture

2. **Model Compilation**

- The loss function utilized to compile the model is binary cross-entropy, which is well-suited for binary classification.
- The most popular optimization algorithm, Adam, is utilized as the optimizer.
- The model's performance during training is evaluated using the accuracy metric.

3. **Model Training:**

- The fit() function is utilized to train the model.
- The model undergoes 10 epochs, iterating over the entire training dataset.
- During training, a validation split of 0.2 is used to allocate 20% of the training set for validation.
- The verbose parameter is set to 1, providing progress and training logs during the training process.

4. **Model Usage**

To use the trained model to detect electricity theft, new consumption data from households can be fed into the model. The model will then analyze the data and predict whether or not electricity theft is occurring. Preprocessing the new data is necessary to ensure that it is in the same format as the training data.

3.4 Power Factor Correction

The system aims to achieve several objectives, including reducing losses and lowering costs. One of the notable features of our system is an automatic power factor correction mechanism that enhances power quality, leading to reduced electricity bills and losses.

To optimize the power factor correction process, a pre-test is conducted to identify common combinations of devices found in residential areas with unequal houses. Selecting appropriate training data plays a crucial role in ML model efficiency, which includes actual power and power factor measurements. This enables each house to achieve the best possible power factor improvement by determining the ideal capacitance value.

The system operates by measuring the power and power factor of home instruments using PZEM-004T. The measured data is then transmitted by Arduino Wi-Fi to the cloud server. The power factor data is displayed through a web app. Additionally, the Random Forest algorithm is utilized to select the optimal capacitor value for Automatic Power Factor Correction (APFC). The capacitor value will be sent to the capacitor selector for precise adjustment of the power factor. Figure 3.9 show the block diagram of two nodes for power factor correction.

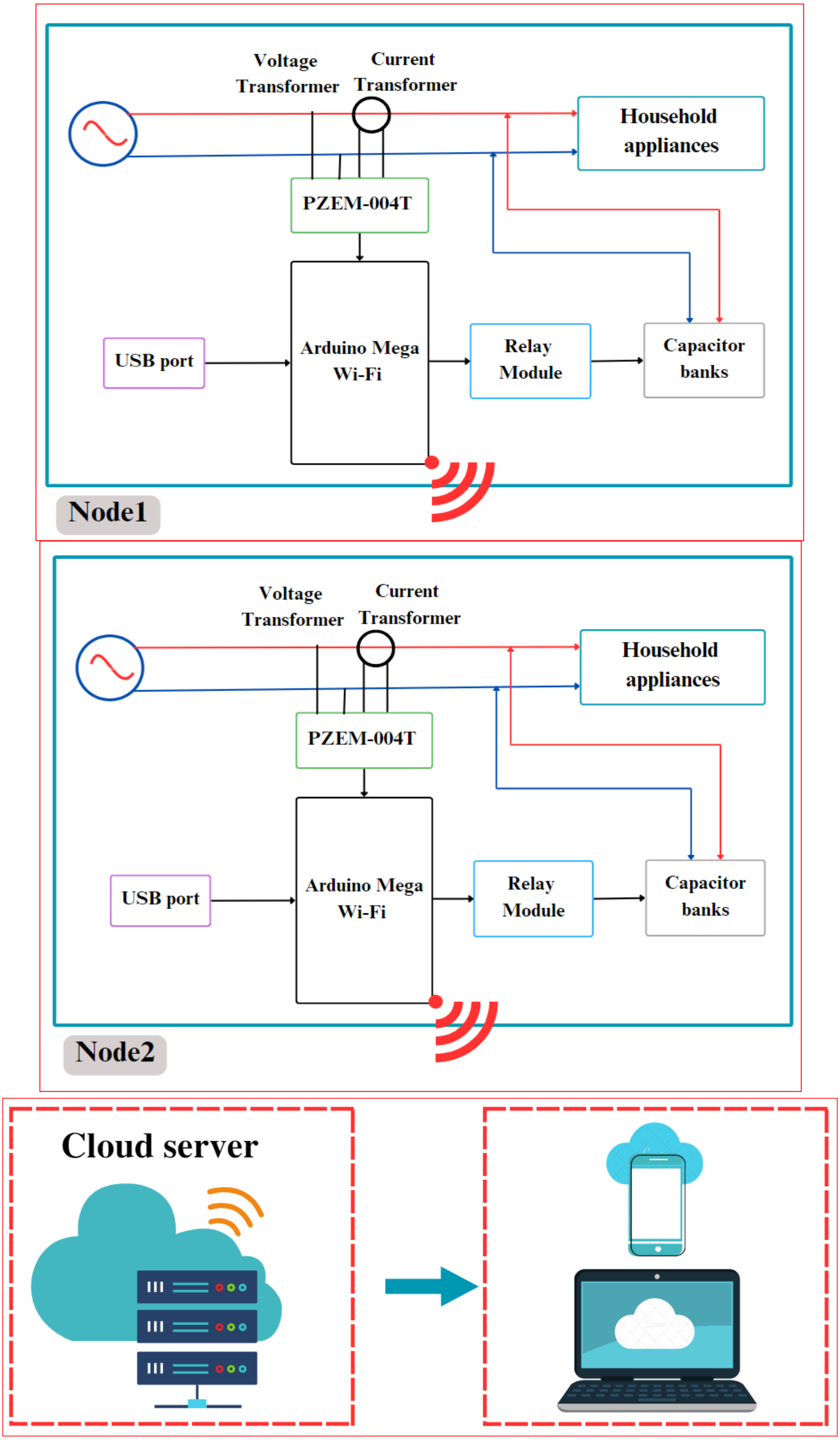


Figure 3.8: The block diagram for two nodes.

3.4.1 Algorithm of Power factor correction

The flowchart for the process algorithm of the power factor correction can be shown in Figure 3.10.

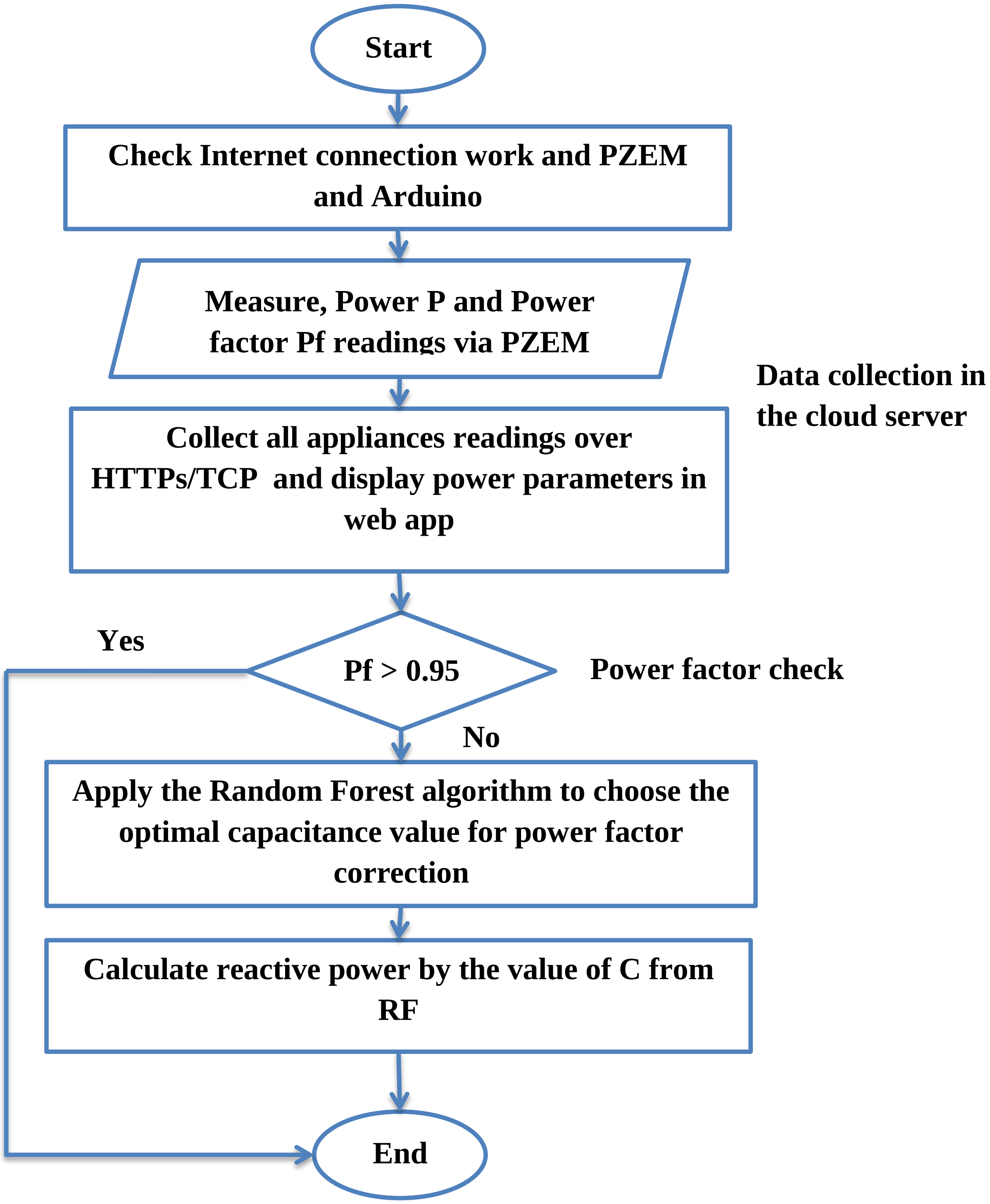

Start
Check Internet connection work and PZEM and Arduino
Measure, Power P and Power factor Pf readings via PZEM
Data collection in the cloud server
Collect all appliances readings over HTTPs/TCP and display power parameters in web app
Yes
Pf > 0.95
Power factor check
No
Apply the Random Forest algorithm to choose the optimal capacitance value for power factor correction
Calculate reactive power by the value of C from RF
End

Figure 3.9: The schematic algorithm of power factor correction.

3.4.2 Random Forest Algorithm for Power Factor Correction

A dataset must be prepared to train and evaluate the Random Forest algorithm. This dataset includes power and power factor measurements obtained from different household appliances. The collected data is used to create an extended dataset that can be utilized for training and testing purposes. Table 3.3 presents a collection of household devices and corresponding readings.

Table 3.3: Different loads with power and power factor

Load device	Power (W)	Pf
LED light lamp Ingco	30	0.55
LED light Ingco	80	0.58
Stand Fan Gosonic GSF-165	55	0.82
Freezer Iceberg 10302	155	0.46
Fridge Concord 540L TE1900-W	237	0.64
LED light Ingco	140	0.55
Gaming console Sony Ps4 pro	88	0.78
LED light Aswar	64	0.55
home random loads 1	266	0.67
home random loads 2	181	0.73
home random loads 3	377	0.61

This table presents a collection of household devices along with their corresponding power and power factor readings. The devices included in the dataset are representative of typical household appliances and cover a range of power consumption levels.

The purpose of this dataset is to provide a diverse set of measurements that capture the characteristics of different appliances. This enables the Random

Forest algorithm to learn patterns and relationships between the power and power factor readings of these appliances.

3.5 Web Application Development

The web application is designed using ASP.NET Core MVC to monitor electrical parameters and display results of analysis of electrical larceny detection models.

ASP.NET Core MVC follows the Model-View-Controller (MVC) architectural pattern, which divides the application into three distinct components: Model, View, and Controller, as shown in Figure 3.10.

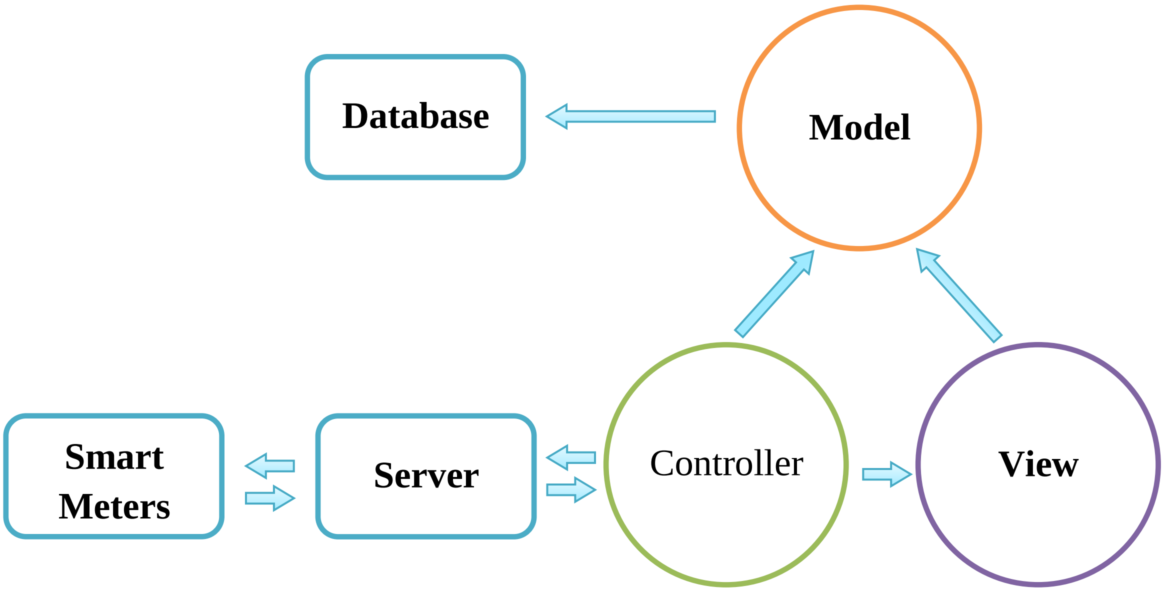
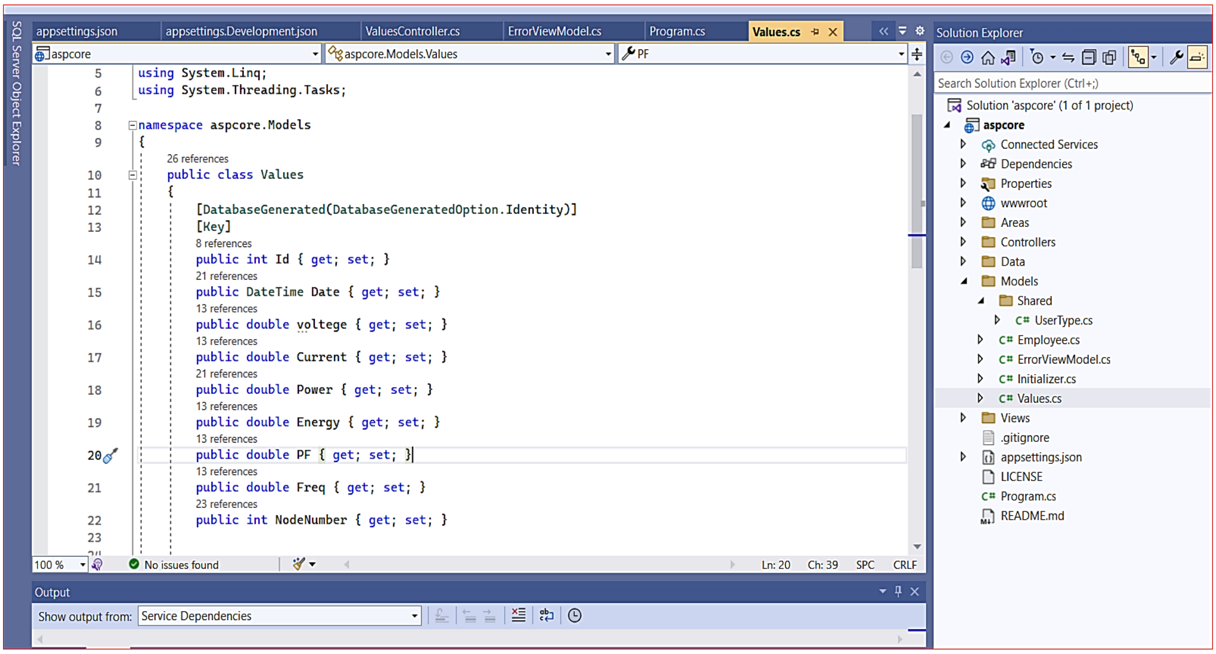


Figure 3.10: Model-View-Controller (MVC) architecture

The Model represents the data and business logic of the application. In this context, it is specifically defined to handle the data received from the Arduino device. The Model class is structured to match the data fields sent by the Arduino device, as illustrated in Figure 3.12.



```
using System.Linq;
using System.Threading.Tasks;

namespace aspcore.Models
{
    26 references
    public class Values
    {
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        [Key]
        8 references
        public int Id { get; set; }
        21 references
        public DateTime Date { get; set; }
        13 references
        public double voltege { get; set; }
        13 references
        public double Current { get; set; }
        21 references
        public double Power { get; set; }
        13 references
        public double Energy { get; set; }
        13 references
        public double PF { get; set; }
        13 references
        public double Freq { get; set; }
        23 references
        public int NodeNumber { get; set; }

```

Figure 3.12: Defined classes of data

The Controller, on the other hand, plays a crucial role in managing electrical parameters and facilitating CRUD operations (Create, Read, Update, and Delete) on these parameters. It serves as the intermediary between the user's actions and the Model, orchestrating data retrieval, manipulation, and updates. Additionally, the Controller interacts with the database to store and retrieve the electrical parameter data.

Regarding data retrieval, the View handles the presentation and user interface aspects. It receives data from the Controller and then populates a template, enabling the user to visualize the information. It is essential to note that the View does not alter the data; its primary responsibility is to display the retrieved data from the Model in a user-friendly manner.

Entity Framework (EF) is using to interact with the database and retrieve information. EF is an ORM (Object-Relational Mapping) tool provided by Microsoft for working with databases in .NET applications. It streamlines the data access process, eliminating the need for writing raw SQL queries and providing a more object-oriented approach to database interaction.

4

CHAPTER FOUR

RESULTS AND DISCUSSIONS

Chapter Four

Results and Discussions

4.1 Introduction

This chapter provides a detailed analysis of the results obtained from the electricity larceny detection system and power factor correction system. It delves into the specific outcomes achieved through the implementation of a hybrid approach that combines both hardware-oriented and data-oriented methods.

Regarding the electricity larceny detection system, the chapter presents the results obtained from data-oriented methods by training a CNN (Convolutional Neural Network) model. Different techniques are applied to address the challenge of data imbalance. The chapter examines the impact of these techniques on the performance of the CNN model and evaluates their effectiveness in improving the detection accuracy of power theft instances.

Furthermore, the chapter explores the device-oriented approach within the power theft detection system. The effectiveness of this approach is evaluated under various conditions.

The chapter also discusses the results obtained from using the Random Forest algorithm to select the optimal capacitance for power factor correction. Additionally, the chapter evaluates the quality of the training data for the Random Forest classifier and examines the performance of the classifier itself.

4.2 Smart Energy Meter

The smart energy meter was built with simple and low-cost components. This meter can measure load voltage, current, frequency, power, power factor, and energy. As mentioned before, the proposed system is consisting of two types of meters, the Consumer meter (CM) and the Checkpoint meter (CPM) as shown in Figure 4.1.

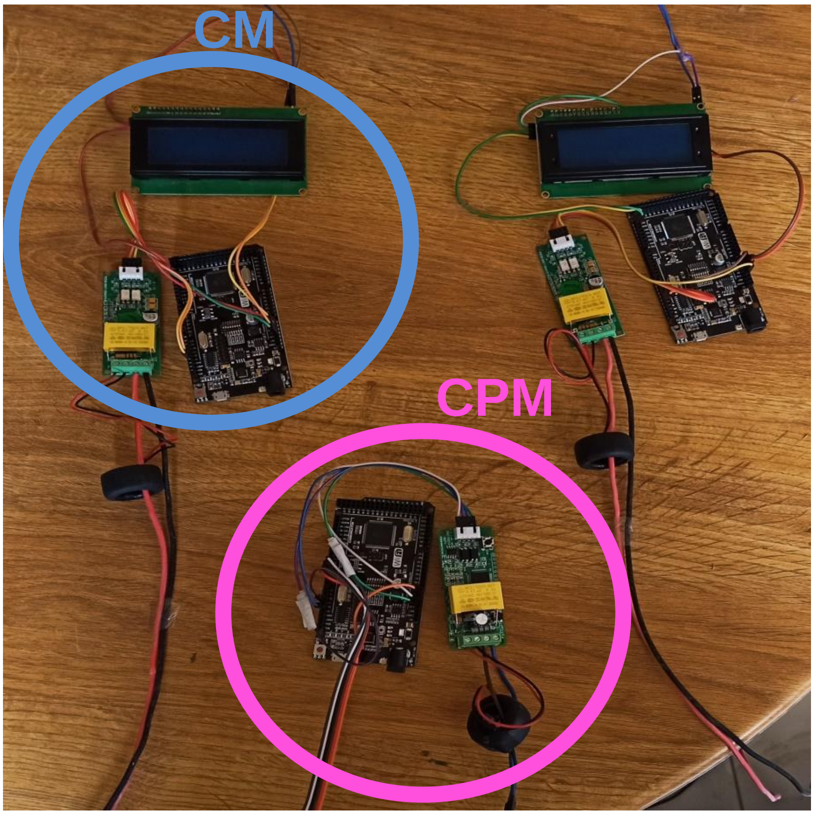
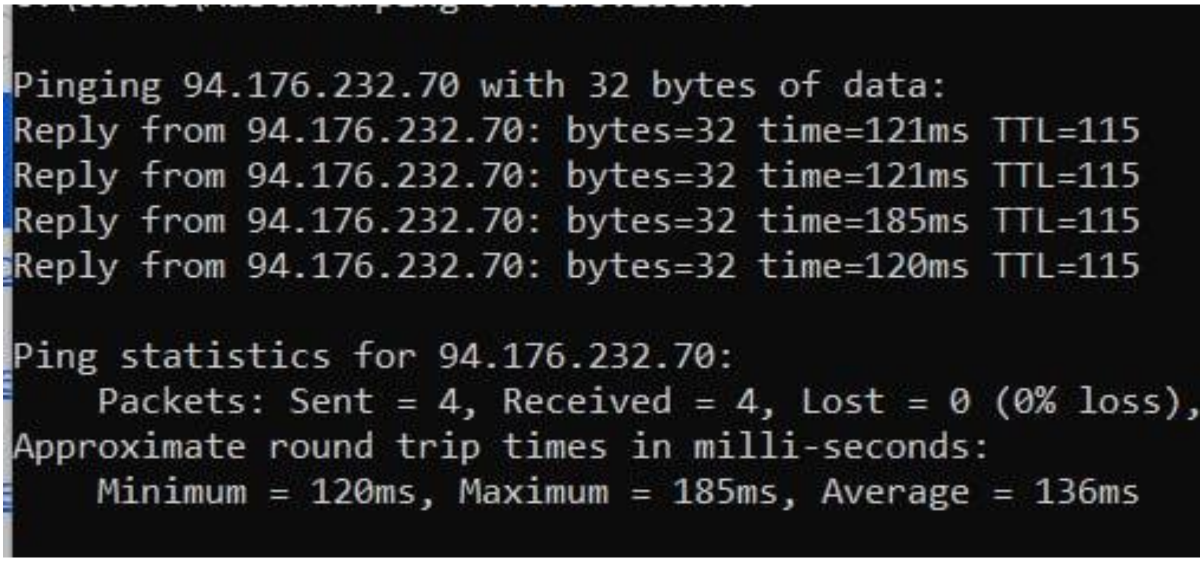


Figure 4.1: Consumer meter (CM) and the Checkpoint meter (CPM).

The Arduino in the smart meter is responsible for sending data to the cloud server for storage and analysis. The ping utility is commonly used to assess the reachability and responsiveness of a network host or IP address, such as a cloud server. In the context of a smart meter, the ping utility is used to test the connectivity and response time of the communication network that connects the smart meter to the cloud server. It does this by sending small packets of data from the Arduino to the cloud server and measuring the round trip time (RTT).



```
Pinging 94.176.232.70 with 32 bytes of data:
Reply from 94.176.232.70: bytes=32 time=121ms TTL=115
Reply from 94.176.232.70: bytes=32 time=121ms TTL=115
Reply from 94.176.232.70: bytes=32 time=185ms TTL=115
Reply from 94.176.232.70: bytes=32 time=120ms TTL=115

Ping statistics for 94.176.232.70:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 120ms, Maximum = 185ms, Average = 136ms
```

Figure 4.2: Ping Results

The ping results in Figure 4.2 indicate the following:

1. **Successful Connection:** All four packets sent from the smart meter to the server were received without any loss, confirming a reliable network connection.

2. **Round Trip Times:** The minimum round trip time recorded was 120ms, representing the shortest time for a packet to complete the round trip. The maximum round trip time was 185ms, indicating the longest time recorded. The average round trip time of 136ms gives an overall measure of the typical time for a packet to complete the round trip.

Based on these results, we can conclude that the smart meter successfully communicated with the cloud server. The consistent round trip times and absence of packet loss indicate a stable network connection, enabling accurate transmission of electrical parameter measurements.

4.3 Evaluation

In this section, we will discuss the impact of data preprocessing techniques and the effects of data imbalance handling techniques, such as RUS, ROS, SMOTE, SMOTETOMK, and ADASYN, on the performance of the CNN model. We will examine how these techniques affect the model's ability to learn from imbalanced datasets and improve its performance in detecting power theft instances.

Furthermore, we will evaluate the performance of the CNN model after applying these data preprocessing and imbalance handling techniques. This evaluation will provide insights into the effectiveness of each technique in enhancing the model's accuracy and its ability to correctly classify power theft cases.

4.3.1 Data preprocessing

The dataset initially contains information on 42,372 electricity consumers, with 3,615 identified as thieves and the remaining customers being honest. After preprocessing, the data includes 37,892 instances of honest customers and 3,602 instances of thieves, totaling 41,494 instances.

Table 4.1: Impact of preprocessing on the Dataset

	Consumers with larceny	Consumers with no larceny	Total Consumers
Data before preprocessing	3,615	38,757	42,372
Data after preprocessing	3,602	37,892	41,494

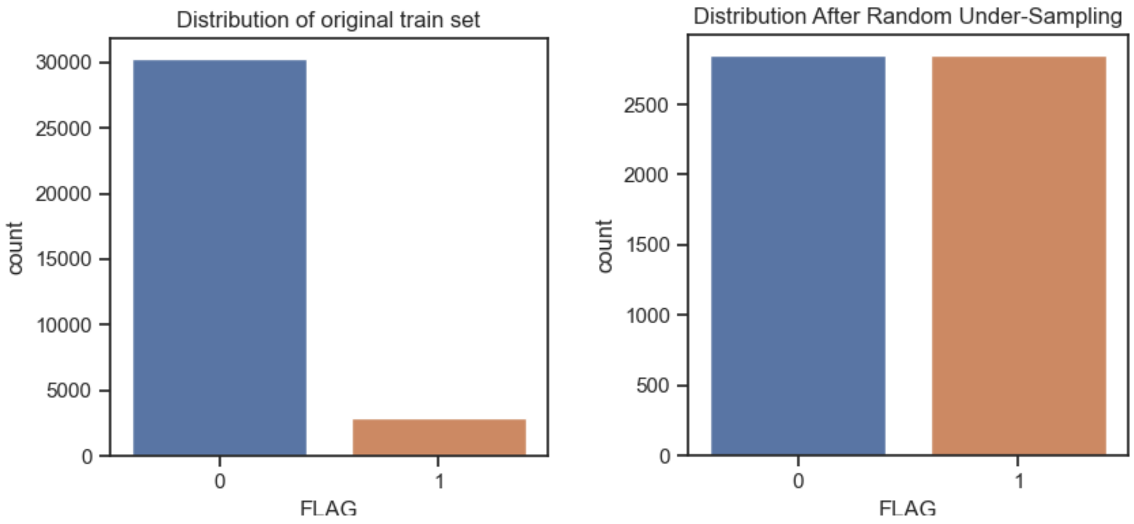
4.3.2 Handling Data Imbalance

Table 4.2 illustrates the effects of using different methods to handle imbalanced data on the train set. The "Original" column represents the train set without balancing. The techniques used are RUS, ROS, SMOTE, SMOTETomek, and ADASYN. Table 4.2 contains the number of consumers with no larceny, the number of consumers with larceny, and the total number of consumers.

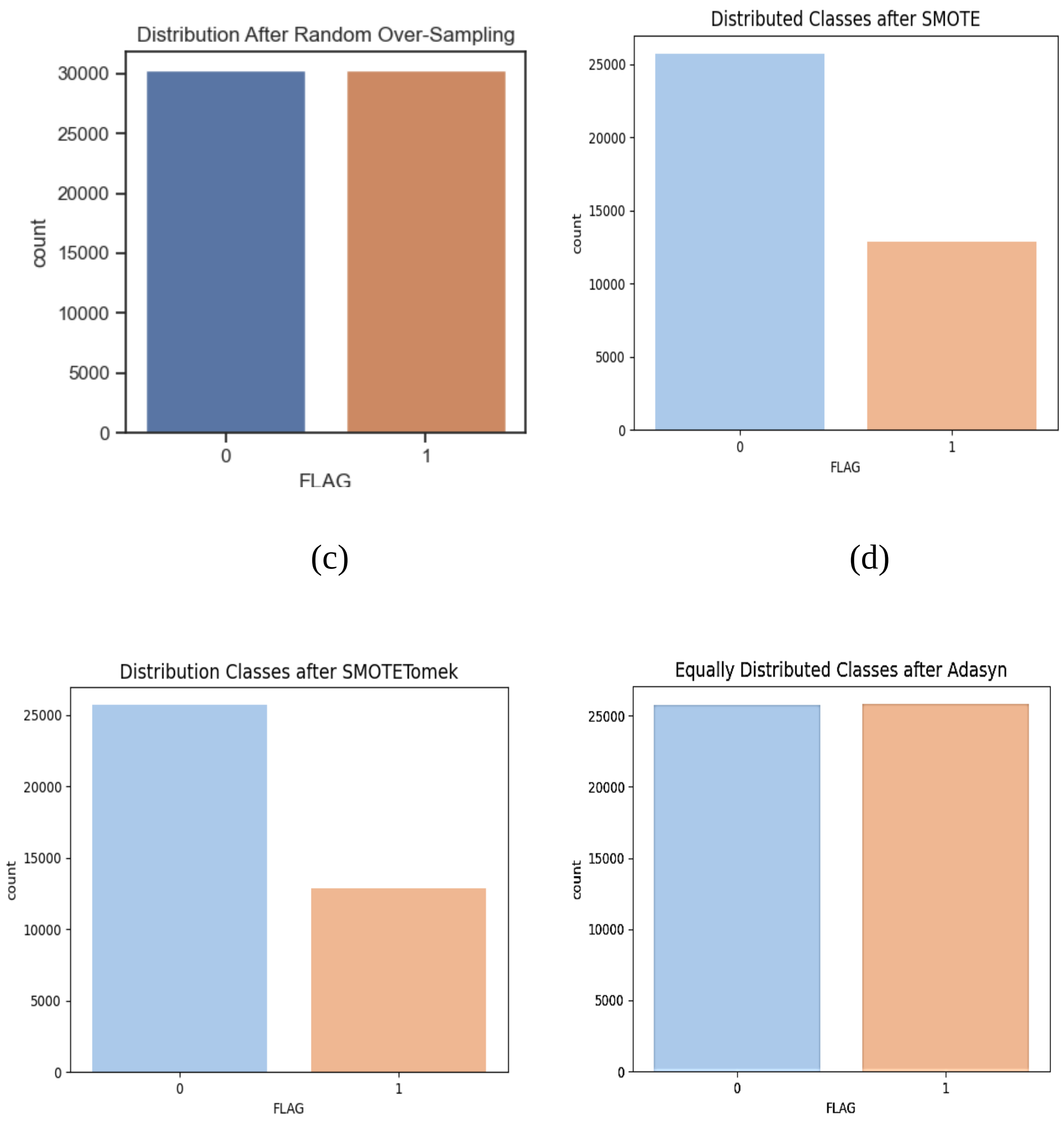
Table 4.2: Distribution of Consumers with and Without Larceny after apply Different Data Balance Techniques

Data Balancing Techniques	Original	RUS	ROS	SMOTE	SMOTETomek	ADASYN
Consumers with no larceny	30341	2854	30341	29370	25705	25705
Consumers with larceny	2854	2854	30341	14685	12852	25795
Total consumers	40256	7158	73354	44055	38557	51500

Figure 4.3 illustrates the distribution of classes (Larceny vs. No larceny classes) after employing various techniques to address the imbalance problem.



(a) (b)



(c) (d)

(e) (f)

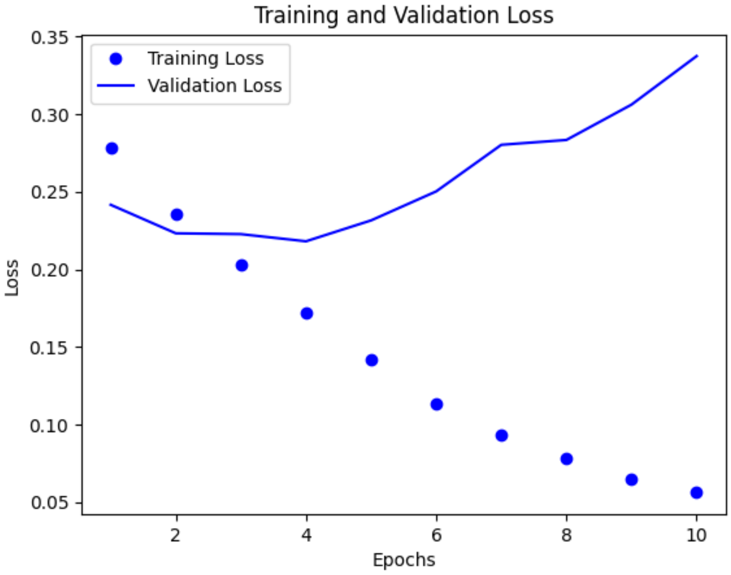
Figure 4.3: Classes distribution a. Original Data. b. Classes distribution after Random Under-Sampling (RUS). c. Classes distribution after Random Over-Sampling (ROS). d. Classes distribution after Synthetic Minority Over-sampling Technique (SMOTE). e. Classes distribution after (SMOTETomek). f. Classes distribution after Adaptive Synthetic Sampling (ADASYN).

4.3.3 CNN model evaluation

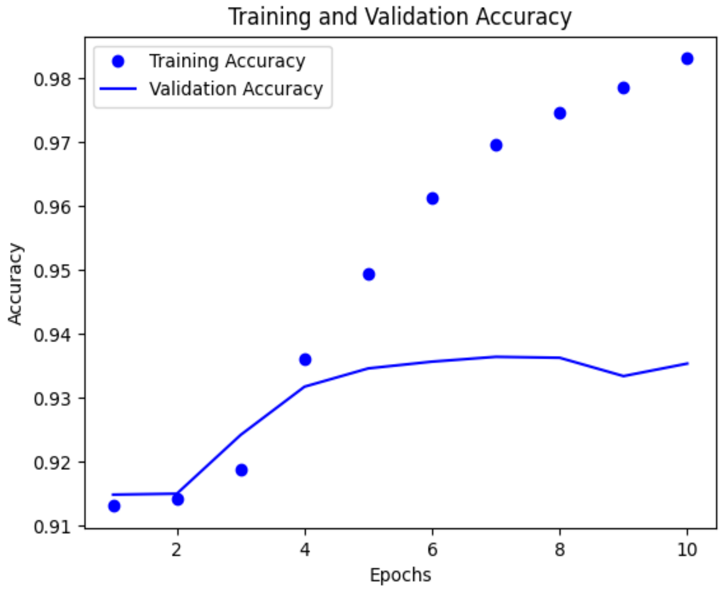
The model was trained for ten epochs, and during training, the training accuracy reaching 99.34% and validation accuracy at 93.63%. The test set evaluation resulted in an accuracy of 93.18%. The F1-Scores for the positive class (larceny) and the negative class (no larceny) were 57.61% and 96.29% respectively. The lower F1-Score for the positive class suggests a struggle in accurately detecting instances of larceny, requiring further investigation and improvement. The AUC value of 73.96% indicates moderate discriminative power, leaving room for enhancement. The precision of 67.2% means that only 67.2% of the instances predicted as larceny are true positives, while the recall of 50.4% indicates the model's ability to correctly identify 50.4% of actual larceny instances. Table 4.3 contains results related to the performance of a Convolutional Neural Network (CNN) model when balance techniques are not applied.

Table 4.3 performance of CNN model without balance techniques.

Training accuracy	Validation accuracy	Test accuracy	F1-Score for (larceny) class	F1-Score for (No larceny) class	AUC	Precision	Recall
99.34%	93.63%	93.18%	57.61%	96.29%	73.96%	67.2%	50.4%



(a)



(b)

Figure 4.4: (a) Training and validation loss curve of CNN without balance technique. **(b)** Training and validation curve accuracy of CNN without balance technique.

The loss curve Figure 4.4 (a) shows decreasing loss for the training data but increasing loss for the validation data. The accuracy curve Figure 4.4 (b) demonstrates improving accuracy for the training data but limited improvement for the validation data.

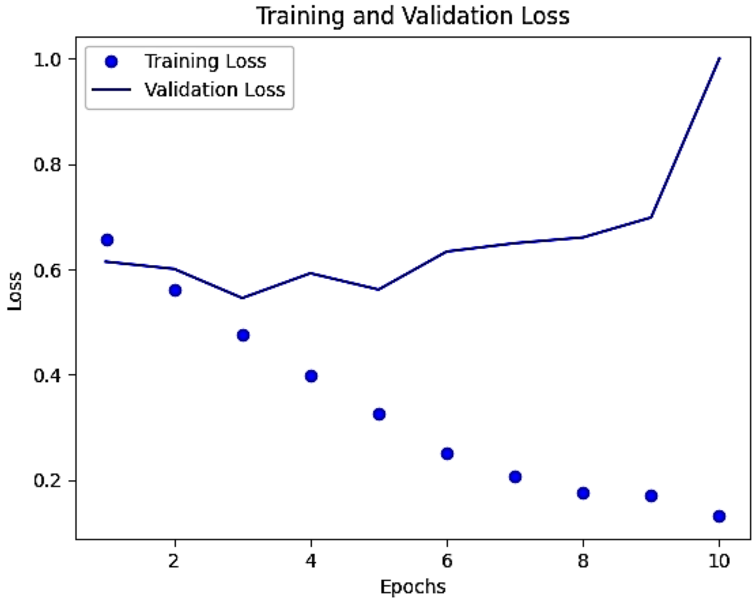
4.3.4 CNN Model performance after RUS

The model was trained for ten epochs, and during training, the accuracy of the training data reached 94.91. However, when evaluating the model on the validation data, the accuracy dropped to 75.65%. This suggests that the model performed well on the data it was trained on but struggled to generalize to new, unseen data. To further evaluate, a test set evaluation was conducted, resulting in an accuracy of 78.21% .

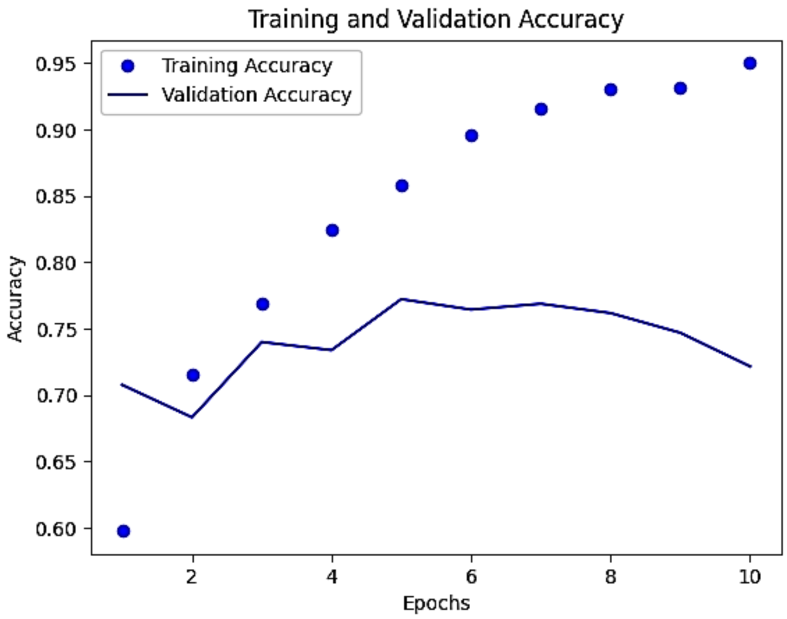
However, the RUS technique improved the AUC and F1 score for (Larceny) class. The original model might have been biased toward the majority class, leading to a lower F1-score and AUC. RUS rebalances the class distribution by removing instances from the majority class, allowing the model to focus more on learning from the positive class. This improves the model's ability to correctly identify positive instances and distinguish them from negative instances. However, it's important to note that the RUS technique comes with a trade-off. While it improves the F1-score and AUC, it results in a decrease in overall accuracy as it involves sacrificing information by removing instances from the majority class. Table Table 4.4 presents various performance metrics for the CNN model after applying the RUS technique

Table 4.4 performance of CNN model after RUS.

Training accuracy	Validation accuracy	Test accuracy	F1-Score for (larceny) class	F1-Score for (No larceny) class	AUC	Precision	Recall
94.91	76.61%	79.61%	78.21%	75.65%	78.06%	79.1%	74.3%



(a)



(b)

Figure 4.5: (a) Training and validation loss curve of CNN after RUS technique. **(b)** Training and validation curve accuracy of CNN after RUS technique.

The loss curve Figure 4.5 (a) shows decreasing loss for the training data but increasing loss for the validation data. The accuracy curve Figure 4.5 (b) demonstrates improving accuracy for the training data but limited improvement for the validation data.

4.3.5 CNN Model performance after ROS

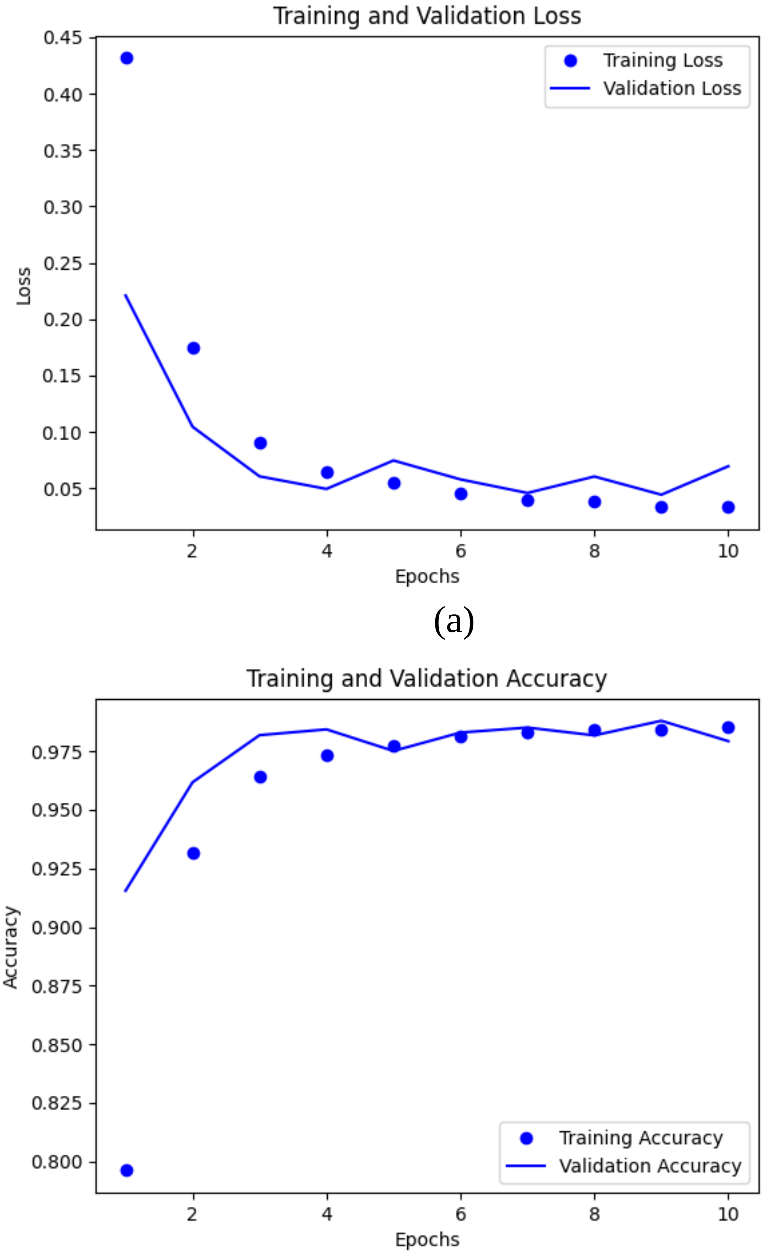
The CNN model achieved high accuracy on both the training and validation sets, with 98.35% and 98.55% respectively. It also performed excellently on the test set, achieving a test accuracy of 98.35%. The F1-Scores for both the positive class (larceny) and the negative class (no larceny) are also high, at 98.33% and 98.37% respectively. The AUC value is 98.35%, indicating excellent discriminative power.

However, these high results may be attributed to overfitting, where the model memorizes the training data without fully understanding the underlying patterns. This means that the model might struggle when applied to new, unseen data.

Table 4.5 provides performance metrics for the CNN model after applying the ROS technique.

Table 4.5 performance of CNN model after ROS.

Training accuracy	Validation accuracy	Test accuracy	F1-Score for (larceny) class	F1-Score for (No larceny) class	AUC	Precision	Recall
98.35%	98.37%	98.33%	98.35%	98.55%	98.35%	96.9%	99.9%



(a)

(b)

Figure 4.6: (a) Training and validation loss curve of CNN after ROS technique. **(b)** Training and validation curve accuracy of CNN after ROS technique.

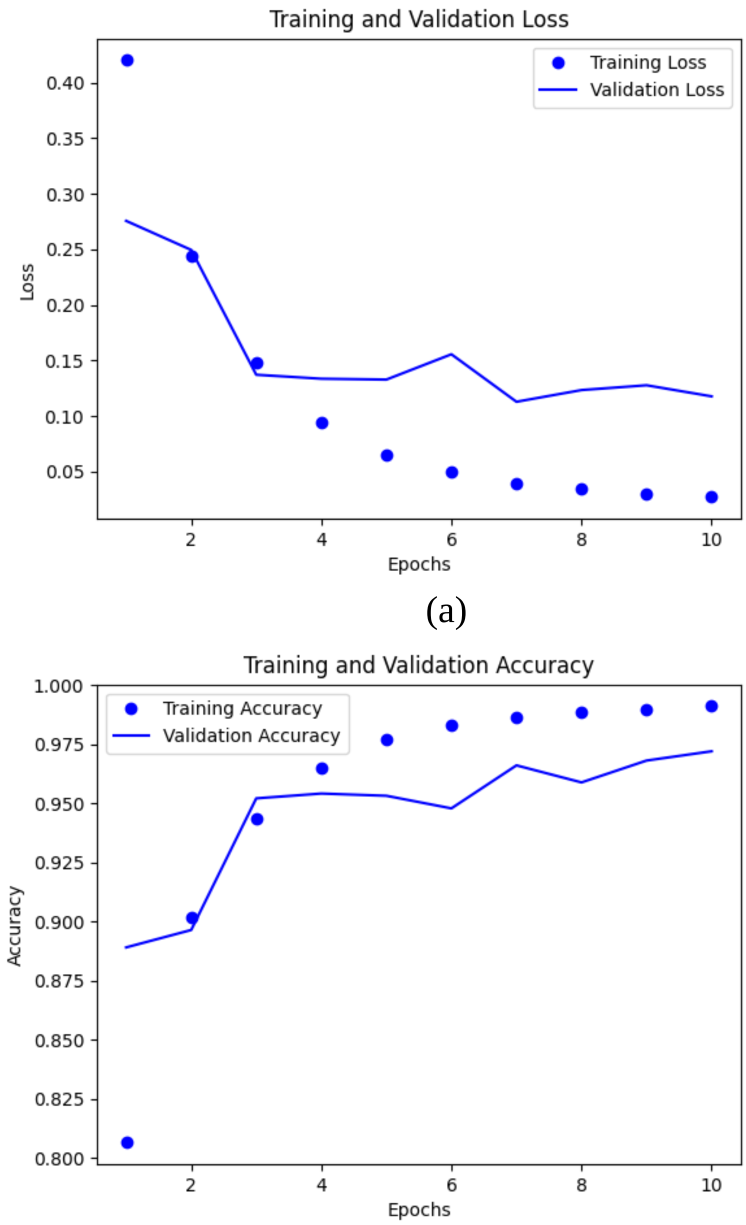
The loss curve Figure 4.6 (a) shows a decreasing trend for both the training and validation data, indicating that the model is learning and improving over the epochs. The accuracy curve Figure 4.6 (b) also demonstrates an increasing trend for both the training and validation data, further indicating the model's learning progress.

4.3.6 CNN Model performance after SMOTE

The model was trained for ten epochs, with the training accuracy reaching 99.49% and the validation accuracy at 94.70%. A test set evaluation was conducted on the trained model, resulting in a test accuracy of 94.70%. This indicates that the model exhibits good generalization to unseen data and is capable of making accurate predictions. The F1-Score for the positive class (larceny) is 95.91% and 92.45% for the negative class (no larceny), indicating that the model is effective at detecting instances of larceny. The AUC value of 95.27% signifies that the model demonstrates strong discriminative power and can effectively differentiate between instances of larceny and no larceny.

Table 4.6 performance of CNN model after SMOTE.

Training accuracy	Validation accuracy	Test accuracy	F1-Score for (larceny) class	F1-Score for (No larceny) class	AUC	Precision	Recall
99.49%	94.70%	94.70%	95.91%	92.45%	95.27%	88.3%	97%



(a)

(b)

Figure 4.7: (a) Training and validation loss curve of CNN after SMOTE technique. **(b)** Training and validation curve accuracy of CNN after SMOTE technique.

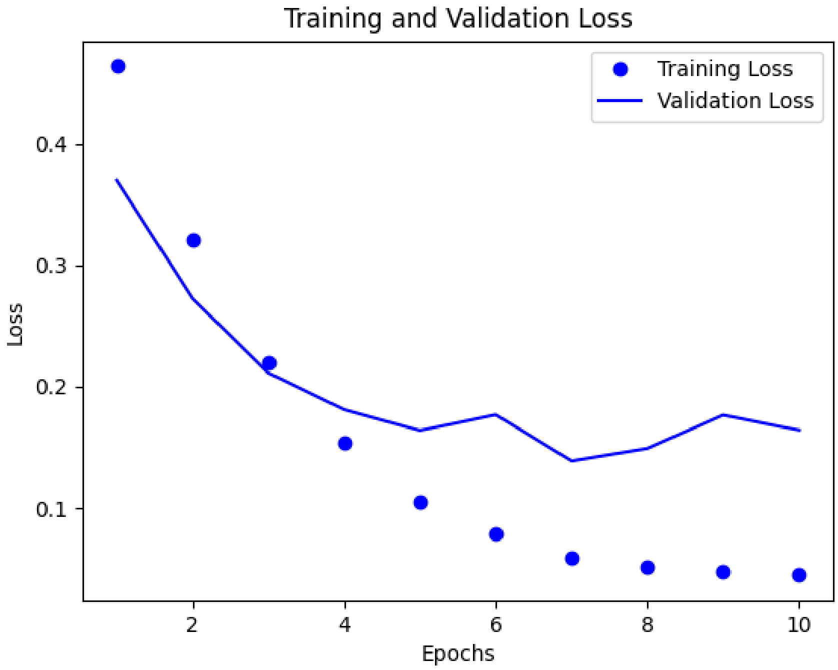
4.3.7 CNN Model performance after SMOTETomek

The model was trained for ten epochs, with the training accuracy reaching 98.90% and the validation accuracy at 96.44% .

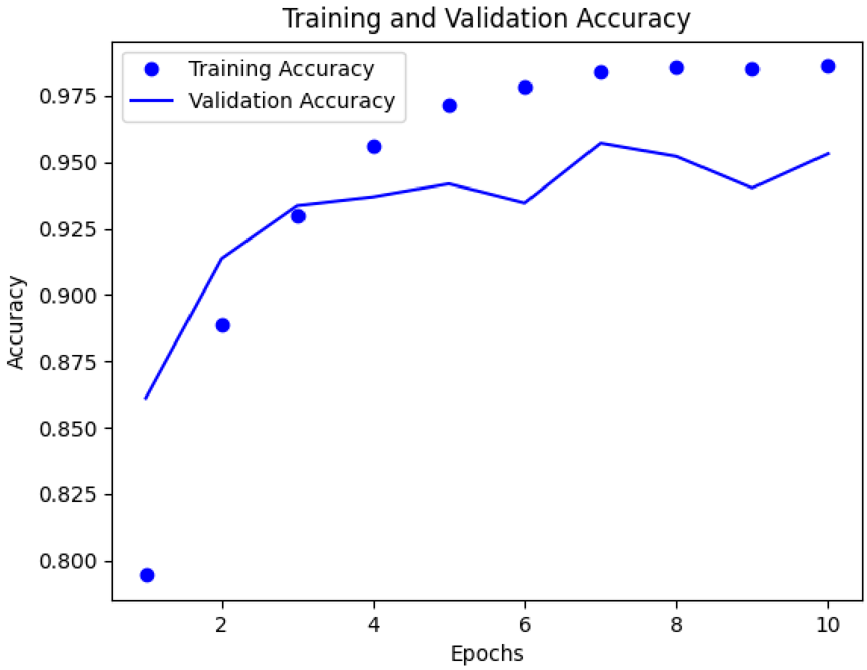
A test set evaluation was conducted on the trained model, resulting in a test accuracy of 95.93%. This indicates that the model exhibits good generalization to unseen data and is capable of making accurate predictions. The F1-Score for the (Larceny) class is 96.88% and The F1-Score for the (No larceny) class is 94.15%, suggesting that the model performs well in identifying larceny cases. These high F1-Score values indicate that the model is capable of achieving a good balance between precision and recall for both classes. Table 4.7 presents performance metrics for the CNN model after applying the SMOTETomek technique.

Table 4.7 performance of CNN model after SMOTETomek.

Training accuracy	Validation accuracy	Test accuracy	F1-Score for (larceny) class	F1-Score for (No larceny) class	AUC	Precision	Recall
98.90%	96.44%	95.93%	96.88%	94.15%	96.30%	91.11%	97.41%



(a)



(b)

Figure 4.8: (a) Training and validation loss curve of CNN after SMOTETomek technique. **(b)** Training and validation curve accuracy of CNN after SMOTETomek technique.

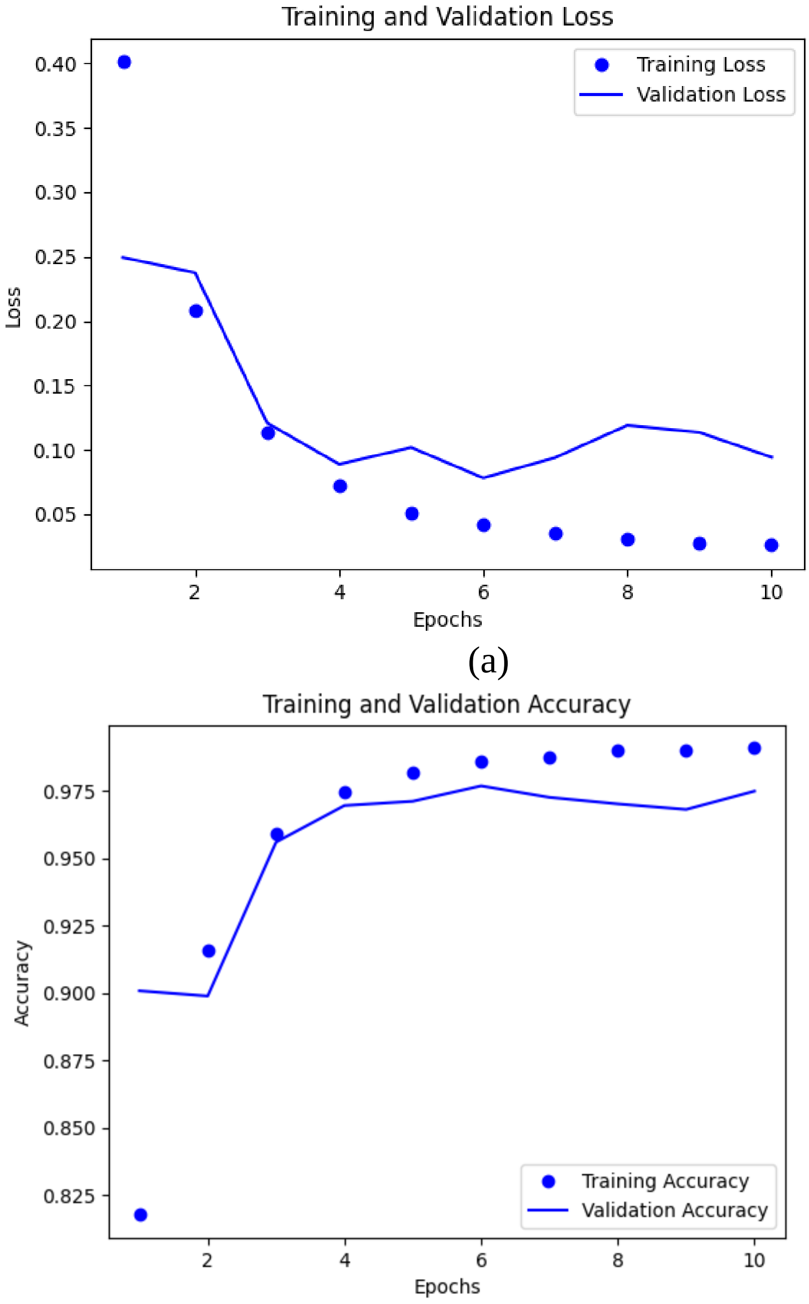
4.3.8 CNN Model performance after ADASYN

The results show that the CNN model trained after applying the ADASYN technique achieved impressive performance. The model was trained for ten epochs, with the training accuracy reaching 99.34% and the validation accuracy at 97.73%. This indicates that the model generalizes well to unseen data and can effectively classify instances outside the training set.

On the test set, the model achieved a high accuracy of 97.22%, indicating its robustness in making accurate predictions on new and unseen data. The AUC value of 97.22% further reinforces the model's discriminative power in distinguishing between positive (larceny) and negative (no larceny) instances. The F1-score for the (Larceny) class and F1-score for the (No larceny) class were 97.14% and 97.30% respectively.

Table 4.8 performance of CNN model after ADASYN.

Training accuracy	Validation accuracy	Test accuracy	F1-Score for (larceny) class	F1-Score for (No larceny) class	AUC	Precision	Recall
99.34%	97.73%	97.22%	97.14%	97.30%	97.22%	96.8%	99.9%



(a)

(b)

Figure 4.9: (a) Training and validation loss curve of CNN after **ADASYN** technique. **(b)** Training and validation curve accuracy of CNN after **ADASYN** technique.

The training and validation loss curve Figure 4.9 (a) indicates a consistent decrease in loss over the epochs, further indicating the effectiveness of the ADASYN technique. Additionally, the training and validation curve accuracy Figure 4.9 (b) shows a steady increase in accuracy.

Considering the provided results, the ADASYN technique demonstrated comparable or better performance than other techniques across multiple evaluation metrics, including accuracy, F1-scores, and AUC. This suggests that ADASYN is a reliable choice for addressing class imbalance and improving the model's performance on minority classes.

Table 4.9: Comparison of the performance of the CNN model with different data balancing techniques.

Metrics	Original dataset	RUS	ROS	SMOTE	SMOTE-Tomek	ADASYN
Training accuracy	99.34%	94.91	98.35%	99.49%	98.90%	99.34%
Validation accuracy	93.63%	76.61%	98.37%	94.70%	96.44%	97.73%
Test accuracy	93.18%	79.61%	98.33%	94.70%	95.93%	97.22%
F1-Score for positive class (larceny)	57.61%	78.21%	98.35%	95.91%	96.88%	97.14%
F1-Score for negative class (No larceny)	96.29%	75.65%	98.55%	92.45%	94.15%	97.30%
AUC	73.96%	78.06%	98.35%	95.27%	96.30%	97.22%
Precision	67.2%	79.1%	96.9%	88.3%	91.11%	96.8%
Recall	50.4%	74.3%	99.9%	97%	97.41%	99.9%

Table 4.9 provides a comparison of the performance of a CNN model with various data balancing techniques, including RUS, ROS, SMOTE, SMOTE-Tomek, and ADASYN.

The performance of the CNN model can be easily understood through confusion matrices which showcase the impact of various techniques on its predictions for minority and majority classes, as depicted in Figure 4.10.

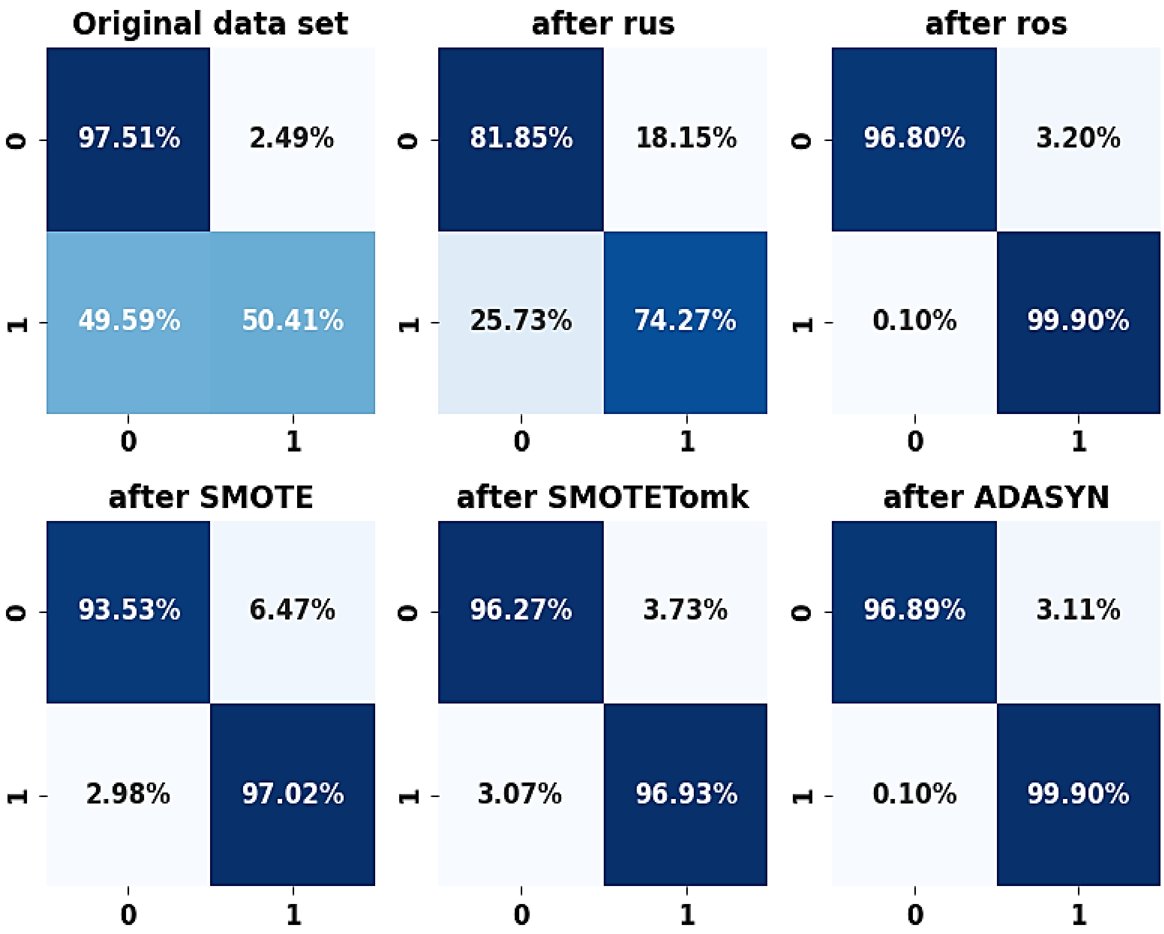


Figure 4.10: The confusion matrices of CNN Model with Various Data Balancing Techniques.

4.4 Home Larceny Detection System

The proposed system, shown in Figure 3.2, was implemented as a prototype environment. The main objective of this system is to identify instances of electricity theft in residential areas. To achieve this, the area is divided into several sectors, and central checkpoint meters (CPMs) are strategically installed at specific locations. These checkpoint meters are used to measure the total power consumption of each sector.

The system was implemented as a prototype in an environment where there are two sectors. Each sector is equipped with two consumer meters (CMs) and one checkpoint meter (CPM), as illustrated in Figure 4.11 of the hardware design.

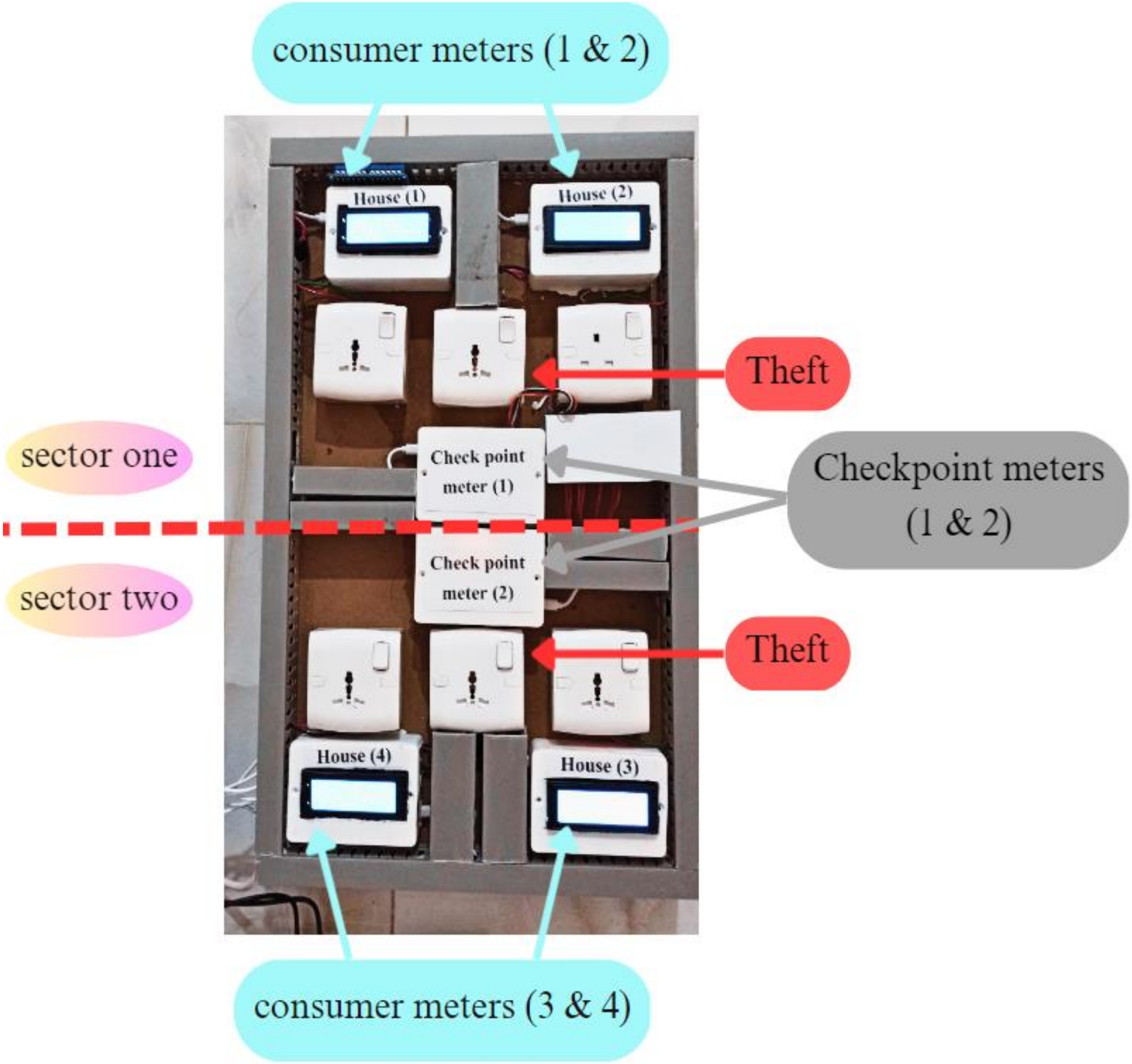


Figure 4.11:A Prototype hardware arrangement for proposed system.

The process of detecting electricity theft involves sequentially checking each sector, starting from the first sector and moving on to the next sector for accurate detection. This sequential approach helps in narrowing down the search for theft, minimizing the impact on legitimate consumers, and ensuring a more efficient detection process.

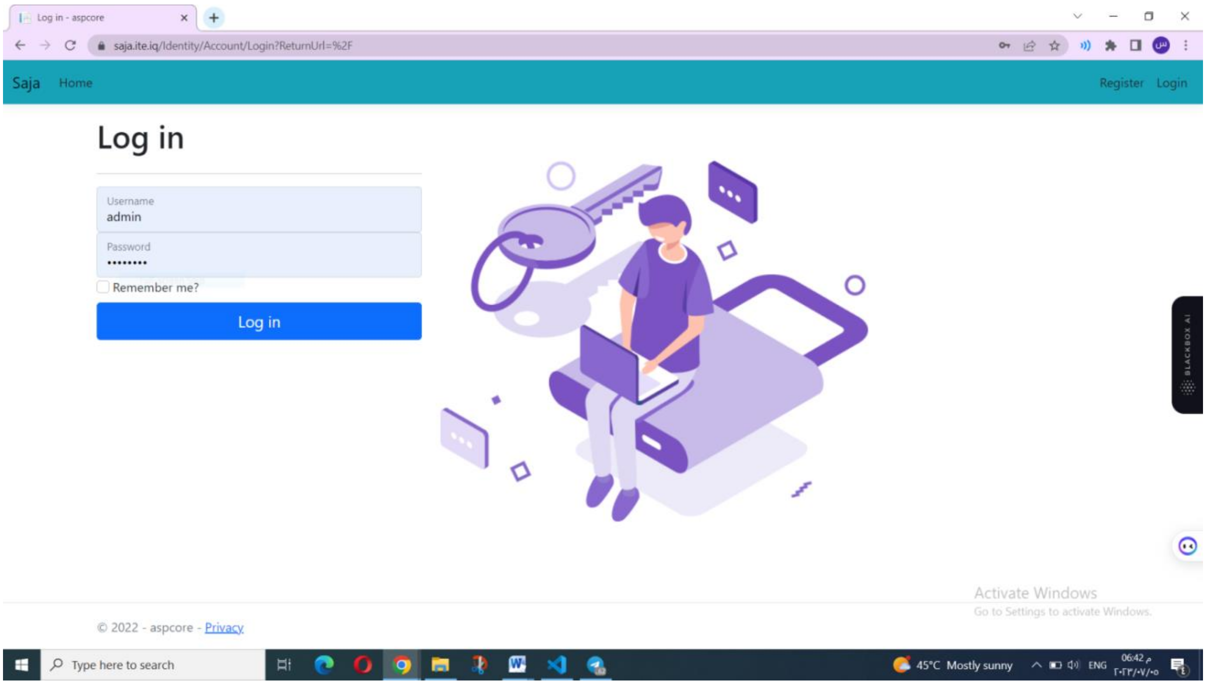
These meters send the collected data to the cloud server via HTTPS. The cloud server chosen for this purpose has the following specifications:

- CPU: 2 cores, each operating at 2.6 GHz.
- Operating System: Windows Server 2019
- Bandwidth: 8TB
- RAM: 4GB of memory to support the server's processing requirements.
- Storage:60GB SSD, offering faster read/write speeds compared to traditional HDDs.
- Public IP: The server has a public IP address, allowing it to be accessed from the internet.

The cloud server provides space to execute theft detection models (mathematical model described in section 3.3.1 and a CNN model). It also offers secure storage for consumer data. The server ensures data privacy as it is a private server not shared with others. This dedicated setup allows for efficient and confidential handling of the data, ensuring that it remains solely accessible to authorized parties.

The application is built using ASP.NET Core MVC on a local server. Once the development of the web application is complete, it needs to be deployed to a cloud server. For deployment, the Internet Information Services (IIS) software is required to be installed on the cloud server. The IIS software facilitates the hosting and management of web applications on the cloud server, allowing the application to be accessible over the internet.

The web-based app can be accessed through a valid username and password, limited to authorized administrators only, as shown in Figure 4.12.



Username
admin

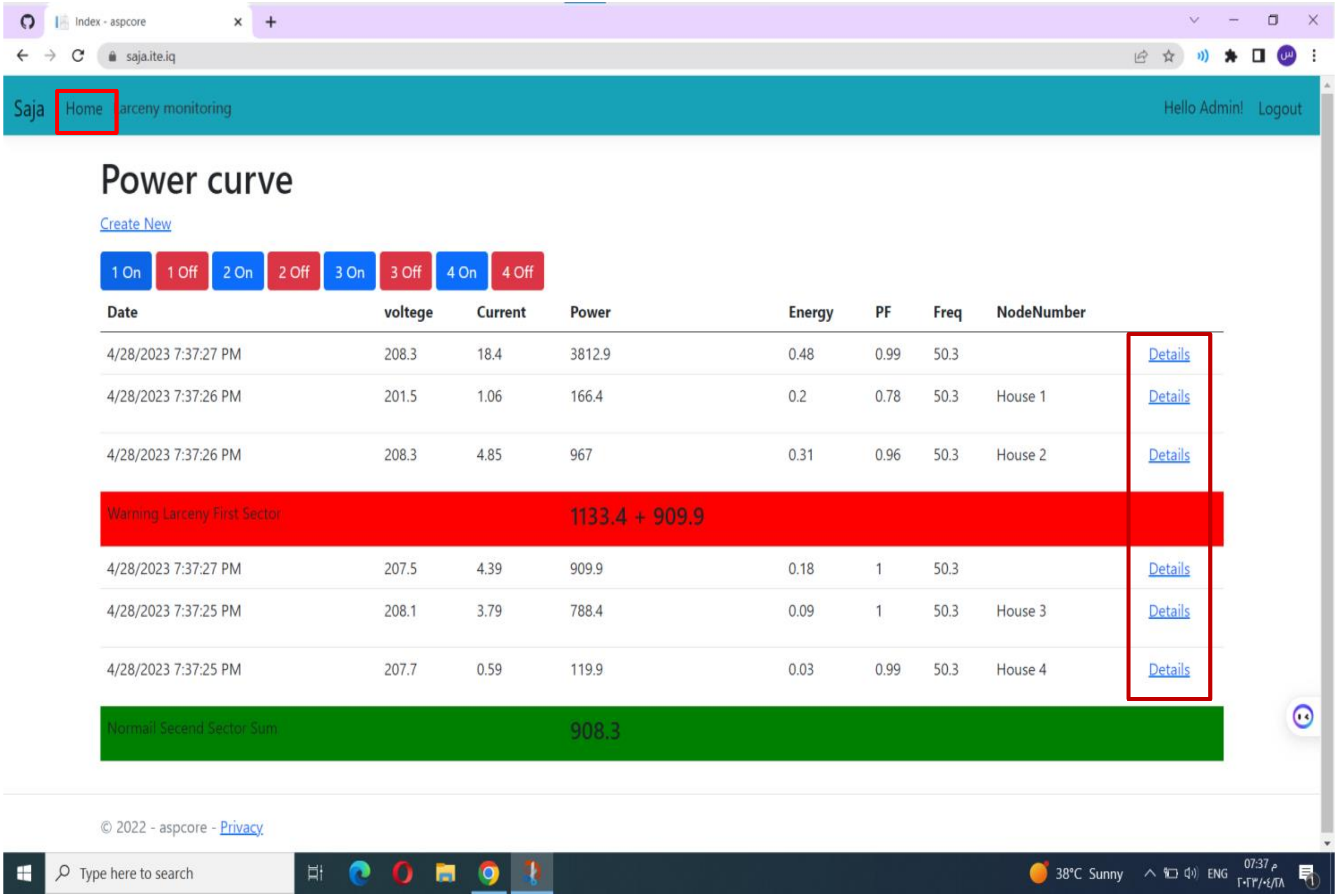
Password
••••••••

□ Remember me?

Figure 4.12: Login page

The data received is processed by the ASP.NET Core framework. Then through this framework data is organized and stored in the SQL Server database. The ASP.NET Core framework communicates with the database through Entity Framework technology for efficient data management and retrieval.

The cloud server obtains data from the SQL server upon user login and displays it on the application's main interface. The Home interface of the application (see Figure 4.13) displays real-time readings of all smart meters, along with the results of the mathematical model used for detecting electricity larceny. A green indicator represents normal power usage, while a red is activated when the power consumption discrepancy between the homes and the distribution node exceeds 20%, suggesting a potential case of energy larceny.



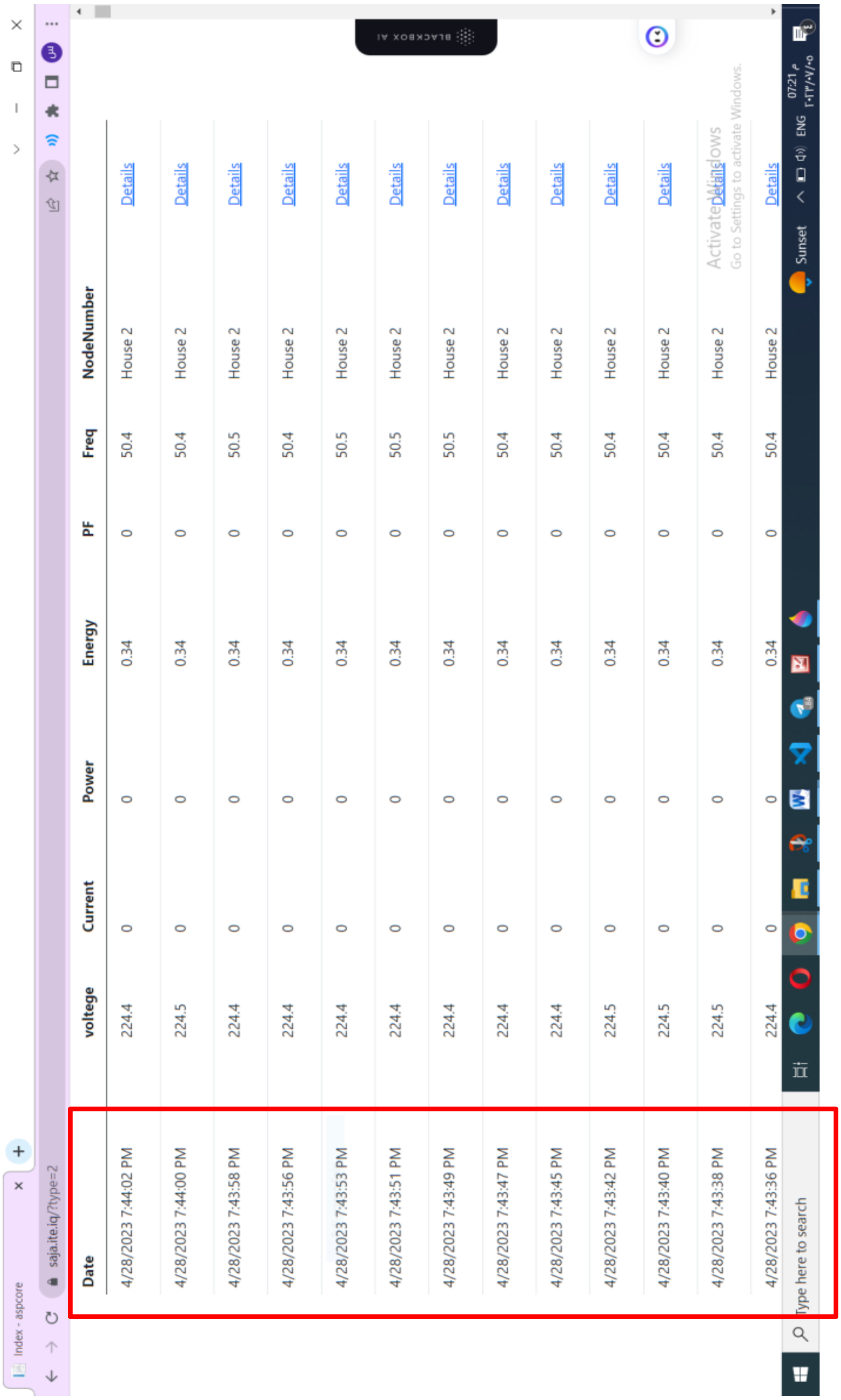
Power curve

Date	voltege	Current	Power	Energy	PF	Freq	NodeNumber	
4/28/2023 7:37:27 PM	208.3	18.4	3812.9	0.48	0.99	50.3		Details
4/28/2023 7:37:26 PM	201.5	1.06	166.4	0.2	0.78	50.3	House 1	Details
4/28/2023 7:37:26 PM	208.3	4.85	967	0.31	0.96	50.3	House 2	Details
Warning Larceny First Sector			1133.4 + 909.9					
4/28/2023 7:37:27 PM	207.5	4.39	909.9	0.18	1	50.3		Details
4/28/2023 7:37:25 PM	208.1	3.79	788.4	0.09	1	50.3	House 3	Details
4/28/2023 7:37:25 PM	207.7	0.59	119.9	0.03	0.99	50.3	House 4	Details
Normail Secend Sector Sum			908.3					

© 2022 - aspcore - Privacy

Figure 4.13: Home interface of the application

The "Details" feature provides a historical record of readings for a selected house. By observing the "Date" column in Figure 4.15, it can be noticed that the readings are updated approximately every (2-3) seconds.



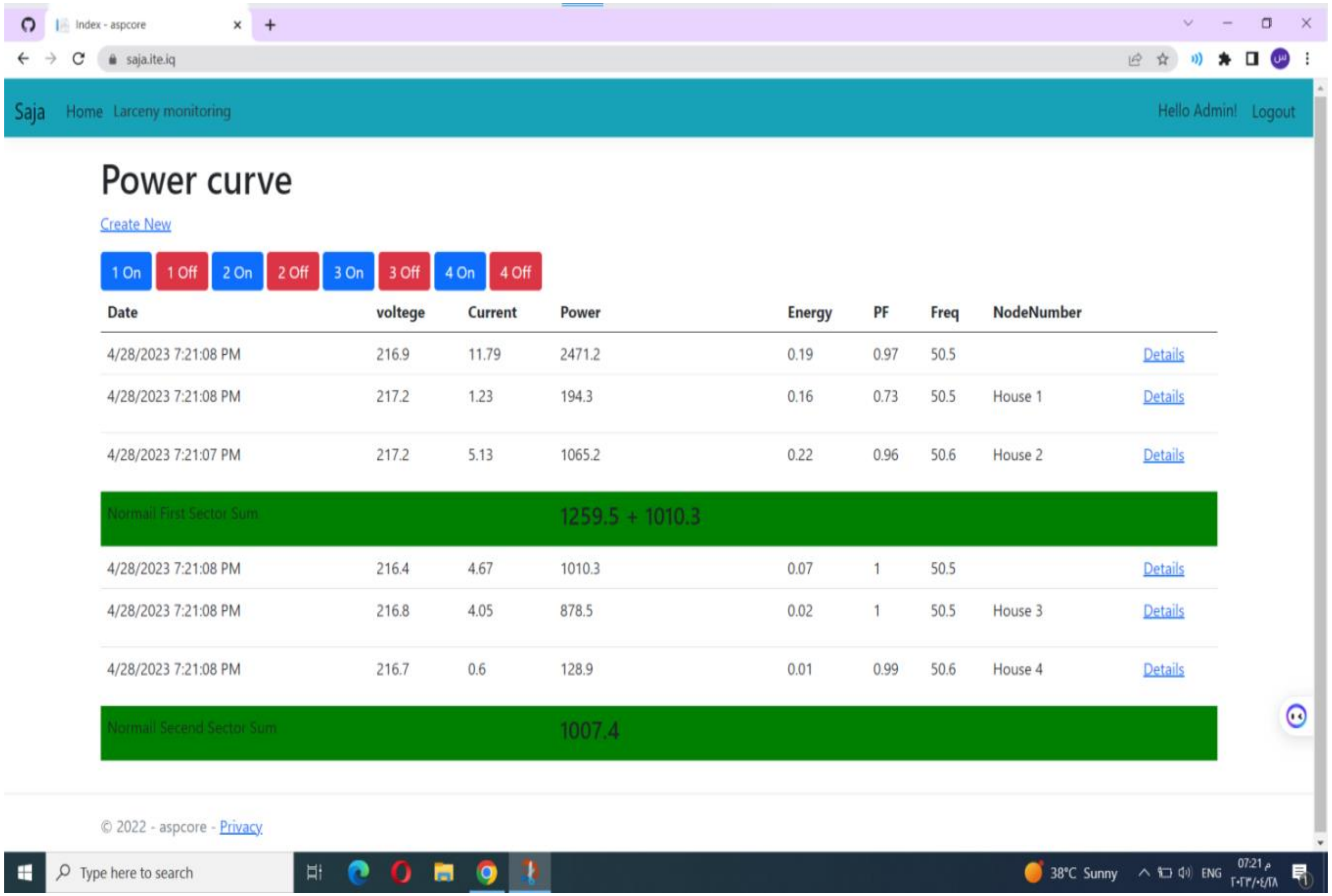
Date	voltege	Current	Power	Energy	PF	Freq	NodeNumber	
4/28/2023 7:44:02 PM	224.4	0	0	0.34	0	50.4	House 2	Details
4/28/2023 7:44:00 PM	224.5	0	0	0.34	0	50.4	House 2	Details
4/28/2023 7:43:58 PM	224.4	0	0	0.34	0	50.5	House 2	Details
4/28/2023 7:43:56 PM	224.4	0	0	0.34	0	50.4	House 2	Details
4/28/2023 7:43:53 PM	224.4	0	0	0.34	0	50.5	House 2	Details
4/28/2023 7:43:51 PM	224.4	0	0	0.34	0	50.5	House 2	Details
4/28/2023 7:43:49 PM	224.4	0	0	0.34	0	50.5	House 2	Details
4/28/2023 7:43:47 PM	224.4	0	0	0.34	0	50.4	House 2	Details
4/28/2023 7:43:45 PM	224.4	0	0	0.34	0	50.4	House 2	Details
4/28/2023 7:43:42 PM	224.5	0	0	0.34	0	50.4	House 2	Details
4/28/2023 7:43:40 PM	224.5	0	0	0.34	0	50.4	House 2	Details
4/28/2023 7:43:38 PM	224.5	0	0	0.34	0	50.4	House 2	Details
4/28/2023 7:43:36 PM	224.4	0	0	0.34	0	50.4	House 2	Details

Figure 4.15: Details of House 2 (Historical Readings)

The following scenarios were tested in the system:

A. Normal case:

The power consumption readings from the checkpoint meter are compared to the aggregate readings of power consumption of individual consumers' meters in this sector, and if the difference is below the predetermined threshold value, it indicates normal power consumption, as shown in Figure 4.16.



Power curve

Create New

1 On | 1 Off | 2 On | 2 Off | 3 On | 3 Off | 4 On | 4 Off

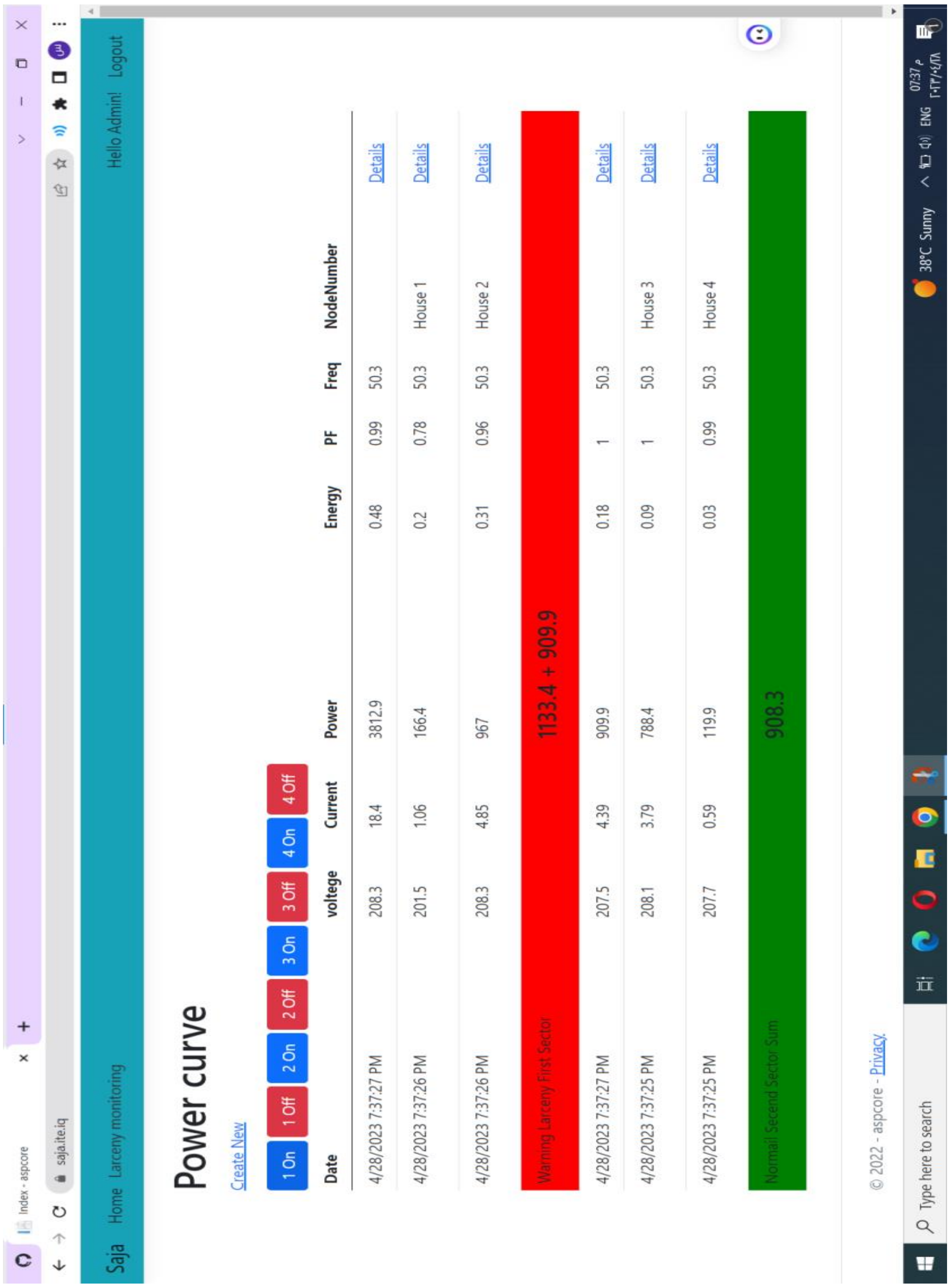
Date	voltege	Current	Power	Energy	PF	Freq	NodeNumber	
4/28/2023 7:21:08 PM	216.9	11.79	2471.2	0.19	0.97	50.5		Details
4/28/2023 7:21:08 PM	217.2	1.23	194.3	0.16	0.73	50.5	House 1	Details
4/28/2023 7:21:07 PM	217.2	5.13	1065.2	0.22	0.96	50.6	House 2	Details
Normail First Sector Sum			1259.5 + 1010.3					
4/28/2023 7:21:08 PM	216.4	4.67	1010.3	0.07	1	50.5		Details
4/28/2023 7:21:08 PM	216.8	4.05	878.5	0.02	1	50.5	House 3	Details
4/28/2023 7:21:08 PM	216.7	0.6	128.9	0.01	0.99	50.6	House 4	Details
Normail Secend Sector Sum			1007.4					

© 2022 - aspcore - Privacy

Figure 4.16: Normal power consumption case.

B.Electricity Larceny By Direct Line Tapping:

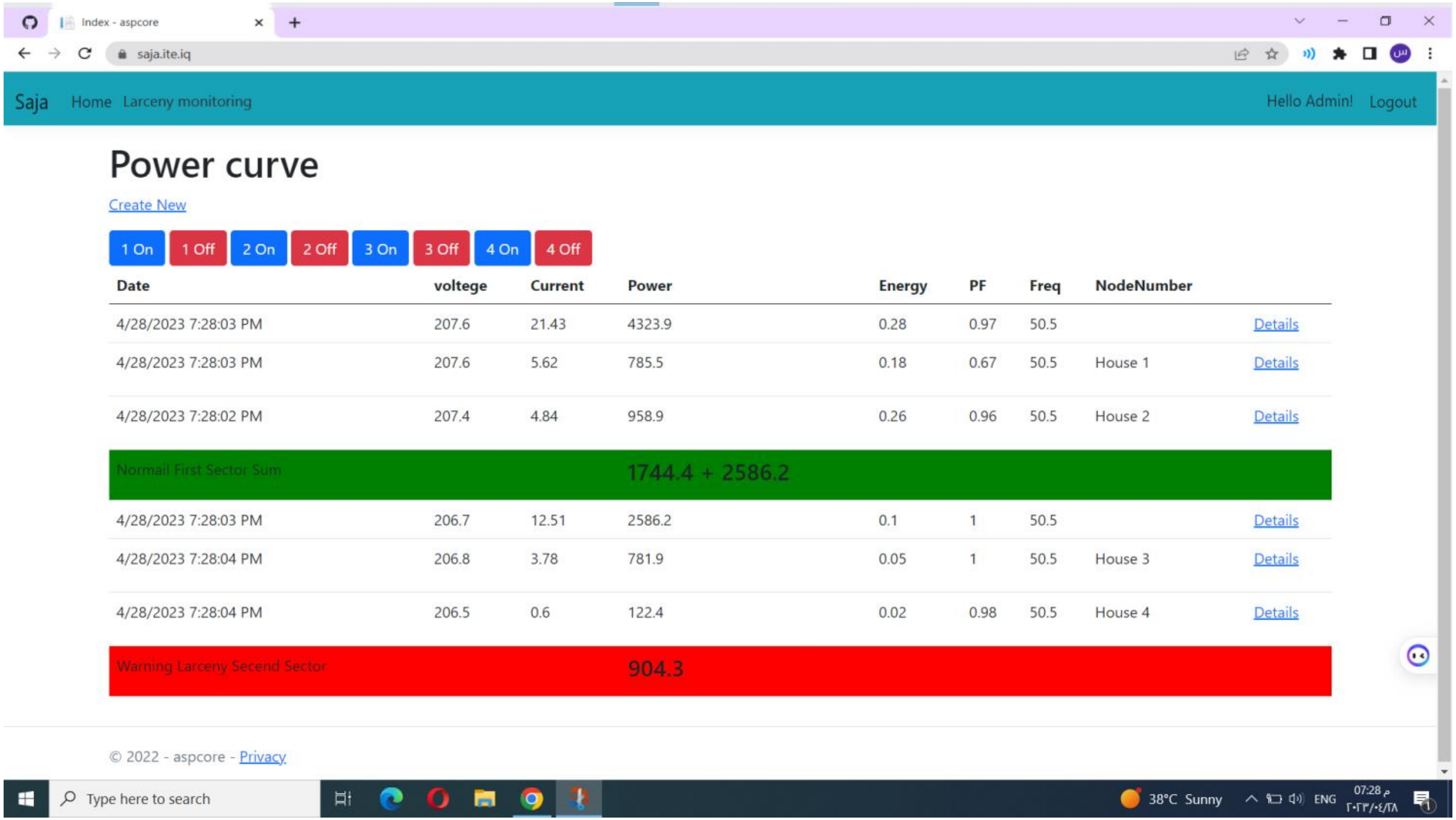
If the difference between the power consumption readings from the checkpoint meter and that of individual consumers' meters exceeds the predetermined threshold value in this sector, it indicates the presence of energy larceny. The location of the theft can be identified as shown in Figure 4.17, where Figure 4.17 (a) shows theft in the first sector, Figure 4.17 (b) shows theft in the second sector, and Figure 4.17 (c) shows theft in both sectors.



Power curve

Date	voltege	Current	Power	Energy	PF	Freq	NodeNumber	
4/28/2023 7:37:27 PM	208.3	18.4	3812.9	0.48	0.99	50.3		Details
4/28/2023 7:37:26 PM	201.5	1.06	166.4	0.2	0.78	50.3	House 1	Details
4/28/2023 7:37:26 PM	208.3	4.85	967	0.31	0.96	50.3	House 2	Details
Warning Larceny First Sector			1133.4 + 909.9					
4/28/2023 7:37:27 PM	207.5	4.39	909.9	0.18	1	50.3		Details
4/28/2023 7:37:25 PM	208.1	3.79	788.4	0.09	1	50.3	House 3	Details
4/28/2023 7:37:25 PM	207.7	0.59	119.9	0.03	0.99	50.3	House 4	Details
Normail Secend Sector Sum			908.3					

Figure 4.17 (a) Electricity Larceny detection in the first sector

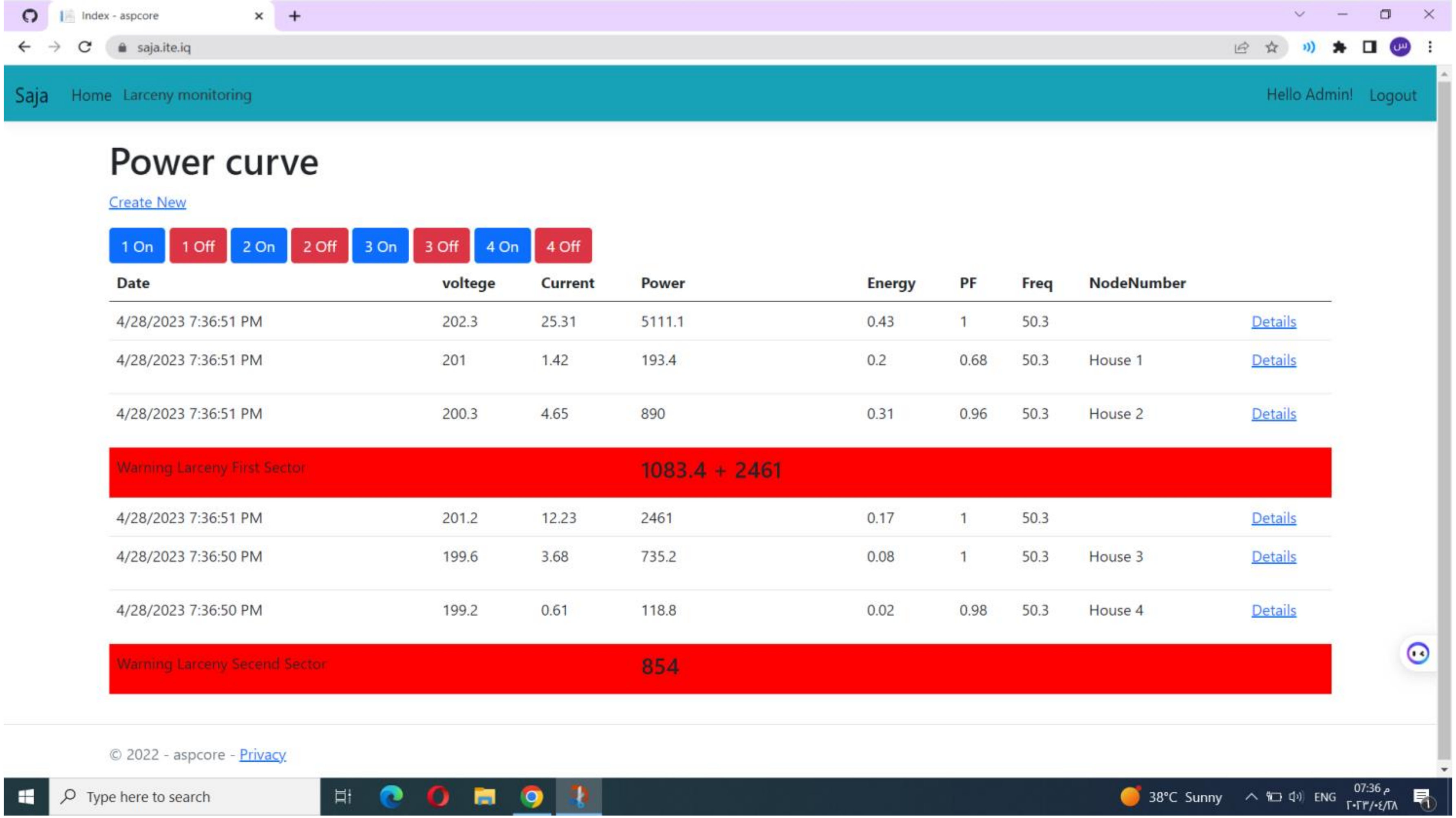


Power curve

Date	voltege	Current	Power	Energy	PF	Freq	NodeNumber	
4/28/2023 7:28:03 PM	207.6	21.43	4323.9	0.28	0.97	50.5		Details
4/28/2023 7:28:03 PM	207.6	5.62	785.5	0.18	0.67	50.5	House 1	Details
4/28/2023 7:28:02 PM	207.4	4.84	958.9	0.26	0.96	50.5	House 2	Details
Normail First Sector Sum			1744.4 + 2586.2					
4/28/2023 7:28:03 PM	206.7	12.51	2586.2	0.1	1	50.5		Details
4/28/2023 7:28:04 PM	206.8	3.78	781.9	0.05	1	50.5	House 3	Details
4/28/2023 7:28:04 PM	206.5	0.6	122.4	0.02	0.98	50.5	House 4	Details
Warning Larceny Secend Sector			904.3					

© 2022 - aspcore - Privacy

Figure 4.17 (b) Electricity Larceny detection in the second sector



Power curve

Create New

1 On | 1 Off | 2 On | 2 Off | 3 On | 3 Off | 4 On | 4 Off

Date	voltege	Current	Power	Energy	PF	Freq	NodeNumber	
4/28/2023 7:36:51 PM	202.3	25.31	5111.1	0.43	1	50.3		Details
4/28/2023 7:36:51 PM	201	1.42	193.4	0.2	0.68	50.3	House 1	Details
4/28/2023 7:36:51 PM	200.3	4.65	890	0.31	0.96	50.3	House 2	Details
Warning Larceny First Sector			1083.4 + 2461					
4/28/2023 7:36:51 PM	201.2	12.23	2461	0.17	1	50.3		Details
4/28/2023 7:36:50 PM	199.6	3.68	735.2	0.08	1	50.3	House 3	Details
4/28/2023 7:36:50 PM	199.2	0.61	118.8	0.02	0.98	50.3	House 4	Details
Warning Larceny Secend Sector			854					

© 2022 - aspcore - Privacy

Figure 4.17 (c) Electricity Larceny detection in both sectors

C. Energy Larceny By Meter Tampering or fraud:

The CNN model is utilized to identify this specific type of larceny. It analyzes the historical readings obtained from the SQL server to ascertain the load patterns of individual houses. Based on this analysis, the houses are categorized into two groups: normal or indicative of larceny. The findings from the analysis are displayed through a web application, as shown in Figure 4.18.

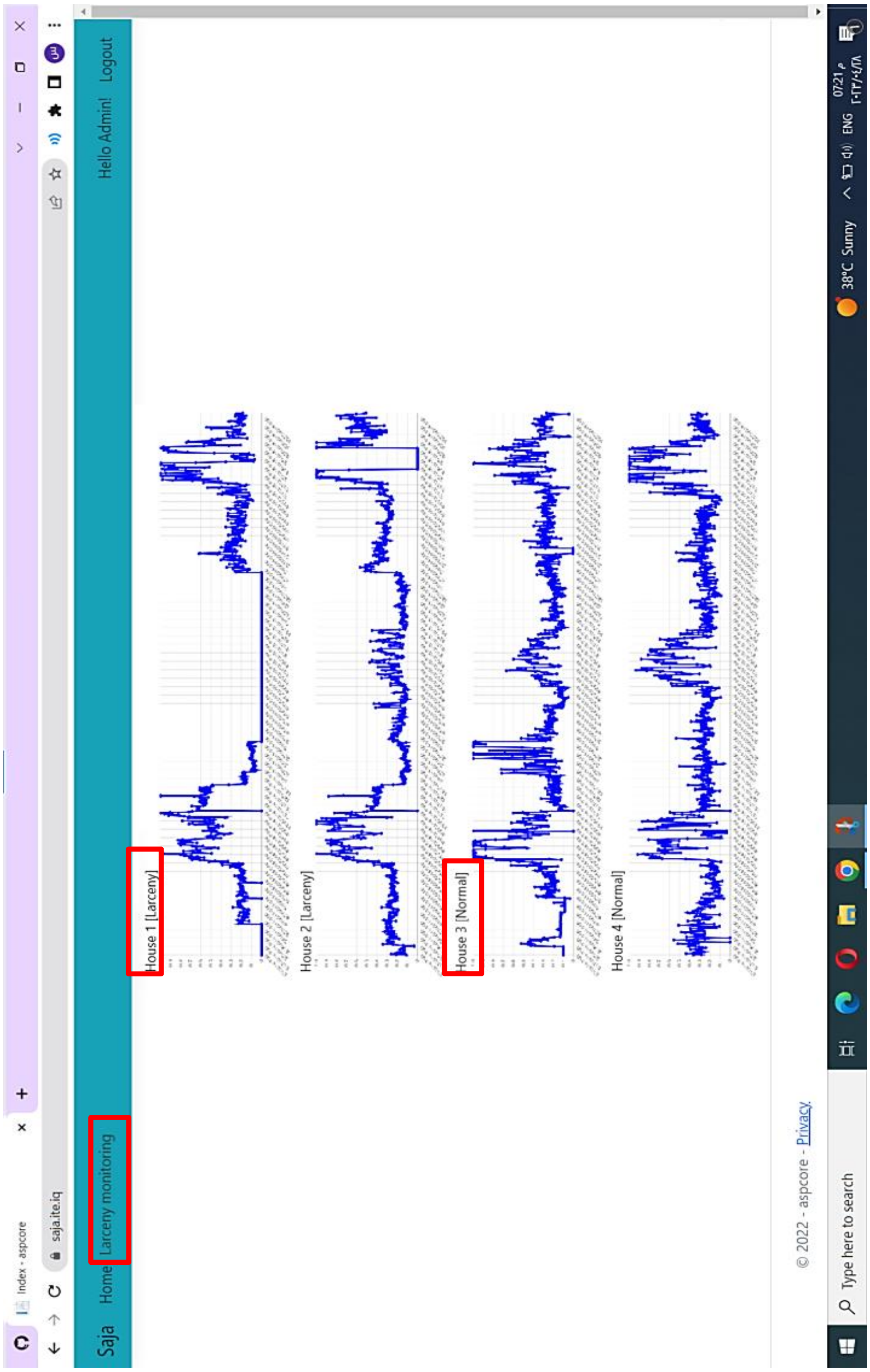


Figure 4.18: Home load patterns Larceny monitoring interface.

4.5 Features of the Home Electricity Larceny Detection System

The system has the following features:

1. The system employs advanced algorithms and data analytics to detect and identify instances of power theft with high accuracy. By analyzing consumption patterns and discrepancies in meter readings, the system promptly alerts utility providers to potential theft, enabling timely action to rectify the situation.
2. One of the key aspects of the system is its ability to facilitate a seamless transition from traditional conventional meters to advanced smart meters. These intelligent devices bring enhanced functionality and capabilities to the table, such as real-time monitoring, remote reading, and bidirectional communication with the utility company.
3. With centralized monitoring capabilities, the system offers real-time data on power consumption across distribution networks.
4. By effectively detecting and curbing power theft instances, the system significantly reduces revenue losses experienced by utility companies. This leads to increased revenue collection and financial stability.
5. Centralized monitoring of power distribution lines ensures a better understanding of grid health and stability. Early detection of potential issues, such as overloading or unauthorized connections, allows utility companies to take preventive measures, minimizing disruptions and enhancing grid reliability.
6. The system streamlines processes through centralized data management and cloud-based infrastructure. This results in improved operational efficiency.
7. The system effectively tackles non-technical losses resulting from fraudulent activities by users.

4.6 Random Forest Algorithm for Power Factor Correction

The Random Forest data table was created from the power and power factor of the appliances from Table 3.3. This data was created to train and test the Random Forest algorithm. The appliances in Table 3.3 are switched alternately to give 1023 cases of actual reading power and power factor.

To calculate the total number of possible combinations, we can use the concept of combinations in combinatorics. The number of combinations of selecting L items from a set of t items is given by the binomial coefficient formula:

$$C(t,L) = \frac{t!}{(L!\,(t - L)!)} \qquad 4.1$$

In this case, we have 10 appliances (t = 10) and we want to calculate the total number of combinations when selecting any number of appliances from 1 to 10 (L = 1 to 10).

$$\mathrm{C}(10,1) = \frac{10!}{(1! * 9!)} = 10$$

$$\mathrm{C}(10,2) = \frac{10!}{(2! * 8!)} = 45$$

$$\mathrm{C}(10,3) = \frac{10!}{(3! * 7!)} = 120$$

$$\mathrm{C}(10,4) = \frac{10!}{(4! * 6!)} = 210$$

$$\mathrm{C}(10,5) = \frac{10!}{(5! * 5!)} = 252$$

$$\mathrm{C}(10,6) = \frac{10!}{(6! * 4!)} = 210$$

$$\mathrm{C}(10,7) = \frac{10!}{(7! * 3!)} = 120$$

$$\mathrm{C}(10,8) = \frac{10!}{(8! * 2!)} = 45$$

$$\mathrm{C}(10,9) = \frac{10!}{(9! * 1!)} = 10$$

$$C(10,10) = \frac{10!}{(10! * 0!)} = 1$$

Calculating this sum will give us the total number of possible combinations when selecting from the given set of 10 appliances:

$$Total\ combinations = 10 + 45 + 120 + 210 + 252 + 210 + 120 + 45 + 10 + 1 = 1023.$$

The required capacitance value was calculated using the equations (2.4) and (2.5) to get the supposed power factor of 0.95 as desired value. Fifteen levels of actual hardware capacitance values are presented from a combination of (2,4,8,16) µF. These levels as (2,4,6,8,10,.....,30) µF and in numbers (1,2,3,........,15).

The task of the Random Forest algorithm circumscribes the level of capacitance nearest or the same value from mentioned levels to give real power factor correction. Table 4.10 show the first twenty row of the data.

Table 4.10 Power and power factor and capacitance data

	Load device	Power (w)	Pf	desired pf	Required VAR	Required C (µF)	Real C(µF)	Level	Pf for real C
1	LED light lamp A	30	0.55	0.95	35.693	2.35	2	1	0.957
2	LED light Ingco: B	80	0.58	0.95	86.066	5.66	6	3	0.946
3	Stand Fan: C	55	0.82	0.95	20.312	1.34	2	1	0.986
4	Freezer: D	155	0.46	0.95	248.244	16.33	16	8	0.906
5	Fridge: E	237	0.64	0.95	206.640	13.59	14	7	0.966
6	LED light: F	140	0.55	0.95	166.571	10.95	10	5	0.951
7	Ps4 pro: G	88	0.78	0.95	41.676	2.74	2	1	0.994
8	LED light Aswar: H	64	0.55	0.95	76.147	5.01	6	3	0.920
9	Random	266	0.67	0.95	207.298	13.63	14	7	0.972

	home loads 1: I								
10	Random home loads 2: J	181	0.73	0.95	109.965	7.23	8	4	0.980
11	A + B	110	0.571	0.95	121.760	8.01	8	4	0.949
12	A + C	85	0.711	0.95	56.006	3.68	4	2	0.978
13	A + D	185	0.472	0.95	283.937	18.67	18	9	0.915
14	A + E	267	0.628	0.95	242.334	15.94	16	8	0.965
15	A + F	170	0.550	0.95	202.265	13.3	14	7	0.937
16	A + G	118	0.712	0.95	77.370	5.09	6	3	0.974
17	A + H	94	0.550	0.95	111.840	7.36	8	4	0.933
18	A + I	296	0.656	0.95	242.992	15.98	16	8	0.971
19	A + J	211	0.700	0.95	145.659	9.58	10	5	0.977
20	B + C	135	0.667	0.95	106.378	7	8	4	0.965

The data in Table 4.10 is divided into a training set and a test set. The Random Forest (RF) classifier is trained using the following hyperparameters:

- **n_estimators**: The forest consists of ten trees. Each tree is constructed using a random subset of the training data. Having multiple trees helps improve the model's accuracy and robustness.

- **max_depth**: The maximum depth of each tree is not explicitly limited(is set to None). This allows the trees to grow until all the leaf nodes are pure (containing only instances of a single class) or until they contain fewer samples than the 'min_samples_split'.

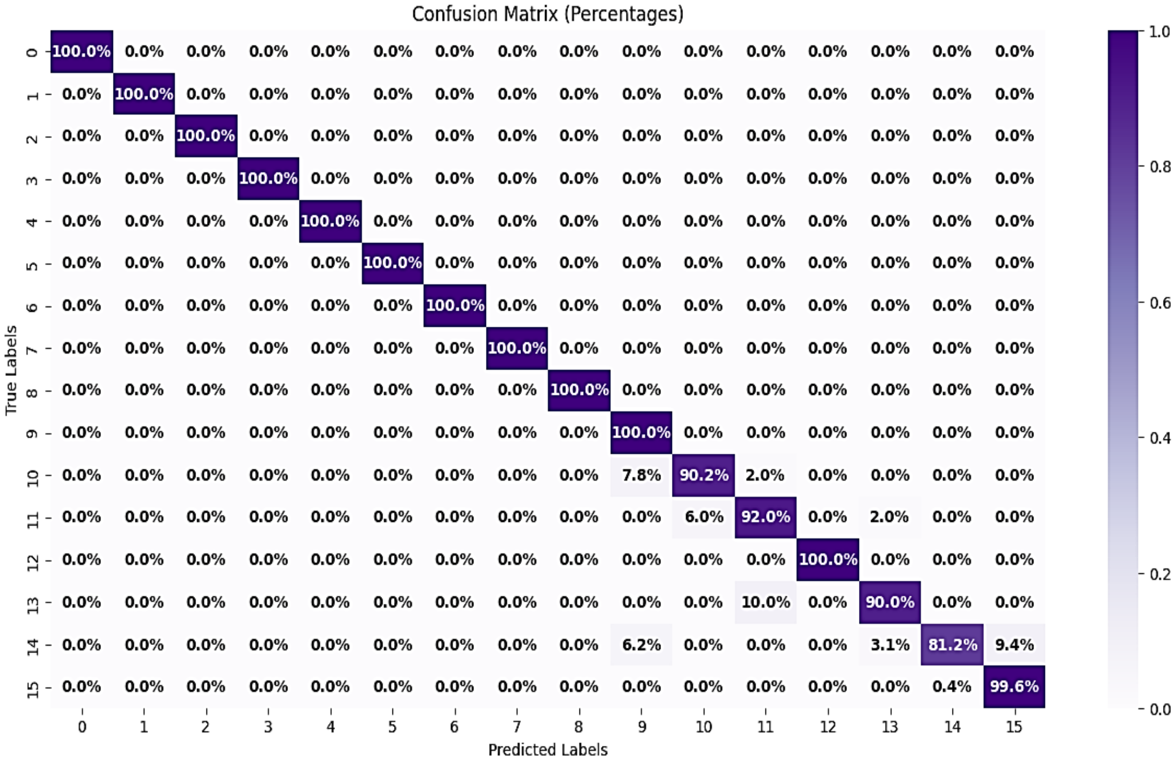
- **min_samples_split**: This parameter determines the minimum number of samples required to split an internal node in a tree. In this case, it is set to 2. If a node has fewer than 2 samples, it will not be split further and becomes a leaf node.

After training, the model is evaluated on the test set to assess its performance. The evaluation results are as follows:

Table 4.11 Performance of Random Forest for PFC

Accuracy	F1 score	Precision	Recall
97.93%	97.91%	97.98%	97.93%

Additionally, the confusion matrix is depicted in Figure 4.19.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
1	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
2	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
3	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
4	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
5	0.0%	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
6	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
7	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
8	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
9	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
10	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	7.8%	90.2%	2.0%	0.0%	0.0%	0.0%	0.0%
11	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	6.0%	92.0%	0.0%	2.0%	0.0%	0.0%
12	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%
13	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	10.0%	0.0%	90.0%	0.0%	0.0%
14	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	6.2%	0.0%	0.0%	0.0%	3.1%	81.2%	9.4%
15	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.4%	99.6%

Figure 4.19: confusion matrix of RF for power factor correction

4.7 The Error of Power Factor

After constructing a table for the power factor, it is important to assess the data quality on which the random forest was trained.

Table 4.12 displays the power factor error, average pf error, and average power factor.

Table 4.12: Power factor error and capacitance error.

ID	Load device	desired Pf Pf_d	Calculate C (µF)	Real C(µF)	Pf for real C Pf_r	Pf Error approximate = $\frac{pf_d - pf_r}{pf_d}$	C error in µF = $C\ calc - C\ real$
1	LED light lamp A	0.95	2.35	2	0.957	0	0.35
2	LED light Ingco: B	0.95	5.66	6	0.946	0. 3182224%	0.34
3	Stand Fan: C	0.95	1.34	2	0.986	0	0.66
4	Freezer: D	0.95	16.33	16	0.906	4.6238494%	0.33
5	Fridge: E	0.95	13.59	14	0.966	0	0.41
6	LED light: F	0.95	10.95	10	0.951	0	0.95
7	Ps4 pro: G	0.95	2.74	2	0.994	0	0.74
8	LED light Aswar: H	0.95	5.01	6	0.920	3.0922157%	0.99
9	1. Random home loads: I	0.95	13.63	14	0.972	0	0.37
10	2. Random home loads: J	0.95	7.23	8	0.980	0	0.77
11	A + B	0.95	8.01	8	0.949	0.000968343%	0.01
12	A + C	0.95	3.68	4	0.978	0	0.32
13	A + D	0.95	18.67	18	0.915	3.656825%	0.67
14	A + E	0.95	15.94	16	0.965	0	0.06
15	A + F	0.95	13.3	14	0.937	1.3600003%	0.7
16	A + G	0.95	5.09	6	0.974	0	0.91
17	A + H	0.95	7.36	8	0.933	1.7661607%	0.64
18	A + I	0.95	15.98	16	0.971	0	0.02
19	A + J	0.95	9.58	10	0.977	0	0.4
20	B + C	0.95	7	8	0.965	0	1
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
1023	A + B + C + D + E + F + G + H + I +	0.95	37.29	30	0.996	0	0.35

	J						
Average pf					0.973		
Average pf error						0.65%	
Average capacitance error							0.67 µF

As stated in Table 4.3, the achieved average power factor is 0.973. The power factor error is calculated for the values lower than the desired power factor, which represents the percentage difference between the desired power factor and the achieved power factor if it is less than the desired power factor, using the power factor error formula [(desired - actual) / desired]. The average power factor error, or average pf error, was reported to be 0.65% is lower than the value mentioned in reference [84], where it was reported to be 2.65%.

The capacitance error refers to the average difference in capacitance between the computed and actual values, and it is 0.67 μF is lower than the value mentioned in reference [84], where it was reported to be 1.428 μF. This suggests an improvement in accurately determining the required capacity to correct the power factor. This value reflects the average variance in determining the required capacity to correct the power factor.

4.8 Features of Power Factor Correction System

The proposed system has many features as follows:

1. The system is specifically designed to be implemented in typical residential settings where appliances have internet access. This ensures compatibility and seamless integration with existing residential infrastructure.
2. Multi-Home Power Factor Correction: The system supports power factor correction for multiple homes through a centralized device. This allows for efficient power factor management and optimization across multiple households, leading to improved energy efficiency.
3. Scalability with Cloud Concept: The system leverages cloud computing to support scalability and handle large amounts of data. This enables the

system to accommodate a growing number of residents and effectively manage power quality improvements.

4. Accurate Power Factor Correction: The system includes an accurate determination mechanism for capacitance-based power factor correction. This ensures precise and efficient correction of power factor imbalances, resulting in optimized energy consumption.
5. Real-Time Monitoring: The system enables real-time monitoring of power consumption and correction processes.
6. Support for Smart Grid Concept: The system aligns with the principles of the smart grid concept. It integrates with the broader energy infrastructure, allowing for seamless communication and coordination between the proposed system and other smart grid components.

5

CHAPTER FIVE

CONCLUSIONS AND FUTURE WORK

Chapter Five

Future Works and Conclusions

5.1 Conclusions

In this thesis, various disciplines such as software and server programming, control, electronics, communications, and Artificial Intelligence techniques were explored in the development and implementation of the proposed systems. Through this process, abundant knowledge has been acquired. The following points were conclude:

1. Using the cloud server provides storage space and data processing capabilities, ensuring secure handling of data. The dedicated server efficiently accommodates the large volume of data from smart meters and can easily scale resources to handle any increase, saving storage space while ensuring reliable and secure data management.
2. A hybrid approach was utilized for accurate electricity larceny detection, combine of hardware-oriented and data-oriented methods.
3. Tests were conducted on the hardware-oriented approach, and it proved to be effective in real-time electricity larceny detection. Furthermore, the approach contributed to identifying the location of the larceny approximately.
4. The data-oriented approach achieved excellent results in the accurate detection of electricity larceny. Here is a summary of the results:
   - The CNN model was trained for ten epochs using original data without employing balancing techniques, leading to a testing accuracy of 93.18%.
   - The combination of CNN model and RUS (Random Under-Sampling) technique achieved a testing accuracy of 79.61%. The decrease in accuracy is attributed to the loss of information that

CNN model can learn from, which occurs due to the random removal of data points as a result of applying RUS technique.

- The combination of CNN model and ROS (Random Over-Sampling) technique achieved a testing accuracy of 98.33%. However, this high accuracy may be a result of overfitting. Overfitting occurs when the CNN model memorizes the training data rather than learning the underlying patterns and relationships in the data.
- The combination of CNN model and SMOTE (Synthetic Minority Over-sampling Technique) achieved a testing accuracy of 94.70%, which is considered a relatively good result.
- The combination of CNN model and SMOTETomek technique achieved a testing accuracy of 95.93%, surpassing the performance of CNN model alone and CNN model with SMOTE technique.
- The combination of the CNN model and the ADASYN technique achieved a testing accuracy of 97.22%, outperforming all other techniques, this suggests that ADASYN is a reliable choice for addressing class imbalance and improving the model's performance.

5. The Random Forest algorithm achieved a testing accuracy of 97.93% in identifying the appropriate capacitors to correct the power factor. This high accuracy indicates that the Random Forest algorithm is effective in selecting the suitable capacitors for power factor correction, making it a reliable and efficient method for this task.
6. The average power factor correction error based on real and decided capacitance via the Random Forest power factor correction algorithm was 0.67%.
7. The power factor corrected from 0.41 to 0.95 with an average power factor of 0.973.
8. The average capacitance error is 0.67 µF this indicates that the measurements are closer to the desired values.

5.2 Researcher testaments for Future Works

The following points provide suggestions for future work:

1. Implementing dynamic pricing and demand management with the help of cloud computing.
2. Analyzing harmonic effects and losses with the assistance of cloud computing.
3. Creating a cloud-based system for labeling appliances with power-saving agreements.
4. Developing a cloud-based system for analyzing load patterns in hybrid sourced (renewable and traditional) energy systems.
5. Expanding the larceny detection system to calculate the cost of frauded energy.

References

[1] M. Jeffin, G. Madhu, A. Rao, G. Singh, and C. Vyjayanthi, "Internet of Things Enabled Power Theft Detection and Smart Meter Monitoring System," in *2020 International Conference on Communication and Signal Processing (ICCSP)*, 2020: IEEE, pp. 0262-0267.

[2] M. Pampalle, S. Srikanth, V.C.J. Mohan, and T.R. Babu, "IOT-Based Automatic Power Factor Correction (APFC) Unit for Industries to Curtail the Retribution of Power Utility Companies," in *Proceedings of SSIC 2021: Smart Systems: Innovations in Computing, Singapore, Springer Singapore*, 2021, pp. 485-497.

[3] M. A. Al Rakib, S. Nazmi, M. H. Imam, and M. N. Uddin, "Arduino based automatic power factor control," *International Journal of Smart Grid*, vol. 5, no. 3, 2021.

[4] F. d. S. Savian, J. C. M. Siluk, T. B. Garlet, F. M. do Nascimento, J. R. Pinheiro, and Z. Vale, "Non-technical losses: A systematic contemporary article review," *Renewable and Sustainable Energy Reviews*, vol. 147, pp. 1-13, 2021.

[5] P. Glauner, J. A. Meira, P. Valtchev, R. State, and F. Bettinger, "The challenge of non-technical loss detection using artificial intelligence: A survey," *arXiv preprint arXiv*:1606.00626, 2016.

[6] A. J. Pansini, "Power Transmission & Distribution, Second Edition," 2nd ed. River Publishers, 2005. [Online]. Available: https://doi.org/10.1201/9781003151203.

[7] L. Feng, S. Xu, L. Zhang, J. Wu, J. Zhang, C. Chu, et al., "Anomaly detection for electricity consumption in cloud computing: framework methods applications and challenges", *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, pp. 1-12, 2020.

[8] S. Rusitschka, K. Eger and C. Gerdes, "Smart Grid Data Cloud: A Model for Utilizing Cloud Computing in the Smart Grid Domain," 2010 *First IEEE*

International Conference on Smart Grid Communications, Gaithersburg, MD, USA, 2010, pp. 483-488, doi: 10.1109/SMARTGRID.2010.5622089.

[9] A. Rashid and A. Chaturvedi, "Cloud computing characteristics and services: a brief review," *International Journal of Computer Sciences and Engineering*, vol. 7, no. 2, pp. 421-426, 2019.

[10] M. I. Malik, S. H. Wani, and A. Rashid, "CLOUD COMPUTING-TECHNOLOGIES," *International Journal of Advanced Research in Computer Science*, vol. 9, no. 2, pp. 1-8, 2018.

[11] S. Bera, S. Misra and J. Rodrigues, "Cloud Computing Applications for Smart Grid: A Survey," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 5, pp. 1477-1494, 2015.

[12] T. Kotsiopoulos, P. Sarigiannidis, D. Ioannidis, and D. Tzovaras, "Machine Learning and Deep Learning in smart manufacturing: The Smart Grid paradigm", *Computer Science Review*, Vol. 40, PP. 100341. 2021.

[13] E. Hossain, I. Khan, F. Un-Noor, S. Sikander and M. Sunny, "Application of Big Data and Machine Learning in Smart Grid, and Associated Security Concerns: A Review," *IEEE Access*, vol. 7, pp. 13960-13988, 2019.

[14] M. Massaoudi, H. Abu-Rub, S. Refaat, I. Chihi and F. Oueslati, "Deep Learning in Smart Grid Technology: A Review of Recent Advancements and Future Prospects," *IEEE Access*, vol. 9, pp. 54558-54578, 2021.

[15] R. Mills and M. Salman," POWERING IRAQ: CHALLENGES FACING THE ELECTRICITY SECTOR IN IRAQ" *Al-Bayan Center* for *Planning and Studies*, 2020, [Online]. Available: https://www.bayancenter.org/2020/11/6485/

[16] R. E. Ogu, G. Chukwudebe, and I. A. Ezenugu, "An IoT Based Tamper Prevention System for Electricity Meter," *American Journal of Engineering Research*, vol. 5, no. 10, pp. 347-353, 2016.

[17] N. Pranau, T. Raghuraman, S. Vishnuguhan, and B. Meenakshi, "Load Monitoring and Detection of Tampering in Power Lines Using Internet of

Things (Iot)," *IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE) e-ISSN*, vol. 12, no. 2, pp. 2278-1676, 2017.

[18] F. M. Aljumaah, K. K. Abdalla and S. Alwash, "Design and Implementation of Smart Energy Billing System Based on RFID and Zigbee Technologies," 2022 International Conference for Natural and Applied Sciences (ICNAS), Baghdad, Iraq, 2022, pp. 58-63, doi: 10.1109/ICNAS55512.2022.9944714.

[19] R. Meenal, K. M. Kuruvilla, A. Denny, R. V. Jose, and R. Roy, "Power Monitoring and Theft Detection System using IoT," in *Journal of Physics: Conference Series*, vol. 1362, no. 1, p. 012027, 2019.

[20] R. Aswini and V. Keerthihaa, "IoT Based Smart Energy Theft Detection and Monitoring System for Smart Home," in 2020 *International Conference on System, Computation, Automation and Networking (ICSCAN): IEEE*, pp. 1-6, 2020.

[21] K. Kumaran, N. Ananthi, G. Saranya, S. Priyadharshini, T. Thiviyabala and K. Vaishnavi, "Power Theft Detection and Alert System using IOT", *Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 10, pp. 1135-1139, 2021.

[22] M. M. Shubbar, L. A. Abdul-Rahaim and A. A. Hamad, "Larceny Revelations of Electric Energy with Cloud Computing," 2021 *International Conference on Advance of Sustainable Engineering and its Application (ICASEA),* Wasit, Iraq, 2021, pp. 77-82, doi: 10.1109/ICASEA53739.2021.9733009.

[23] S. Saadhavi, R. Bindu, S. Sadhana, N. Srilalitha, K. Rekha, and H. Phaneendra "IoT Based Electricity Theft Monitoring System." 2022 *Intelligent Data Communication Technologies and Internet of Things*. Springer, Singapore, pp. 477-489, 2022.

[24] V. Saritha, A. Kumar Karra, S. Khader Zelani, and Ch. Prasanth "Cloud-Connected Smart Energy Meter with Remote Monitoring and

Control." *Smart and Intelligent Systems*. Springer, Singapore, pp. 297-305, 2022.

[25] S. Jain and A. M. Karandikar, "A Hybrid Approach to Power Theft Detection," *International Journal of Advanced Engineering, Management and Science,* vol. 2, no. 6, pp. 748-750, June 2016.

[26] S. Chatterjee, V. Archana, K. Suresh, R. Saha, R. Gupta and F. Doshi, "Detection of non-technical losses using advanced metering infrastructure and deep recurrent neural networks," *2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*, Milan, Italy, 2017, pp. 1-6, doi: 10.1109/EEEIC.2017.7977665.

[27] Z. Zheng, Y. Yang, X. Niu, H. -N. Dai and Y. Zhou, "Wide and Deep Convolutional Neural Networks for Electricity-Theft Detection to Secure Smart Grids," in *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1606-1615, April 2018, doi: 10.1109/TII.2017.2785963.

[28] d. Hasan, R. Toma, A. Nahid, M. Islam, and J. Kim, “Electricity Theft Detection in Smart Grid Systems: A CNN-LSTM Based Approach,” *Energies*, vol. 12, no. 17, p. 3310, Aug. 2019.

[29] M. Adil, N. Javaid, U. Qasim, I. Ullah, M. Shafiq, and J. Choi, “LSTM and Bat-Based RUSBoost Approach for Electricity Theft Detection,” *Applied Sciences*, vol. 10, no. 12, p. 4378, Jun. 2020.

[30] S. Mujeeb, N. Javaid, R. Khalid, M. Imran and N. Naseer, "DE-RUSBoost: An Efficient Electricity Theft Detection Scheme with Additive Communication Layer,*" ICC 2020 - 2020 IEEE International Conference on Communications (ICC),* Dublin, Ireland, pp. 1-6, 2020.

[31] Z. Chen, D. Meng, Y. Zhang, T. Xin and D. Xiao, "Electricity Theft Detection Using Deep Bidirectional Recurrent Neural Network," *2020 22nd International Conference on Advanced Communication Technology (ICACT),* Phoenix Park, Korea (South), pp. 401-406, 2020.

[32] D. Syed, H. Abu-Rub, S. Refaat and L. Xie, "Detection of Energy Theft in Smart Grids using Electricity Consumption Patterns," *IEEE International Conference on Big Data (Big Data)*, Atlanta, GA, USA, pp. 4059-4064, 2020.

[33] J. Pereira, F. Saraiva, "Convolutional neural network applied to detect electricity theft: A comparative study on unbalanced data handling techniques", *International Journal of Electrical Power & Energy Systems*, Vol. 131, PP. 107085, 2021.

[34] J. Huang, G. Cheng, Z. Zhang, Q. Li, Y. Li, and W. Jin,"Energy theft detection in an edge data center using deep learning." *Mathematical Problems in Engineering* 2021 (2021): 1-12.

[35] O. FERIAL, "Fraud detection using deep learning Detecting energy consumption fraud using deep learning.", *archives.univ-biskra.dz*, 2021.

[36] L. Duarte Soares, A. de Souza Queiroz, G. López, E. Carreño-Franco, J. López-Lezama, and N. Muñoz-Galeano, "BiGRU-CNN Neural Network Applied to Electric Energy Theft Detection," *Electronics*, vol. 11, no. 5, p. 693, Feb. 2022.

[37] Y. Sun, J. Lee, S. Kim, J. Seon, S. Lee, C. Kyeong, J. Kim, "Energy Theft Detection Model Based on VAE-GAN for Imbalanced Dataset," *Energies*, vol. 16, no. 3, p. 1109, Jan. 2023.

[38] T. S. Gunawan, M. H. Anuar, M. Kartiwi, and Z. Janin, "Development of Power Factor Meter using Arduino," in 2018 *IEEE 5th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA): IEEE*, pp. 1-4, 2018.

[39] A. Taye, "Design and Simulation of Automatic Power Factor Correction for Industry Application," *International Journal of Engineering Technologies and Management Research*, vol. 5, no. 2, pp. 10-21, 2018.

[40] P. Bhagavathy, R. Latha and E. Thamizhmaran, "Development of IoT Enabled Smart APFC Panel for Industrial Loads," 2019 *10th International Conference on Computing, Communication and Networking Technologies*

(ICCCNT), Kanpur, India, 2019, pp. 1-5, doi: 10.1109/ICCCNT45670.2019.8944899.

[41] A. Barhate, K. Mundada, B. Kulkarni, and M. Deore, "Smart Internet of Things Based Automatic Power Factor Control," *Proceedings of International Conference on Communication and Information Processing (ICCIP)*, May, 2019.

[42] T. S. Gunawan, M. H. Anuar, M. Kartiwi, and Z. Janin, "Design of Power Factor Meter Using Internet of Things for Power Factor Improvement, Remote Monitoring and Data Logging," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 17, no. 2, pp. 700-709, February 2020. DOI: 10.11591/ijeecs.v17.i2.

[43] M. M. Shubbar, L. A. Abdul-Rahaim, and A. A. Hamad, "Cloud-Based Automated Power Factor Correction and Power Monitoring," Mathematical Modelling of Engineering Problems, vol. 8, no. 5, pp. 757-762, Oct. 2021. https://doi.org/10.18280/mmep.080510

[44] D. C. Nugroho, Y. Mayaratri, M. Syai'in, M. K. Hasin, N. H. Rohiem, N. P. U. Putra, and A. Soeprijanto, "Household electricity network monitoring based on IoT with automatic power factors improvement using neural network method," *IOP Conference Series: Materials Science and Engineering*, Vol. 1010. No. 1, 2021.

[45] M. Madhiarasan, "Implementation of IoT-based energy monitoring and automatic power factor correction system," *Thermal Science and Engineering*, vol. 6, no. 1, 2023, doi:10.24294/tse.v6i1.1996.

[46] S. McLaughlin, D. Podkuiko, and P. McDaniel, "Energy Theft in the Advanced Metering Infrastructure," in *International Workshop on Critical Information Infrastructures Security*. vol. 6027, Springer, Berlin, Heidelberg, 2010, https://doi.org/10.1007/978-3-642-14379-3_15.

[47] M. A. de Souza, J. L. R. Pereira, G. de O. Alves, B. C. de Oliveira, I. D. Melo, and P. A. N. Garcia, "Detection and identification of energy theft in

advanced metering infrastructures," *Electric Power Systems Research*, vol. 182, 106258, 2020. https://doi.org/10.1016/j.epsr.2020.106258.

[48] L. L. Pfitscher, A. R. Abaide, D. P. Bernardon, and V. J. Garcia, "Introduction to Smart Operation Centers," in D. Bernardon and V. Garcia (eds), *Smart Operation for Power Distribution Systems, Springer*, Cham, pp. 1-14,2018.

[49] M. P. Raju and A. J. Laxmi, "IoT based online load forecasting using machine learning algorithms," *Procedia Computer Science*, vol. 171, pp. 551-560, 2020.

[50] [49] F. Q. Kamal and A. A. Betti, "Towards securing cloud data in the multicloud scenario," *Periodicals of Engineering and Natural Sciences (PEN)*, vol. 9, no. 2, pp. 868-872, 2021.

[51] P. Qian, D. Zhang, X. Tian, Y. Si, and L. Li, "A novel wind turbine condition monitoring method based on cloud computing," *Renewable energy*, vol. 135, pp. 390-398, 2019.

[52] M. Pau et al., "Design and Accuracy Analysis of Multilevel State Estimation Based on Smart Metering Infrastructure," in *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 11, pp. 4300-4312, Nov. 2019, doi: 10.1109/TIM.2018.2890399.

[53] A. A. Mohammed, M. A. N. Al-hayanni, and H. M. Azzawi, "Detection and segmentation the affected brain using ThingSpeak platform based on IoT cloud analysis," *Periodicals of Engineering and Natural Sciences (PEN)*, vol. 9, no. 2, pp. 858-867, 2021.

[54] N. Sultan, "Making use of cloud computing for healthcare provision: Opportunities and challenges," *International Journal of Information Management*, vol. 34, no. 2, pp. 177-184, 2014.

[55] Y. H. Lin, "Novel smart home system architecture facilitated with distributed and embedded flexible edge analytics in demand-side management," *International Transactions on Electrical Energy Systems*, vol. 29, no. 6, p. e12014, 2019.

[56] D. A. B. Fernandes, L. F. B. Soares, J. V. Gomes, M. M. Freire, and P. R. M. Inácio, "Security issues in cloud environments: a survey," *International Journal of Information Security*, vol. 13, no. 2, pp. 113-170, Apr. 2014. https://doi.org/10.1007/s10207-013-0208-7.

[57] T. Diaby and B. B. Rad, "Cloud Computing: A review of the Concepts and Deployment Models," *International Journal of Information Technology and Computer Science* , vol. 6, pp. 50-58, Jun. 2017. DOI: 10.5815/ijitcs.2017.06.07.

[58] A. Lisdorf, "Cloud Computing Basics: A Non-Technical Introduction," *1st ed. Apress, Berkeley*, CA, 2021. DOI: 10.1007/978-1-4842-6921-3..

[59] B. P. Gajendra, V. K. Singh and M. Sujeet, "Achieving cloud security using third party auditor, MD5 and identity-based encryption," 2016 *International Conference on Computing, Communication and Automation (ICCCA),* Greater Noida, India, 2016, pp. 1304-1309, doi: 10.1109/CCAA.2016.7813920.

[60] K. K. Patel, S. M. Patel, and P. G. Scholar, "Internet of Things-IoT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges," *International journal of engineering science and computing*. Comput., vol. 6, no. 5, pp. 1–10, 2016, DOI: 10.4010/2016.1482

[61] M. Burhan, R. A. Rehman, B. Khan, and B. S. Kim, "IoT elements, layered architectures and security issues: A comprehensive survey," *Sensors (Switzerland)*, vol. 18, no. 9, 2018, DOI: 10.3390/s18092796

[62] M. M. Sadeeq, N. M. Abdulkareem, S. R. Zeebaree, D. M. Ahmed, A. S. Sami, and R. R. Zebari, "IoT and Cloud computing issues, challenges and 120 opportunities: A review," *Qubahan Academic Journal*, vol. 1, no. 2, pp. 1- 7, 2021.

[63] J. Dizdarević, F. Carpio, A. Jukan, and X. Masip-Bruin, "A survey of communication protocols for internet of things and related challenges of fog and cloud computing integration," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, pp. 1-29, 2019.

[64] V. K. Madasu and T. Eltaeib, "Web Authentication and Authorization and Role of HTTP, HTTPS Protocol in Networking," *Journal of Multidisciplinary Engineering Science and Technology (JMEST)*, vol. 2, no. 3, pp. 381, Mar. 2015.

[65] S. Gomathi, T. Venkatesan, and D. S. Vidhya, "Design and implementation of fault current limiters in distribution system using internet of things," *Wireless Personal Communications*, vol. 102, no. 4, pp. 2643-2666, 2018.

[66] "What is a cloud server?," IBM. [Online]. Available: https://www.ibm.com/topics/cloud-server [Accessed: July 1, 2023].

[67] D. Sehrawat and N. S. Gill, "Deployment of IoT based smart environment: key issues and challenges," *International Journal of Engineering & Technology*, vol. 7, no. 2, pp. 544-550, 2018.

[68] O. Babatunde, O. Al-Debagy, "A comparative review of internet protocol version 4 (IPv4) and internet protocol version 6 (IPv6)," *arXiv preprint arXiv:*1407.2717, 2014.

[69] S. G. M. Koo and Sze Wan Kwong, "Teaching Computer Communication Networks: Top-down or Bottom-up?," *Proceedings Frontiers in Education 35th Annual Conference, Indianapolis*, IN, USA, 2005, pp. S2H-S2H, doi: 10.1109/FIE.2005.1612250.

[70] Y. Kabalci, "A survey on smart metering and smart grid communication," *Renewable and Sustainable Energy Reviews*, vol. 57, pp. 302-318, 2016.

[71] S. Nimbargi, S. Mhaisne, S. Nangare and M. Sinha, "Review on AMI technology for Smart Meter," 2016 *IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT)*, Pune, India, 2016, pp. 21-27, doi: 10.1109/ICAECCT.2016.7942549.

[72] S. Hallur, "Smart Components for a Smart Energy Mete," *International Journal of Advance Research in Engineering Science and Technology*, vol. 4, no. 3, pp. 544-556, 2017.

[73] https://www.amazon.com/ [Accessed: July 1, 2023].

[74] J. Navani, N. Sharma, and S. Sapra, "Technical and non-technical losses in power system and its economic consequence in Indian economy," *International Journal of Electronics and Computer Science Engineering*, vol. 1, no. 2, pp. 757-761, 2012

[75] A.B. Shaik and S. Srinivasan, "A Brief Survey on Random Forest Ensembles in Classification Model," in International Conference on Innovative Computing and Communications, vol. 56, Springer, Singapore, pp. 299-305, 2019. https://doi.org/10.1007/978-981-13-2354-6_27 .

[76] T.G. Dietterich, "An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization," Machine Learning, vol. 40, pp. 139-157, 2000. https://doi.org/10.1023/A:1007607513941.

[77] X. Gong, B. Tang, R. Zhu, W. Liao, and L. Song, “Data Augmentation for Electricity Theft Detection Using Conditional Variational Auto-Encoder,” *Energies*, vol. 13, no. 17, p. 4291, Aug. 2020.

[78] R. Madhure, R. Raman and S. Singh, "CNN-LSTM based Electricity Theft Detector in Advanced Metering Infrastructure," *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, pp. 1-6, 2020.

[79] M. M. Taye, “Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions,” *Computation*, vol. 11, no. 3, p. 52, Mar. 2023, doi: 10.3390/computation11030052.

[80] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, "Learning from Imbalanced Data Sets", 1st ed. Vol. 10. Cham: Springer, 2018. https://doi.org/10.1007/978-3-319-98074-4

[81] I. AlShourbaji, N. Helian, Y. Sun, et al., "Anovel HEOMGA Approach for Class Imbalance Problem in the Application of Customer Churn

Prediction," *SN Computer Science*, vol. 2, p. 464, 2021. https://doi.org/10.1007/s42979-021-00850-y

[82] N.A.A. Khleel, K. Nehéz, "A novel approach for software defect prediction using CNN and GRU based on SMOTE Tomek method," *Journal of Intelligent Information Systems*, vol. 60, pp. 673–707, 2023. https://doi.org/10.1007/s10844-023-00793-1

[83] "Electricity Theft Detection Dataset," Kaggle, Available: https://www.kaggle.com/datasets/sreen28g10/electricity-theft-detection

[84] Mafaz Mohammed Abed Jafar Shubbar, “Design and Implementation of Cloud Computing System for Data Collection and Processing for Smart Operation Centre” *A Thesis Submitted to the Department of Electrical Engineering / College of Engineering / University of Babylon*, 2022.

APPENDICES

Appendix(A)

APPENDIX A-1: Arduino MEGA Wi-Fi R3 Specification and Datasheet.

APPENDIX A-2: PZEM-004T Specification and Datasheet.

APPENDIX A-1: Arduino MEGA Wi-Fi R3 Specification and Datasheet.

Mega +WiFi R3 Atmega2560+NodeMCU ESP8266 32Mb Memory USB-TTL CH340G Compatible For Arduino Mega is Full integration on one board Mega R3 ATmega2560 and WiFi ESP8266 with memory 32Mb. All of the modules can work together or each separately. And everyone has their own pinout headers. The convenient solution for the development of new projects requiring Uno and WiFi.

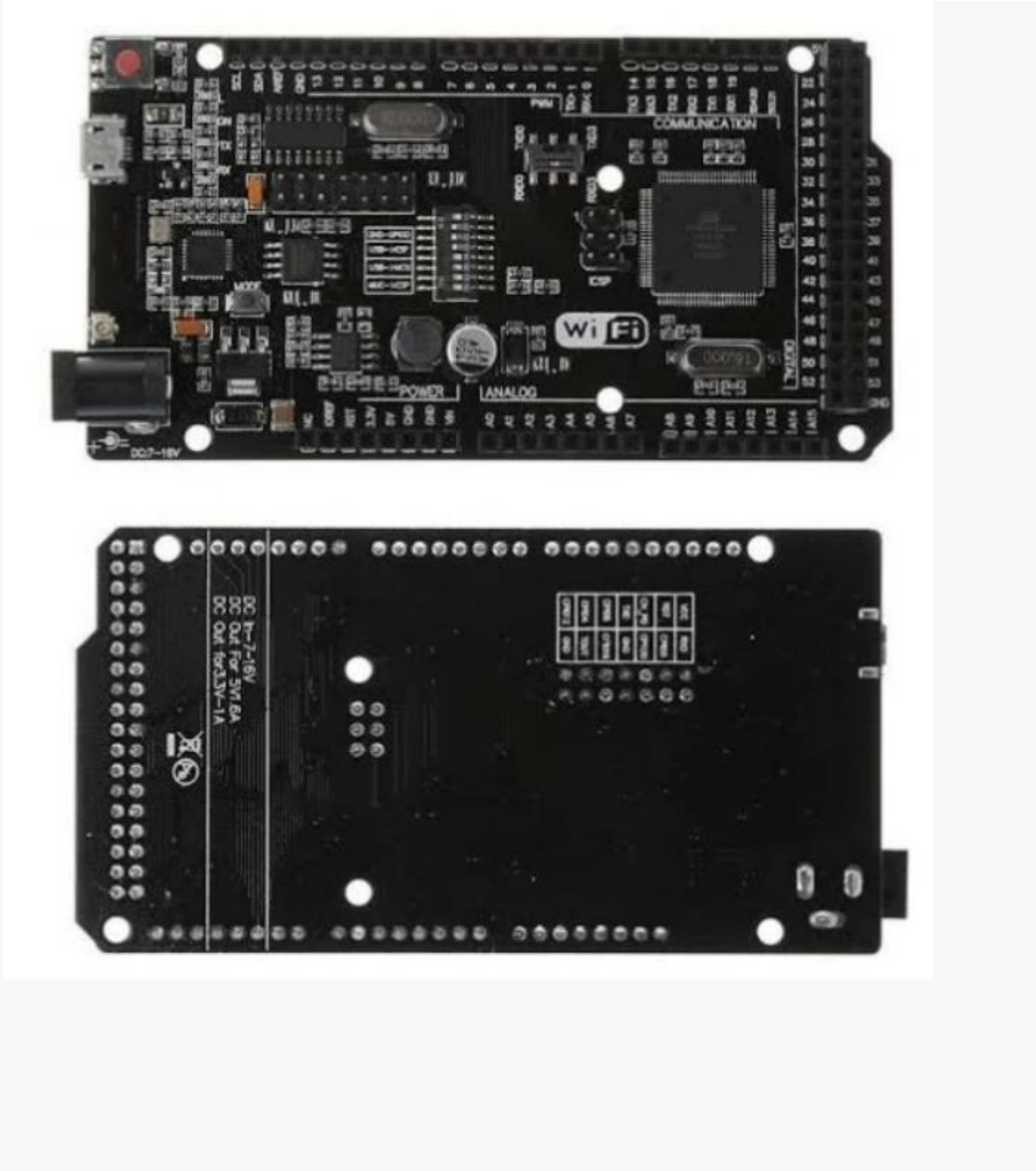
It is a customized version of the classic ARDUINO MEGA R3 board. Full integration of Atmel ATmega2560 microcontroller and ESP8266 Wi-Fi IC, with 32 Mb (megabits) of flash memory, and CH340G USB-TTL converter on a single board! All components can be set up to work together or independently. Use of this board is very simple. The board has DIP-switch, to connect the modules. For example, to USB and ATmeg2560, USB and ESP8266, ATmega2560 and ESP8266.

Specifications:

- Microcontroller:ATmega2560
- IC Wi-Fi : ESP8266
- USB-TTL converter: CH340G
- Power Out: 5V-800mA
- Power Input USB: 5V (500mA max.)
- Power Input VIN/DC Jack: 9-24V
- Power Consumption: 5V 800mA
- Logic Level: 5V
- Wi-Fi 802.11 b/g/n 2.4 GHz
- USB: Micro USB
- Clock Frequency: 16MHz
- Operating Supply Voltage: 5V
- Digital I/O: 54
- Analog I/O: 16
- Memory Size: 256kb
- Data RAM Type/Size: 8Kb
- Data ROM Type/Size: 4Kb
- Interface Type : SerialOTA
- Operating temperature :−40C°/+125C°
- Built-in external antenna

Features:

- Length (mm) : 106
- Width (mm) : 54
- Height (mm) : 12
- Weight (gm) : 40



COMMUNICATION
Wi Fi
POWER
ANALOG
DIGITAL
MODE
DC:7-16V
DC In-7-16V
DC Out For 5V1.6A
DC Out for3.3V-1A

APPENDIX A-2 PZEM-004T Specification and Datasheet

PZEM-004T-10A: Measuring Range 10A (Built-in Shunt)

PZEM-004T-100A: Measuring Range 100A (external transformer)

PZEM-004T V3.0 User Manual

Overview

This document describes the specification of the PZEM-004T AC communication module, the module is mainly used for measuring AC voltage, current, active power, frequency, power factor and active energy, the module is without display function, the data is read through the TTL interface.

PZEM-004T-10A: Measuring Range 10A (Built-in Shunt)

PZEM-004T-100A: Measuring Range 100A (external transformer)

1.Function description

1.1 Voltage

1.1.1 Measuring range:80～260V

1.1.2 Resolution: 0.1V

1.1.3 Measurement accuracy: 0.5%

1.2 Current

1.2.1 Measuring range: 0～10A(PZEM-004T-10A); 0～100A(PZEM-004T-100A)

1.2.2 Starting measure current: 0.01A(PZEM-004T-10A); 0.02A(PZEM-004T-100A)

1.2.3 Resolution: 0.001A

1.2.4 Measurement accuracy: 0.5%

1.3 Active power

1.3.1 Measuring range: 0～2.3kW(PZEM-004T-10A); 0～23kW(PZEM-004T-100A)

1.3.2 Starting measure power: 0.4W

1.3.3 Resolution: 0.1W

1.3.4 Display format:

＜1000W, it display one decimal, such as: 999.9W

≥1000W, it display only integer, such as: 1000W

1.3.5 Measurement accuracy: 0.5%

1.4 Power factor

1.4.1 Measuring range: 0.00～1.00

1.4.2 Resolution: 0.01

1.4.3 Measurement accuracy: 1%

1.5 Frequency

1.5.1 Measuring range: 45Hz～65Hz

1.5.2 Resolution: 0.1Hz

1.5.3 Measurement accuracy: 0.5%

1.6 Active energy

1.6.1 Measuring range: 0～9999.99kWh

1.6.2 Resolution: 1Wh

1.6.3 Measurement accuracy: 0.5%

1.6.4 Display format:

＜10kWh, the display unit is Wh(1kWh=1000Wh), such as: 9999Wh

≥10kWh, the display unit is kWh, such as: 9999.99kWh

1.6.5 Reset energy: use software to reset.

1.7 Over power alarm

Active power threshold can be set, when the measured active power exceeds the threshold, it can alarm

1.8 Communication interface

RS485 interface。

2 Communication protocol

2.1 Physical layer protocol

Physical layer use UART to RS485 communication interface

Baud rate is 9600, 8 data bits, 1 stop bit, no parity

2.2 Application layer protocol

The application layer use the Modbus-RTU protocol to communicate. At present, it only supports function codes such as 0x03 (Read Holding Register), 0x04 (Read Input Register), 0x06 (Write Single Register), 0x41 (Calibration), 0x42 (Reset energy).etc.

0x41 function code is only for internal use (address can be only 0xF8), used for factory calibration and return to factory maintenance occasions, after the function code to increase 16-bit password, the default password is 0x3721

The address range of the slave is 0x01 ~ 0xF7. The address 0x00 is used as the broadcast address, the slave does not need to reply the master. The address 0xF8 is used as the general address, this address can be only used in single-slave environment and can be used for calibration etc.operation.

2.3 Read the measurement result

The command format of the master reads the measurement result is(total of 8 bytes):

Slave Address + 0x04 + Register Address High Byte + Register Address Low Byte + Number of Registers High Byte + Number of Registers Low Byte + CRC Check High Byte + CRC Check Low Byte.

The command format of the reply from the slave is divided into two kinds:

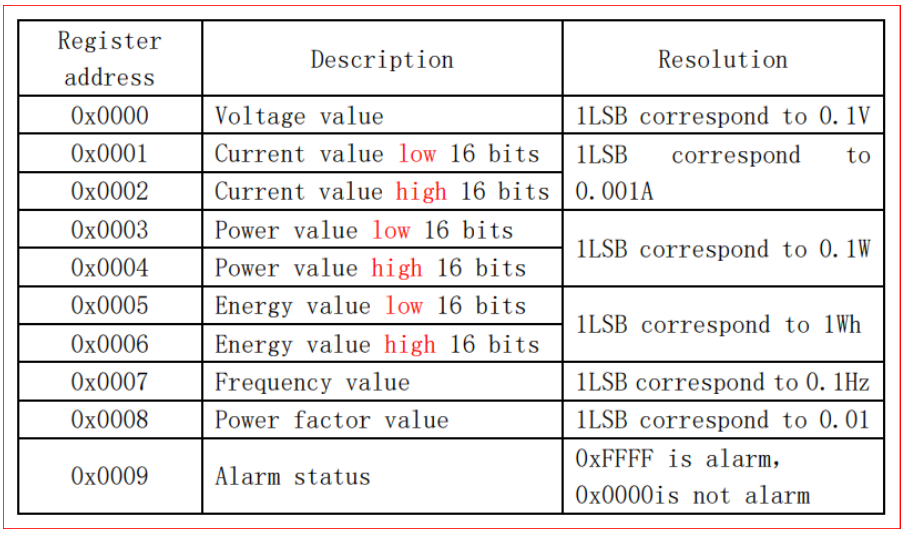
Correct Reply: Slave Address + 0x04 + Number of Bytes + Register 1 Data High Byte + Register 1 Data Low Byte + ... + CRC Check High Byte + CRC Check Low Byte

Error Reply: Slave address + 0x84 + Abnormal code + CRC check high byte + CRC check low byte

Abnormal code analyzed as following (the same below)

□ 0x01,Illegal function
□ 0x02,Illegal address
□ 0x03,Illegal data
□ 0x04,Slave error

The register of the measurement results is arranged as the following table



Register address	Description	Resolution
0x0000	Voltage value	1LSB correspond to 0.1V
0x0001	Current value low 16 bits	1LSB correspond to 0.001A
0x0002	Current value high 16 bits	
0x0003	Power value low 16 bits	1LSB correspond to 0.1W
0x0004	Power value high 16 bits	
0x0005	Energy value low 16 bits	1LSB correspond to 1Wh
0x0006	Energy value high 16 bits	
0x0007	Frequency value	1LSB correspond to 0.1Hz
0x0008	Power factor value	1LSB correspond to 0.01
0x0009	Alarm status	0xFFFF is alarm, 0x0000is not alarm

For example, the master sends the following command (CRC check code is replaced by 0xHH and 0xLL, the same below)

0x01 + 0x04 + 0x00 + 0x00 + 0x00 + 0x0A + 0xHH + 0xLL

Indicates that the master needs to read 10 registers with slave address 0x01 and the start address of the register is 0x0000

The correct reply from the slave is as following:

0x01 + 0x04 + 0x14 + 0x08 + 0x98 + 0x03 + 0xE8+0x00 + 0x00 +0x08 + 0x98+ 0x00 + 0x00 + 0x00 + 0x00 + 0x00 + 0x00 + 0x01 + 0xF4 + 0x00 + 0x64 + 0x00 + 0x00 + 0xHH + 0xLL The above data shows

□ Voltage is 0x0898, converted to decimal is 2200, display 220.0V
□ Current is 0x000003E8, converted to decimal is 1000, display 1.000A
□ Power is 0x00000898, converted to decimal is 2200, display 220.0W
□ Energy is 0x00000000, converted to decimal is 0, display 0Wh
□ Frequency is 0x01F4, converted to decimal is 500, display 50.0Hz
□ Power factor is 0x0064, converted to decimal is 100, display 1.00
□ Alarm status is 0x0000, indicates that the current power is lower than the alarm power threshold

2.4 Read and modify the slave parameters

At present,it only supports reading and modifying slave address and power alarm threshold

The register is arranged as the following table

Register address	Description	Resolution
0x0001	Power alarm threshold	1LSB correspond to 1W
0x0002	Modbus-RTU address	The range is 0x0001~0x00F7

The command format of the master to read the slave parameters and read the measurement results are same(descrybed in details in Section 2.3), only need to change the function code from 0x04 to 0x03.

The command format of the master to modify the slave parameters is (total of 8 bytes):

Slave Address + 0x06 + Register Address High Byte + Register Address Low Byte + Register Value High Byte + Register Value Low Byte + CRC Check High Byte + CRC Check Low Byte.

The command format of the reply from the slave is divided into two kinds:

Correct Response: Slave Address + 0x06 + Number of Bytes + Register Address Low Byte + Register Value High Byte + Register Value Low Byte + CRC Check High Byte + CRC Check Low Byte.

Error Reply: Slave address + 0x86 + Abnormal code + CRC check high byte + CRC check low byte.

For example, the master sets the slave's power alarm threshold:

0x01 + 0x06 + 0x00 + 0x01 + 0x08 + 0xFC + 0xHH + 0xLL

Indicates that the master needs to set the 0x0001 register (power alarm threshold) to 0x08FC (2300W).

Set up correctly, the slave return to the data which is sent from the master. For

example, the master sets the address of the slave

0x01 + 0x06 + 0x00 + 0x02 + 0x00 + 0x05 + 0xHH + 0xLL

Indicates that the master needs to set the 0x0002 register (Modbus-RTU address) to 0x0005

Set up correctly, the slave return to the data which is sent from the master.

2.5 Reset energy

The command format of the master to reset the slave's **energy** is (total 4 bytes): Slave

address + 0x42 + CRC check high byte + CRC check low byte.

Correct reply: slave address + 0x42 + CRC check high byte + CRC check low byte.

Error Reply: Slave address + 0xC2 + Abnormal code + CRC check high byte + CRC check low byte

2.6 Calibration

The command format of the master to calibrate the slave is (total 6 bytes):

0xF8 + 0x41 + 0x37 + 0x21 + CRC check high byte + CRC check low byte.

Correct reply: 0xF8 + 0x41 + 0x37 + 0x21 + CRC check high byte + CRC check low byte. Error

Reply: 0xF8 + 0xC1 + Abnormal code + CRC check high byte + CRC check low byte. It should be

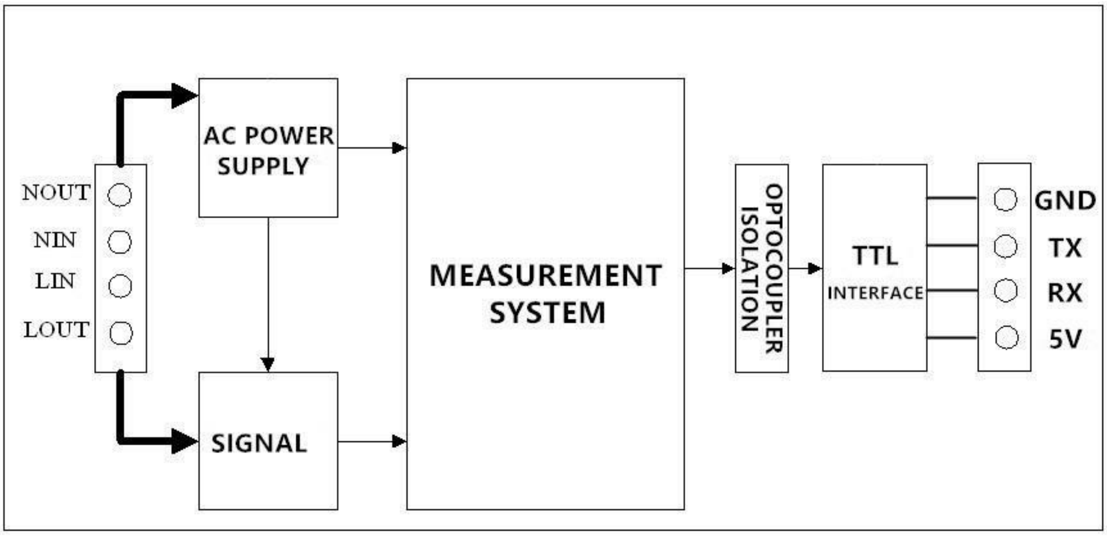
noted that the calibration takes 3 to 4 seconds, after the master sends the command, if the calibration is successful, it will take 3 ~ 4 seconds to receive the response from the slave.

2.7 CRC check

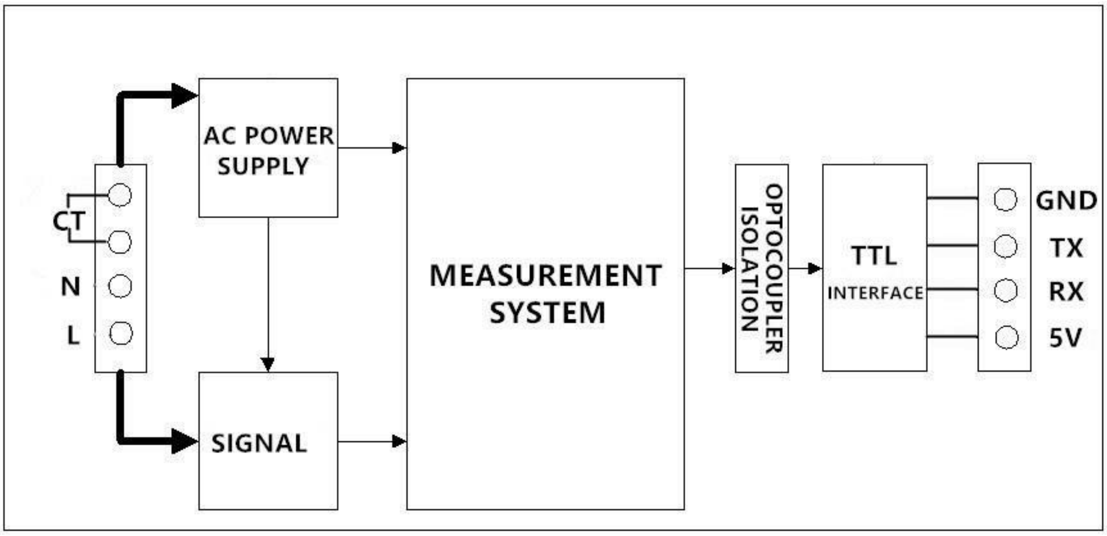
CRC check use 16bits format, occupy two bytes, the generator polynomial is X16 + X15 + X2 +1, the polynomial value used for calculation is 0xA001.

The value of the CRC check is a frame data divide all results of checking all the bytes except the CRC check value.

3 Functional block diagram

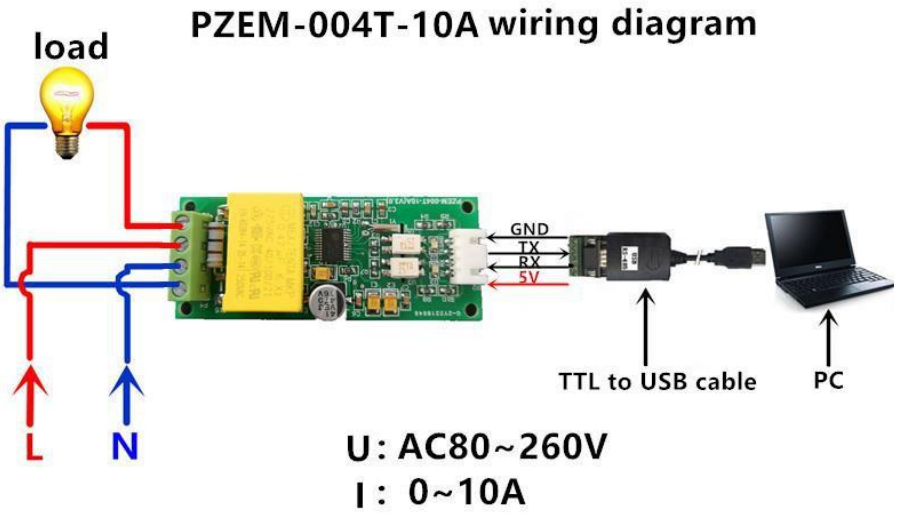


Picture 3.1 PZEM-004T-10A Functional block diagram

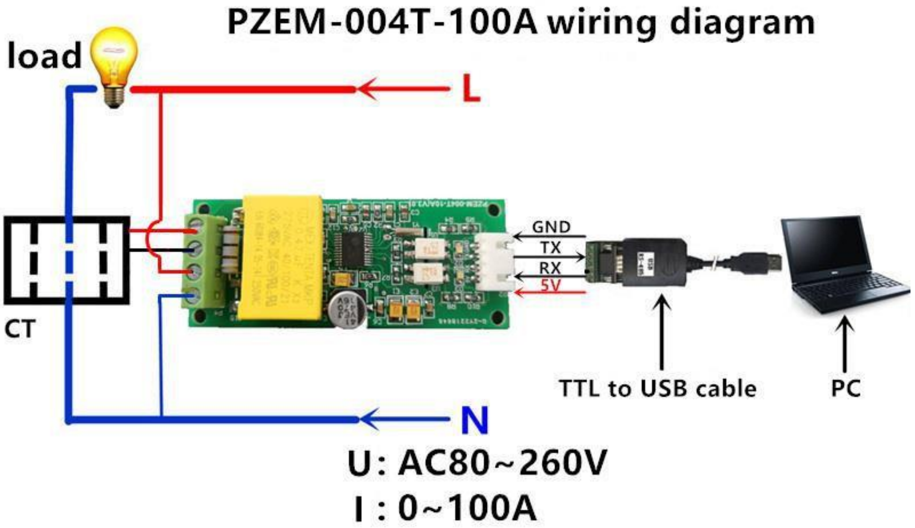


Picture 3.2 PZEM-004T-100A Functional block diagram

4 Wiring diagram



Picture 4.1 PZEM-004T-10A wiring diagram



Picture 4.2 PZEM-004T-100A wiring diagram

5 Other instructions

5.1The TTL interface of this module is a passive interface, it requires external 5V power supply, w hich means, when communicating, all four ports must be connected (5V, RX, TX, GND), otherwis e it cannot communicate.

5.2 Working temperature

-20'C ~ +60'C。

الخلاصة:

يُعتبر الوصول إلى الكهرباء الموثوق بها أمرًا حيويًا لتشغيل العديد من القطاعات، بما في ذلك القطاعات السكنية والصناعية والمرافق الطبية. ومع ذلك، تستمر التحديات مثل سرقة الطاقة وفواتير الكهرباء المفرطة وفقدان الطاقة ، مما يعيق الاستهلاك الفعّال والمستدام للطاقة. تقدم هذه الأطروحة نظام سحابة ذكيًا مصممًا للتغلب على هذه التحديات من خلال اكتشاف سرقة الكهرباء في الوقت الفعلي وتحديد أنماط الحمولة الغير طبيعية مع تصحيح عامل القدرة.

يستخدم النظام المقترح خادمًا سحابيًا لتخزين البيانات والتحليل. يعتمد على نهج هجين يجمع بين الطرق المبنية على الأجهزة و البيانات لاكتشاف سرقة الكهرباء. ينطوي النهج الموجه نحو الأجهزة على تثبيت أجهزة استشعار على الشبكة التوزيعية لتحديد المستهلكين غير القانونيين للكهرباء. في الوقت نفسه، يستخدم النهج الموجه نحو البيانات تقنيات التعلم العميق لتدريب نموذج قادر على اكتشاف أنماط الحمل المشبوهة المرتبطة بالسرقة.

قد خضع النهج الموجه نحو الأجهزة للاختبار في ظروف مختلفة وأظهر فعاليته وكفاءته في اكتشاف حوادث سرقة الكهرباء بدقة. أما النهج الموجه نحو البيانات، باستخدام شبكات التعلم العميقة (Convolutional Neural Networks CNNs)، فقد تم اختياره لقدرته على استخراج سمات ذات صلة من البيانات الكهربائية في هذا النهج، تم استخدام مجموعة بيانات تسمى "مجموعة الشبكة الذكية" المقدمة من شركة الشبكة الكهربائية الحكومية في الصين .(SGCC) لمعالجة توازن البيانات، تم اختبار العديد من تقنيات معالجة البيانات، بما في ذلك تقنيات عينة تحت العشوائية (RUS) وعينة فوق العشوائية (ROS)وتقنية الترفيع الاصطناعي الأقلية (SMOTE) وتقنية SMOTETomek وتقنية العينة الاصطناعية المتكيفة .(ADASYN) من بين هذه التقنيات، تم اختيار تقنية العينة الاصطناعية المتكيفة (ADASYN)للتعامل مع التوازن في البيانات. أظهر تقييم نموذج CNN فعاليته في اكتشاف السرقة بدقة مميزه تبلغ 97.22٪، دقة إيجابية تبلغ 97٪، ومعدل تذكر يبلغ 99.9٪.

من خلال دمج النهجين، يُحسّن النظام بشكل كبير التعرف على سرقة الكهرباء ومنعها، مما يؤدي إلى تحسين استقرار النظام وموثوقيته وكفاءته. بالإضافة إلى ذلك، يشجع النظام على الاستهلاك المسؤول للطاقة، ويضمن الوصول المنصف للكهرباء مع تقليل الأعباء المالية المرتبطة بفواتير الكهرباء المفرطة.

وقد مكّنت تقنيات الحوسبة السحابية تنفيذ ونشر هذه النهجين بشكل كبير. توفر خوادم السحابة السعة التخزينية والموارد الحسابية اللازمة لتنفيذ النماذج وتحليل البيانات وعرض النتائج للسلطات المعنية

بكشف سرقة الكهرباء. سهّلت خوادم السحابة توزيع واستخدام هذا النهج الهجين في سياق اكتشاف سرقة الكهرباء.

بالإضافة إلى ذلك، نقترح استخدام تقنيات التعلم الآلي، وتحديدًا خوارزمية الغابات العشوائية، للتعامل في اختيار قيمة السعة المناسبة لتصحيح عامل القدرة، والذي يشير إلى كفاءة نظام توزيع الكهرباء. من خلال تحسين عامل القدرة، يمكننا تقليل فقدان الطاقة وتخفيض التكلفة الإجمالية لتوفير الكهرباء.

تم تقييم أداء النهج المقترح باستخدام مقاييس مختلفة. حقق النموذج دقة تدريب عالية تبلغ 99.8٪. تدعم دقة الاختبار البالغة 97.93٪ قدرة النموذج على التعميم إلى البيانات الغير معروفة، مما يشير إلى فعاليته في السيناريوهات الحقيقية. علاوة على ذلك، تدل دقة الإيجابية البالغة 97.98٪ على أن غالبية قيم السعة المختارة، في حين تشير نسبة التذكر البالغة 97.93٪ إلى أن نسبة عالية من القيم الحقيقية للسعة تم تحديدها بشكل صحيح من قبل النموذج. بشكل عام، تؤكد هذه النتائج فعالية وموثوقية خوارزمية الغابات العشوائية في تصحيح عامل القدرة وتحسين كفاءة نظام توزيع الكهرباء.

عند مقارنة النتائج التي حققها نظامنا بالنتائج التي حققها الآخرون، من الواضح أن نظامنا يتفوق على الأساليب البديلة.

جمهورية العراق

وزارة التعليم العالي والبحث العلمي

جامعـــــة بابـــل

كلية الهندسة / قسم الهندسة الكهربائية

نظام الحوسبة السحابية الذكية لكشف السرقة لنمط استهلاك الطاقة المنزلي وتصحيح عامل القدرة

رسالة
مقدمة الى قسم الهندسة الكهربائية - كلية الهندسة - جامعة بابل
كجزء من متطلبات نيل درجة الماجستير في الهندسة / الهندسة الكهربائية / الكترونيك

من قبل

سجى عبد الحمزه ياس خضير

اشراف

أ.د. ليث علي عبد الرحيم

و

أ.م.د. سرمد خليل ابراهيم

1445هـ 2023م