**CHAPTER-1**

**INTRODUCTION**

* 1. **Introduction:**

ELECTRICITY THEFT is a problem that affects utility companies worldwide. More than $96 billion is lost by utility companies worldwide due to Non-Technical Losses (NTLs) every year, of which electricity theft is the major contributor [1]. In sub-Saharan Africa, 50% of generated energy is stolen, as reported by World Bank [2].

The ultimate goal of electricity thieves is to consume energy without being billed by utility companies [3], or pay the bills amounting to less than the consumed amount [4]. As a result, utility companies suffer a huge revenue loss due to electricity theft. [5] reports that in 2015, India lost $16.2 billion, Brazil lost $10.5 billion and Russia lost $5.1 billion. It is estimated that approximately $1.31 billion (R20 billion) revenue loss incurred by South Africa (through Eskom) per year is due to electricity theft [2].

Apart from revenue loss, electricity theft has a direct negative impact on the stability and reliability of power grids [3]. It can lead to surging electricity, electrical systems overload and public safety threats such as electric shocks [4]. It also has a direct impact on energy tariff increases, which affect all customers [3]. Implementation of smart grids comes with many opportunities to solve the electricity theft problem [4]. Smart grids are usually composed of traditional power grids, smart meters and sensors, computing facilities to monitor and control grids, etc., all connected through the communication network [6]. Smart meters and sensors collect data such as electricity usage, grid status, electricity price, etc. [6]. Many Utilities sought to curb electricity theft in traditional grids by examining meters' installation and configurations, checking whether the power line is bypassed, etc. [4]. These methods are expensive, inefficient and cannot detect cyber-attacks [4], [7]. Recently, researchers have worked towards detecting electricity theft by utilizing machine learning classification techniques using readily available smart meters data. These theft detection methods have proved to be of relatively lower costs [8]. However, existing classification techniques consider time-domain features and do not regard frequency-domain features, thereby limiting their performance.

Regardless of the fact that there is active ongoing research on electricity theft detection, electricity theft is still a problem. The major cause of delay in solving this problem may be that smart grids deployment is realized in developed nations while developing nations are lagging behind [9]. The challenges of deploying smart grids include the lack of communication infrastructure and users' privacy concerns over data reported by the smart meters [10]. However, [10] reports that smart meters are being considered by many developed and developing countries with aims that include solving NTLs. [11] predicted smart grids global market to triple in size between 2017 and 2023, with the following key regions leading smart grids deployment: North America, Europe and Asia.

In this paper, we present an effective electricity theft detection method based on carefully extracted and selected features in Deep Neural Network (DNN)-based classification approach. We show that employing frequency-domain features as opposed to using time-domain features alone enhances classification performance. We use a realistic electricity consumption dataset released by State Grid Corporation of China (SGCC) accessible at [12]. The dataset consists of electricity consumption data taken from January 2014 to October 2016.

**1.2 THE MAIN CONTRIBUTIONS OF OUR PROJECT ARE AS FOLLOWS:**

1. Based on the literature, we propose a novel DNN classification-based electricity theft detection method using comprehensive time-domain features. We further propose using frequency-domain features to enhance performance.
2. We employ Principal Component Analysis (PCA) to perform classification with reduced feature space and compare the results with classification done with all input features to interpret the results and simplify the future training process.
3. We further use the Minimum Redundancy Maximum Relevance (mRMR) scheme to identify the most significant features and validate the importance of frequency-domain features over time-domain features for detecting electricity theft.
4. We optimize the hyper parameters of the model for overall improved performance using a Bayesian optimizer. We further employ an adaptive moment estimation (Adam) optimizer to determine the best ranges of values of the other key parameters that can be used to achieve good results with optimal model training speed.
5. Lastly, we show 1% improvement in AUC and competitive accuracy of our model in comparison to other data-driven electricity theft detection methods in the literature evaluated on the same dataset.

**The remainder of this paper is organized as follows:**

1. Section II covers the related work done in literature to tackle the electricity theft problem.
2. In Section III, we briefly introduce techniques used in this paper.
3. Section IV covers step by step method taken in this work; which includes dataset analysis and work done to improve its quality and customers' load profile analysis which lead to features extraction and classification.
4. In Section V, we show and discuss the results.
5. We conclude the paper in Section VI. Many Utilities sought to curb electricity theft in traditional grids by examining meters' installation and configurations, checking whether the power line is bypassed, etc. [4].

**CHAPTER-2**

**LITERATURE SURVEY**

**2.1 ‘‘Electricity theft detection using pipeline in machine learning,’’**

**M. Anwar, N. Javaid, A. Khalid, M. Imran, and M. Shoaib**

Electricity theft is the primary cause of electrical power loss that significantly affects the revenue loss and the quality of electrical power. Nevertheless, the existing methods for the detection of this criminal behavior of theft are diversified and complicated since the imbalanced nature of the dataset, and high dimensionality of time-series data make it challenging to extract meaningful information. This paper addresses these problems by developing a novel electricity theft detection model, integrating three algorithms in a pipeline. The proposed method first applies the synthetic minority oversampling technique (SMOTE) for balancing the dataset, secondly integration of kernel function and principal component analysis (KPCA) for the feature extraction from high dimensional time-series data, and support vector machine (SVM) for the classification. Besides, the performance of the proposed pipeline is measured using a comprehensive list of performance metrics. Extensive experiments are performed by using real electricity consumption data, and results show that the proposed method outperforms other methods in terms of theft detection.

**2.2 ‘‘Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids,’’**

**Z. Zheng, Y. Yang, X. Niu, H.-N. Dai, and Y. Zhou**

Electricity theft is harmful to power grids. Integrating information flows with energy flows, smart grids can help to solve the problem of electricity theft owning to the availability of massive data generated from smart grids. The data analysis on the data of smart grids is helpful in detecting electricity theft because of the abnormal electricity consumption pattern of energy thieves. However, the existing methods have poor detection accuracy of electricity theft since most of them were conducted on one-dimensional (1-D) electricity consumption data and failed to capture the periodicity of electricity consumption. In this paper, we originally propose a novel electricity-theft detection method based on wide and deep convolutional neural networks (CNN) model to address the above concerns. In particular, wide and deep CNN model consists of two components: the wide component and the deep CNN component. The deep CNN component can accurately identify the nonperiodicity of electricity theft and the periodicity of normal electricity usage based on 2-D electricity consumption data. Meanwhile, the wide component can capture the global features of 1-D electricity consumption data. As a result, wide and deep CNN model can achieve the excellent performance in electricity-theft detection. Extensive experiments based on realistic dataset show that wide and deep CNN model outperforms other existing methods.

**2.3 ‘‘Smart grid—The new and improved power grid: A survey,’’**

**X. Fang, S. Misra, G. Xue, and D. Yang**

The Smart Grid, regarded as the next generation power grid, uses two-way flows of electricity and information to create a widely distributed automated energy delivery network. In this article, we survey the literature till 2011 on the enabling technologies for the Smart Grid. We explore three major systems, namely the smart infrastructure system, the smart management system, and the smart protection system. We also propose possible future directions in each system. Colored {specifically, for the smart infrastructure system, we explore the smart energy subsystem, the smart information subsystem, and the smart communication subsystem.} For the smart management system, we explore various management objectives, such as improving energy efficiency, profiling demand, maximizing utility, reducing cost, and controlling emission. We also explore various management methods to achieve these objectives. For the smart protection system, we explore various failure protection mechanisms which improve the reliability of the Smart Grid, and explore the security and privacy issues in the Smart Grid.

**2.4 ‘‘Efficient detection of electricity theft cyber attacks in AMI networks,’’**

**M. Ismail, M. Shahin, M. F. Shaaban, E. Serpedin, and K. Qaraqe**

Advanced metering infrastructure (AMI) networks are vulnerable against electricity theft cyber attacks. Different from the existing research that exploits shallow machine learning architectures for electricity theft detection, this paper proposes a deep neural network (DNN)-based customer-specific detector that can efficiently thwart such cyber attacks. The proposed DNN-based detector implements a sequential grid search analysis in its learning stage to appropriately fine tune its hyper-parameters, hence, improving the detection performance. Extensive test studies are carried out based on publicly available real energy consumption data of 5000 customers and the detector's performance is investigated against a mixture of different types of electricity theft cyber attacks. Simulation results demonstrate a significant performance improvement compared with state-of-the-art shallow detectors.

**2.5 ‘‘Tackling energy theft in smart grids through data-driven analysis,’’**

**A. Jindal, A. Schaeffer-Filho, A. K. Marnerides, P. Smith, A. Mauthe, and L. Granville**

The increasing use of information and communication technology (ICT) in electricity grid infrastructures facilitates improved energy generation, transmission, and distribution. However, smart grids are still in their infancy with a disparate regional role out. Due to the involved costs utility providers are only embedding ICT in selected parts of the grid, thereby creating only partial smart grid infrastructures. We argue that using the data provided by these partial smart grid deployments can still be beneficial in solving various issues such as energy theft detection. In this paper, we focus on various data-driven techniques to detect energy theft in power networks. These data-driven detection techniques (at the smart meter as well as the aggregated level) can indicate various forms of energy theft (e.g. through clandestine connections or meter tampering). This paper also presents two case studies to show the effectiveness of these approaches.

**2.6 ‘‘Progress and challenges in smart grids: Distributed generation, smart metering, energy storage and smart loads,’**

**I. Diahovchenko, M. Kolcun, Z. Čonka, V. Savkiv, and R. Mykhailyshyn**

The future power system must provide electricity that is reliable and affordable. To meet this goal, both the electricity grid and the existing control system must become smarter. In this paper, some of the major issues and challenges of smart grid’s development are discussed, and ongoing and future trends are presented with the aim to provide a reader with an insight on the relevant research topics, challenges and actual engineering tasks in smart grids. The focus areas of this review study are distributed generation, microgrids, smart meters’ deployment, energy storage technologies, and the role of smart loads in primary frequency response provision. The exploration of smart grid technologies and distributed generation systems has been accomplished, and a general comparison of the conventional grid and a future smart model is included. The issue of increasing penetration of renewable energy sources to the power system and posers related to the integration of distributed generation are also presented.

**2.7 ‘‘Minimizing household electricity theft in Nigeria using GSM based prepaid meter,’’**

**D. O. Dike, U. A. Obiora, E. C. Nwokorie, and B. C. Dike**

Many households indulge in different forms of electricity theft and illegal tampering of electric metering devices. These lead to distribution system faults and overload as well as loss of revenue by the distribution companies,this paper envisages the utilization of the global system for mobile communication (GSM) into the prepaid energy meter for increased generation of revenue in developing countries like Nigeria. The proposed meter is set to carry a unique identification number such as the consumer’s phone number which may be encrypted into the memory of the microcontroller. Electricity theft is being detected as the GSM module sends message to the distribution company. Revenue generated can be increased through the use of the proposed meter as unaccountability by utility workers and billing irregularities are eliminated. The results obtained from the simulation shows that immediately an illegal load is connected to the utility system either within the residential meter jurisdiction or otherwise stated, the GSM module alerts the utility company no matter how small the illegal load is.

**2.8 ‘‘Power theft detection &intimate energy meter information through SMS with auto power cut off,’’**

**P. Dhokane, M. Sanap, P. Anpat, J. Ghuge, and P. Talole**

Smart grid is a two-way digital communication which allows monitoring, control, and analysis of energy delivery network. In this paper, a smart energy meter is proposed for overload detection, automatic billing, and theft detection. The integration of the microcontroller and GSM provides the meter reading network. The proposed system can send the information like consumed energy in kWh, produced charge, security services (line Cut/On) through GSM module. The information can be incorporated into existing energy management system situated at power organizations or associations. This project can offer all necessary types of assistance distantly for metering and billing accurately. In addition to that, the power theft in house or in industries can be detected by vibration sensor automatically. The energy theft is an issue that keeps tormenting power usage across country. The goal of this paper is to diminish the unlawful utilization of power and moreover decline the odds of theft.

**2.9 ‘‘Prototype development to detect electric theft using PIC18F452 microcontroller,’’**

**S. B. Yousaf, M. Jamil, M. Z. U. Rehman, A. Hassan, and S. O. G. Syed**

This paper presents the development of a prototype to detect electric theft using PIC18F452. The proposed prototype is robust, adaptable, repairable and easy installable. It monitors the flow of charge from the phase line i.e. supply line, the neutral line and constantly compares them. Moreover, it shows real time flow of charge in the both phase line and the neutral line. It also represents the real time voltage and the power being supplied to the load. It is also fitted with an alarm system that sounds an alarm when there is any electric theft. The prototype was able to adapt to different kinds of attenuating voltages between 200-240 volts. It was tested at different loads and findings were inconsistent with the theoretical ones. What makes this device unique is that it can be fitted anywhere in any electrical system. It can be used as metering device. It can also be used as a smart grid surveillance device when used in collaboration with multiple devices of same or different kind.

**CHAPTER-3**

**SYSTEM ANALYSIS**

**3.1 EXISTINGSYSTEM:**

Hardware-based methods [13][19] generally require hardware devices such as specialized microcontrollers, sensors and circuits to be installed on power distribution lines. These methods are generally designed to detect electricity theft done by physically tampering with distribution components such as distribution lines and electricity meters. They can not detect cyber attacks. Electricity cyber attack is a form of electricity theft whereby energy consumption data is modified by hacking the electricity meters [7].

For instance, in [13], an electricity meter was re-designed. It used components that include: a Global System for Mobile Communications (GSM) module, a microcontroller, and an Electrically Erasable Programmable Read-Only Memory (EEPROM). A simulation was done and the meter was able to send a Short Message Service (SMS) whenever an illegal load was connected by bypassing the meter. Limited to detecting electricity theft done by physically tampering with distribution components such as distribution lines and electricity meters. Authors in [16] used the GSM module, ARM-cortex M3 processor and other hardware components to solve the electricity theft problem done in the following four ways: bypassing the phase line, bypassing the meter, disconnecting the neutral line, and tampering with the meter to make unauthorized modifications. A prototype was built to test all four possibilities. The GSM module was able to notify with SMS for each theft case.

Authors in [17] designed ADE7953 chip-based smart meter which is sensitive to current and voltage tempering, and mechanical tempering. ADE7953 was used to detect overvoltage, dropping voltage, overcurrent, the absence of load and other irregularities in voltage and current. It sent an interrupt signal to the Microcontroller Unit (MCU) which reported tampering status. Mechanical tampering was overcome by connecting a tampering switch to MCU's IO ports so that it can send alarm signals to MCU once tampered with. The design was tested with tampering cases such as

connecting the neutral and the phase lines, connecting the meter input and output in reverse mode, and bypassing the phase line to load. The probability of detection failure was 2.13%.

Authors in [15] used a step down transformer, voltage divider circuit, microchip and other hardware components to design a circuitry to detect electricity theft by comparing forward current on the main phase line with reverse current on the neutral line. The circuitry was installed before the meter.The design was tested on Proteus software and on actual hardware. When the meter was bypassed, the problem was detected and an alarm sounded. In [14], a circuit to detect electricity theft done by bypassing the meter was designed. The transformers, rectifiers, microcontroller, GSM module and other hardware components were used. The GSM controller notified the operator with SMS when the meter was bypassed.

**3.2 DISADVANTAGES OF EXISTING SYSTEMS:**

1. An existing system not implemented DNN-BASED ELECTRICITY THEFT DETECTION METHOD.
2. An existing system not implemented Hyperbolic tangent activation function.
3. A lot of Human Intervention in the Existing Systems made them inefficient.
4. Existing systems failed to detect Cyber attacks.

**3.3 PROPOSED SYSTEM**

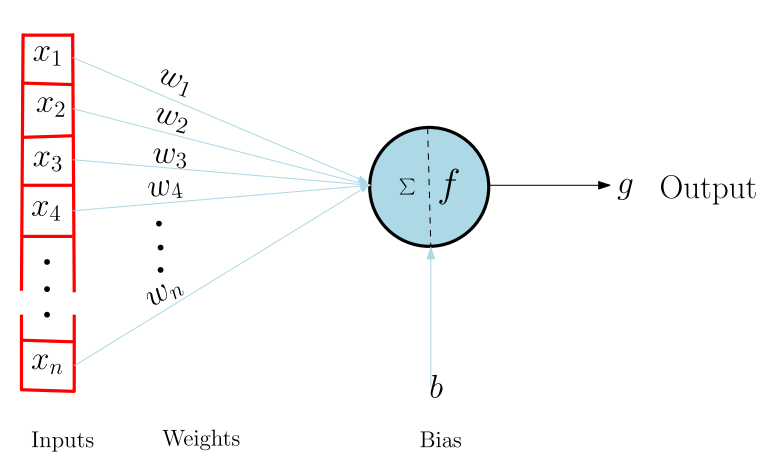
1. Based on the literature, we propose a novel DNN classification-based electricity theft detection method using comprehensive time-domain features. We further propose using frequency-domain features to enhance performance.
2. We employ Principal Component Analysis (PCA) to perform classification with reduced feature space and compare the results with classification done with all input features to interpret the results and simplify the future training process.
3. We further use the Minimum Redundancy Maximum Relevance (mRMR) scheme to identify the most significant features and validate the importance of frequency-domain features over time-domain features for detecting electricity theft.
4. We optimize the hyper parameters of the model for overall improved performance using a Bayesian optimizer. We further employ an adaptive moment estimation(Adam) optimizer to determine the best ranges of values of the other key parameters that can be used to achieve good results with optimal model training speed.
5. Lastly, we show 1% improvement in AUC and competitive accuracy of our model in comparison to other data-driven electricity theft detection methods in theliterature evaluated on the same dataset.

**3.4 Advantages**

1. Huge amount of data obtained by cloud providers and other businesses, making large datasets that train DNNs effectively.
2. Advances in machine learning and signal/information processing research which leads to the evolution of techniques to improve accuracy and broaden the domain of DNNs application.

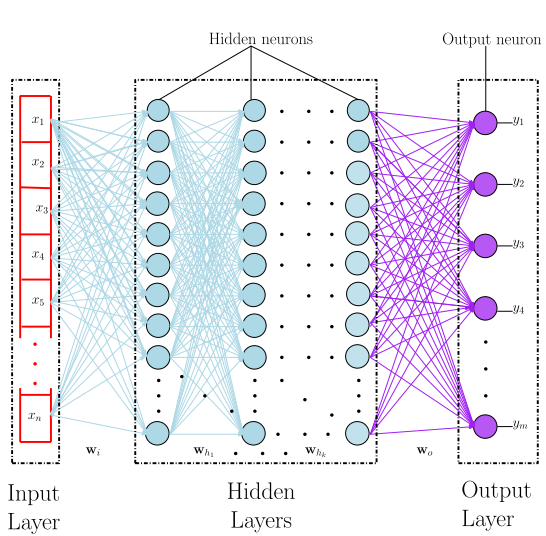
**3.5 ALGORITHMS:**

**3.5.1 DEEP NEURAL NETWORKS:**

Artificial Neural Networks (ANNs) are a class of machine learning techniques that have been built to imitate biological human brain mechanisms .They are typically used for extracting patterns or detecting trends that are difficult to be detected by other machine learning techniques. They consist of multiple layers of nodes/neurons which are connected to subsequent layers. A neuron is the basic element of a neural network, which originates from the McCulloch-Pitts neuron, a simplified model of a human brain’s neuron . Figure 1 shows a model diagram of a neuron that comprises a layer following the input to the ANN. FIGURE 1. First hidden layer neuron model.

**Fig 3.1: First hidden layer Neuron model**

It consists of an activation function f , which takes a weighted sum of the real number input signal and gives real number output y given by Equation (1). g = f ( X(wixi) + b), (1) where xi ∈ x, wi∈ w, x is input vector, w is weights vector and b is the bias . Neural network nodes mimic the brain’s neurons, while connection weights mimic connections between neurons, which are unique for each connection. A neural network stores information in the form of weights and bias. The Deep Neural Networks (DNNs) concept originates from research on ANNs. DNNs are characterized by two or more hidden layers. They are able to learn more complex and abstract features than shallow ANNs. Oftentimes in classification problems, the output layer is made up in such a way that one neuron represents a certain class. All neural network layers are used to filter and learn the complicated features, except for an output layer which classifies based on learnt features.

****

**Fig 2: Fully connected feed-forward DNN general architecture.**

**A typical DNN given just like in Fig 2 has the following major parts:**

* 1. Input layer (x) A layer that comprises input data features or representation.
  2. Input weights (wi) Weights of the connections between the input layer and the first hidden layer of a DNN.
  3. Hidden layers The layers of neurons between the input and output layers. They are used to analyze the relationship between the input and output signals [30].
  4. Hidden neurons weights ([wh1 , · · · , whk ]) Weights of the connections between the hidden layers.
  5. Output weights (wo) Weights between the last hidden layer and the output layer.
  6. Output layer (y) The last layer of a DNN. It gives the output of the network from network inputs.

**3.5.2 USAGE:**

Every Machine learning process involves four Major steps:

1. Data Preprocessing
2. Feature Engineering/ Feature Extraction
3. Classification and Model Training
4. Performance evaluation

**3.5.2.1 DATA PREPROCESSING:**

The dataset consists of daily electricity consumption data of the customers. The sampling rate of the data is uniform for every customer, it is one measurement per day; which corresponds to the total power consumption for that day.Data that we use comes with lot of redundancies such as Missing data ,Errored Data, Unbalanced data due to Meter failures, Data Storage & Transmission problems,Data which has no or both class label.

1. Firstly the records with errors are deleted from the dataset.
2. Data Interpolation algorithms such as **PIECEWISE CUBIC HERMITE INTERPOLATION** has been used for retrieving the Missing record values.
3. **SYNTHETIC DATA GENERATION**  algorithms have been used for addressing the Class Imbalance problem to balance the number of records for both the classes of the dataset(Faithful & Unfaithful).

**3.5.2.2 FEATURE EXTRACTION:**

1. Time domain & Frequency domain features are extracted and are fed to Input layers of DNN.
2. **Principal Component Analysis** is used to get the combined feature space from both the domains.

**3.5.2.3 CLASSIFICATION & MODEL TRAINING:**

**Prerequisites:**

* Number of hidden neurons should be between the size of the input layer and size of the output layer.
* Number of hidden neurons should be approximated to the summation of 2/3 size of input layer and size of the output layer.
* Number of hidden neurons should be less than twice the size of the input layer.

**Activation Function:**

**Rectified Linear unit (ReLU)** activation function was used in the hidden neurons because of its better convergence property in comparison to other activation functions.

**Training:**

The maximum number of training iterations was limited to 1000. The classification approach was split into four parts. In the first part, only time-domain features were used for classification. In the second part, only frequency-domain features were used. The third part comprised of combined features from both domains, while in the last part, classification was performed in reduced feature space by incorporating PCA. Holdout validation scheme was used as follows: in all the procedures, as a rule of thumb, 80% of the whole data was used for training and validation, while 20% of the whole data was used for testing. Within training and validation data, 80% was used for training while 20% was used for validation.

**3.5.2.4 PERFORMANCE EVALUATION:**

Based on true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) obtained from a confusion matrix , we used the following performance metrics to evaluate the classifier’s performance: Recall/True Positive rate (TPR) , Precision/Positive Predictive Value (PPV) , F1-Score [55], Matthews Correlation Co-efficient (MCC) , Accuracy and Area Under the Curve of Receiver Operator Characteristic (AUC-ROC) curve. We briefly introduce performance metrics used as follows.

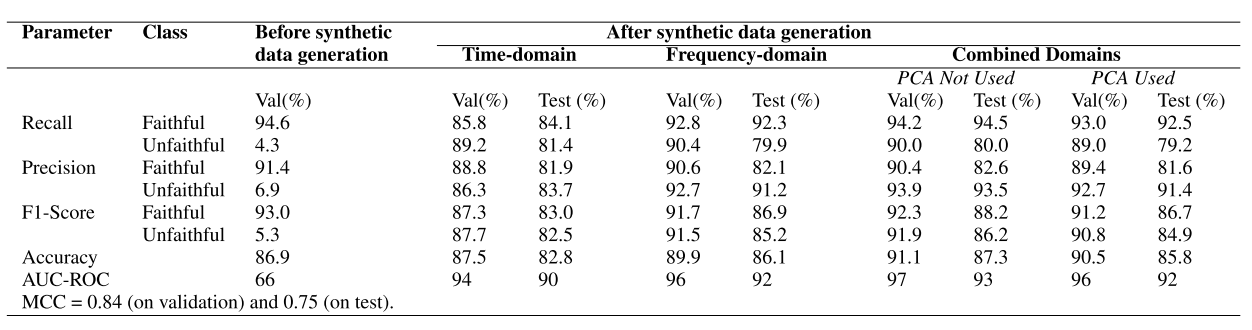
1. **Recall/True Positive Rate (TPR):** is the measure of the fraction of positive examples that are correctly labeled. It is given by: TPR = TP /(TP + FN)
2. **Precision/Positive Predictive Value (PPV):** is the measure of the fraction of examples classified as positive that are truly positive. It is given by: PPV = TP/( TP + FP) .
3. **F1-Score:** shows the balance between precision and recall. It is given by:

F1-Score = 2 ∗ TPR ∗ PPV /(TPR + PPV).

1. **Accuracy**: shows the fraction of predictions classified correctly by the model. It is given by: Accuracy = Number of correct predictions /Total number of predictions =( TP + TN)/( TP + TN + FP + FN )
2. **Matthews Correlation Coefficient (MCC):** a single digit that measures a binary classifier’s performance. Its value ranges from −1 to +1, with values closer to +1 signifying good performance, while values closer to −1 signify bad performance.

MCC = (TP ∗ TN − FP ∗ FN/)(√ (TP + FP)(TP + FN)(TN +FP)(TN + FN)).

1. **Area Under the Curve (AUC):** measures the classifier’s overall quality. Larger AUC values indicate better classifier performance.

****

**3.6 SYSTEM REQUIREMENTS:**

**3.6.1 HARDWARE REQUIREMENTS:-**

* Processor : Pentium–IV
* RAM : 4 GB(min)
* Hard Disk - 20 GB
* KeyBoard - Standard Windows Keyboard
* Mouse - Two or ThreeButton Mouse
* Monitor - SVGA

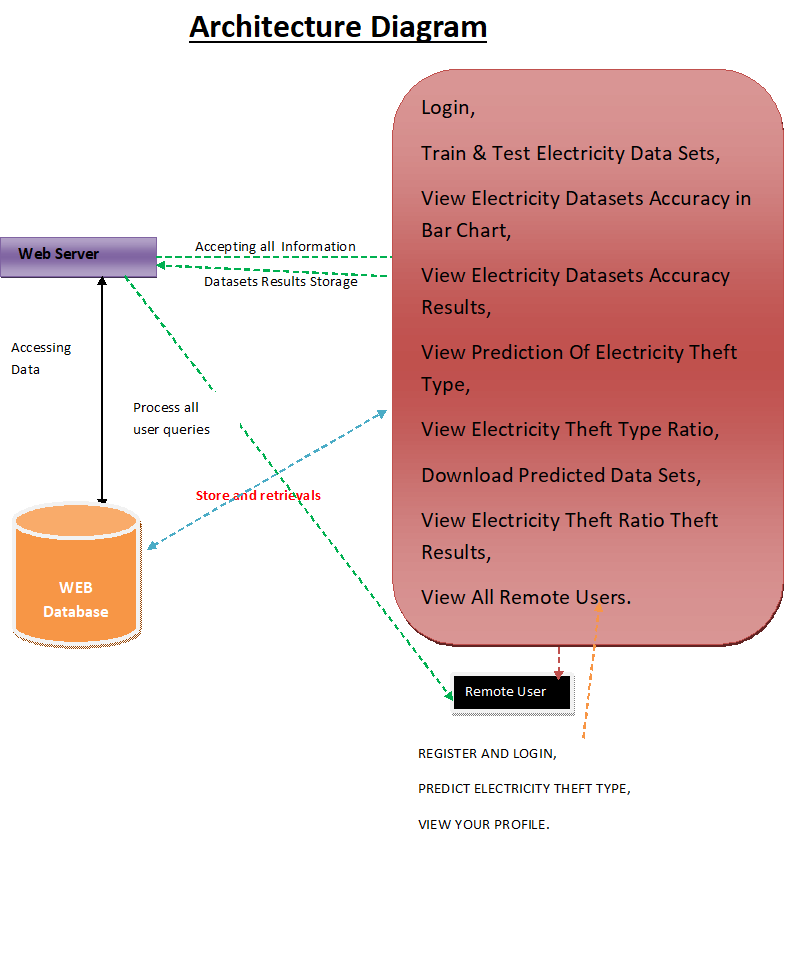
**3.6.2 SOFTWARE REQUIREMENTS:**

* Operating system : Windows 7 Ultimate.
* Coding Language : Python.
* Front-End : Python.
* Back-End : Django-ORM
* Designing : Html, CSS, and JavaScript.
* Data Base : MySQL (WAMP Server).

**CHAPTER – 4**

**SYSTEM DESIGN**

**4.1 SYSTEM ARCHITECTURE:**

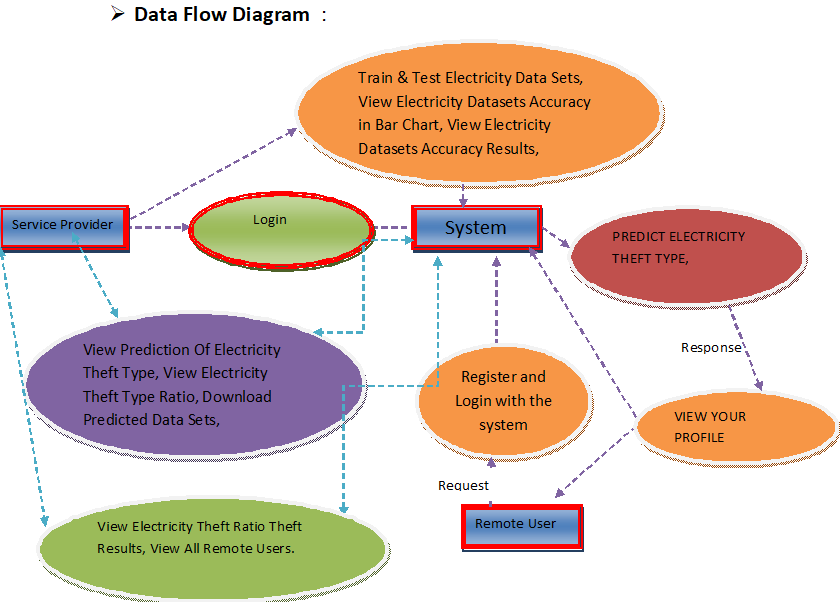
****

**Figure 4.1: Architecture Diagram**

**4.2 DETAILED DESIGN:**

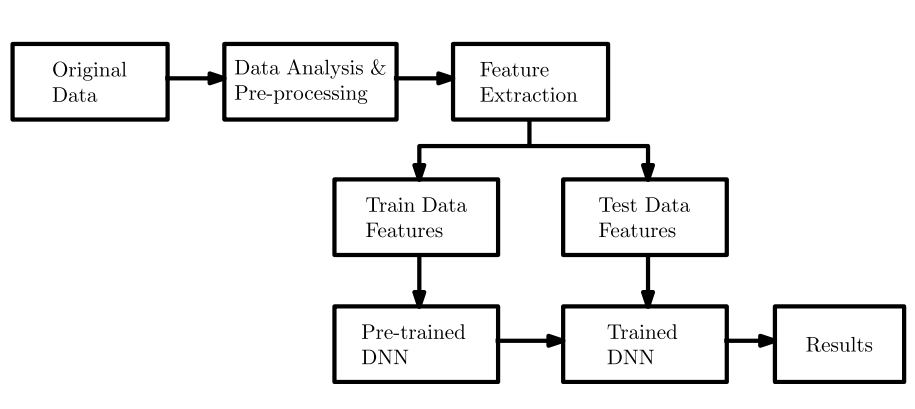
**4.2.1 DATA FLOW DIAGRAM:**

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail



**Figure 4.2.1: Data Flow Diagram**

**4.2.2 WORKFLOW DIAGRAM:**

****

**4.3 UML Diagrams:**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

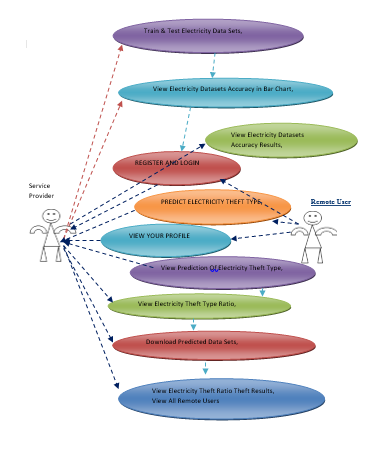
**GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

**4.3.1 USE CASE DIAGRAM:**

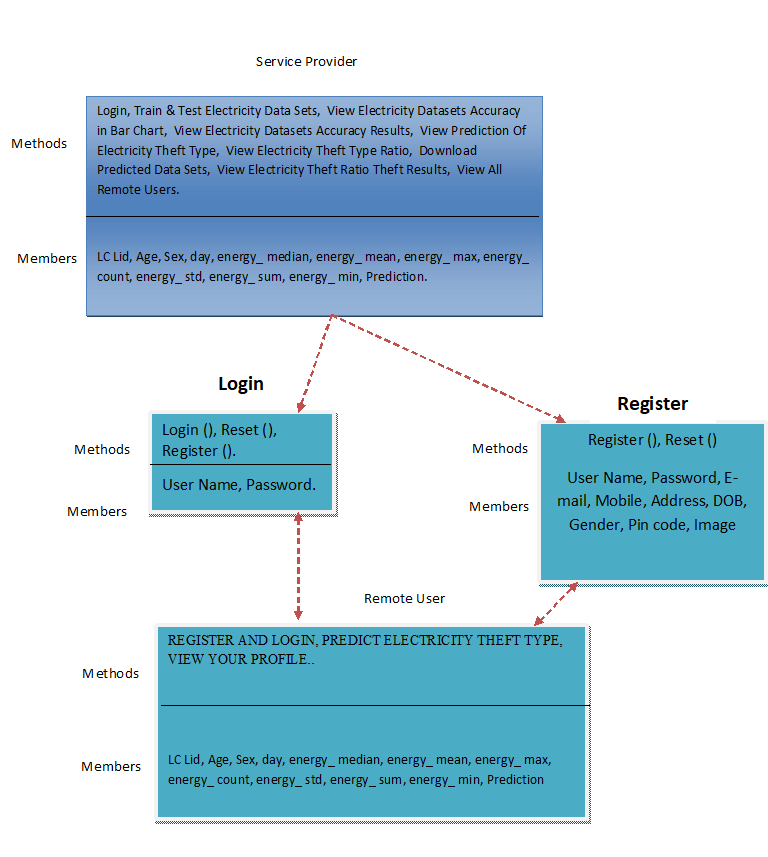
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

****

**Figure 4.3.1: Use Case Diagram**

**4.3.2 CLASS DIAGRAM:**

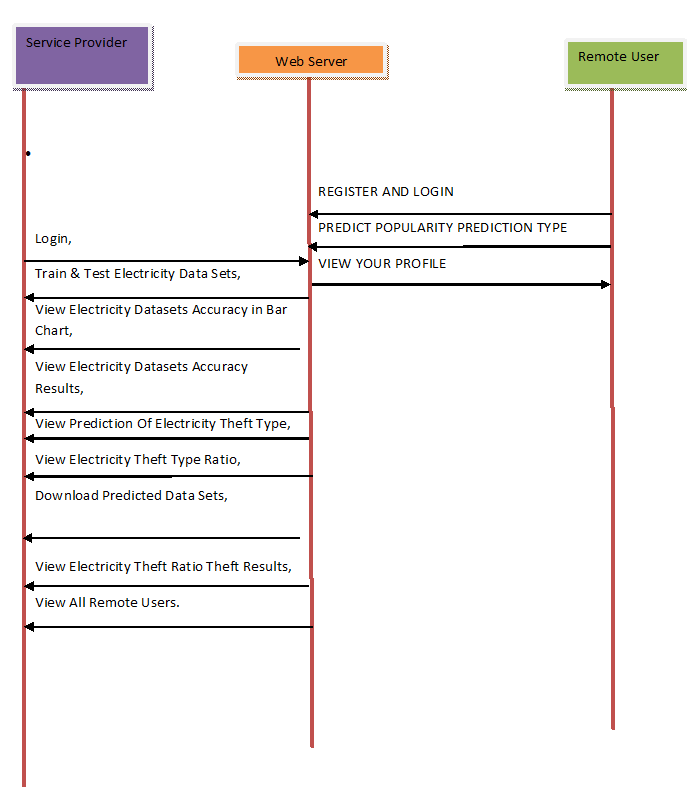
In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



**Figure 4.3.2: Class Diagram**

**4.3.3 SEQUENCE DIAGRAM:**

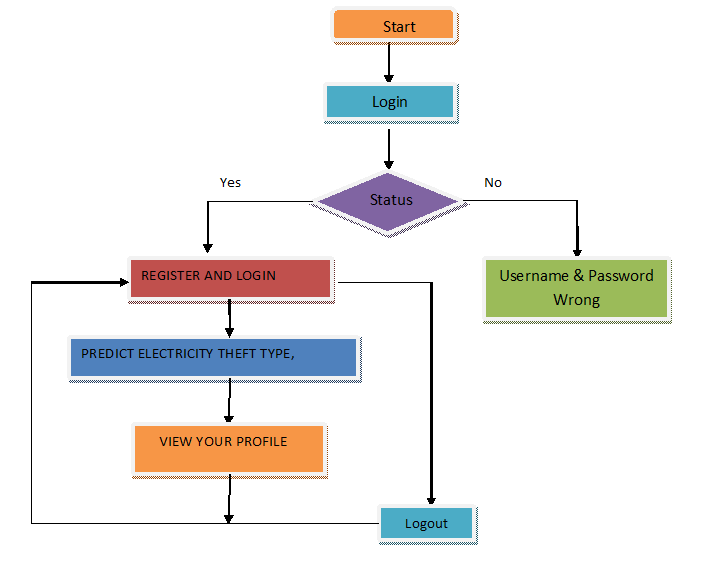
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



**Figure 4.3.3: Sequence Diagram**

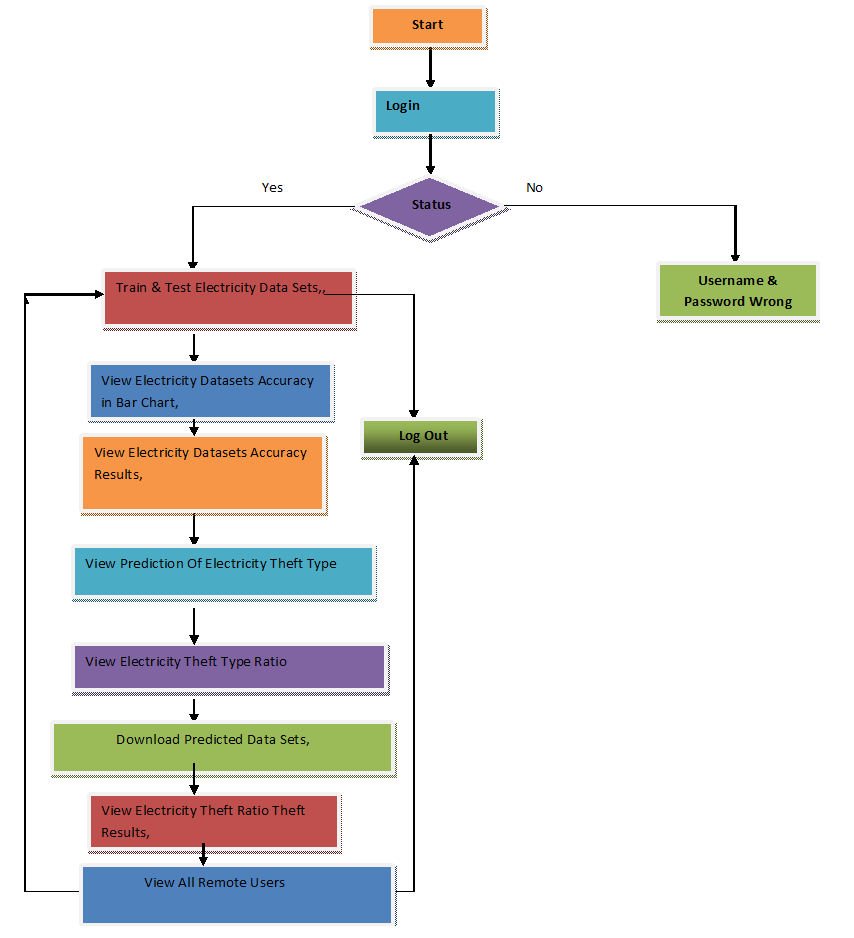
* 1. **FLOW CHAT DIAGRAM:**

**4.4.1 Remote User**

****

**Figure 4.4.1: Flow Chart of Remote User**

* + 1. **Service Provider**

****

**Figure 4.4.2: Flow Chart of Service Provider**

**CHAPTER – 5**

**IMPLEMENTATION**

**MODULES:**

**5.1 SERVICE PROVIDER**

In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as Login, Train & Test Electricity Data Sets, View Electricity Datasets Accuracy in Bar Chart, View Electricity Datasets Accuracy Results, View Prediction Of Electricity Theft Type, View Electricity Theft Type Ratio, Download Predicted Data Sets, View Electricity Theft Ratio Theft Results, View All Remote Users. View All Remote Users.

**5.2 VIEW AND AUTHORIZE USERS**

In this module, the admin can view the list of users who all registered. In this, the admin can view the user’s details such as, user name, email, address and admin authorizes the users.

**5.3 REMOTE USER**

In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like REGISTER AND LOGIN, PREDICT ELECTRICITY THEFT TYPE, VIEW YOUR PROFILE.

**CHAPTER – 6**

**TECHNOLOGY DESCRIPTION**

**6.1 PYTHON:**

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An [interpreted language](https://en.wikipedia.org/wiki/Interpreted_language), Python has a design philosophy that emphasizes code [readability](https://en.wikipedia.org/wiki/Readability) (notably using [whitespace](https://en.wikipedia.org/wiki/Whitespace_character) indentation to delimit [code blocks](https://en.wikipedia.org/wiki/Code_block) rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer [lines of code](https://en.wikipedia.org/wiki/Source_lines_of_code) than might be used in languages such as [C++](https://en.wikipedia.org/wiki/C%2B%2B)or [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many [operating systems](https://en.wikipedia.org/wiki/Operating_system). [C Python](https://en.wikipedia.org/wiki/CPython), the [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) of Python, is [open source](https://en.wikipedia.org/wiki/Open_source) software and has a community-based development model, as do nearly all of its variant implementations. C, Python is managed by the non-profit [Python Software Foundation](https://en.wikipedia.org/wiki/Python_Software_Foundation). Python features a [dynamic type](https://en.wikipedia.org/wiki/Dynamic_type) system and automatic [memory management](https://en.wikipedia.org/wiki/Memory_management). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigm), including [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming), [imperative](https://en.wikipedia.org/wiki/Imperative_programming), [functional](https://en.wikipedia.org/wiki/Functional_programming) and [procedural](https://en.wikipedia.org/wiki/Procedural_programming), and has a large and comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library)

## **What is Python**

## **Python is a popular programming language. It was created by Guido Van Rossum, and released in 1991.**

**It is used for:**

* web development (server-side),
* software development,
* mathematics,
* system scripting.

### **What can Python do**

* Python can be used on a server to create web applications.
* Python can be used alongside software to create workflows.
* Python can connect to database systems. It can also read and modify files.
* Python can be used to handle big data and perform complex mathematics.
* Python can be used for rapid prototyping, or for production-ready software development.

### **Why Python?**

* Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
* Python has a simple syntax similar to the English language.
* Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
* Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
* Python can be treated in a procedural way, an object-orientated way or a functional way.

### **Good to know**

The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.

* In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

### **Python Syntax compared to other programming languages**

* Python was designed for readability, and has some similarities to the English language with influence from mathematics.
* Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
* Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

**6.2 Introduction**

Python applications will often use packages and modules that don’t come as part of the standard library. Applications will sometimes need a specific version of a library, because the application may require that a particular bug has been fixed or the application may be written using an obsolete version of the library’s interface.

This means it may not be possible for one Python installation to meet the requirements of every application. If application A needs version 1.0 of a particular module but application B needs version 2.0, then the requirements are in conflict and installing either version 1.0 or 2.0 will leave one application unable to run.

The solution for this problem is to create a virtual environment, a self-contained directory tree that contains a Python installation for a particular version of Python, plus a number of additional packages.

Different applications can then use different virtual environments. To resolve the earlier example of conflicting requirements, application A can have its own virtual environment with version 1.0 installed while application B has another virtual environment with version 2.0. If application B requires a library be upgraded to version 3.0, this will not affect application A’s environment.

Python is the most widely used multi-purpose, high-level programming language at the moment. Python supports both Object-Oriented and Procedural programming paradigms. Python programmes are typically smaller than those written in other programming languages such as Java. Programmers must type relatively little, and the language's indentation requirement ensures that their code is always readable. Python is used by almost all tech giants, including Google, Amazon, Facebook, Instagram, Dropbox, Uber, and others. Python's greatest strength is its vast collection of standard libraries, which can be used for the following.

• Machine Learning

• GUI Applications (like Kivy, Tkinter, PyQtetc. )

• Web frameworks like Django (used by YouTube, Instagram, Dropbox)

• Image processing (like OpenCV, Pillow)

• Web scraping (like Scrap, Beautiful Soup, and Selenium)

• Test frameworks

• Multimedia

**History of Python:**

What are the similarities between Python and the alphabet? Yes, both begin with ABC. It is abundantly clear that the programming language ABC is being referred to when we talk about ABC in the context of Python. ABC is a broadly useful programming language and programming climate, which had been created in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde and Informatica). Influencing the development of Python was ABC's greatest accomplishment. Python was first thought of at the end of the 1980s. During that time, Guido van Rossum worked on a distributed operating system called Amoeba at the CWI. Guido van Rossum stated in an interview with Bill Venners 1: At Centrum voor Wiskundeen Informatics (CWI), I worked as an implementer on a team developing a language called ABC in the early 1980s. I don't have any idea how well individuals know ABC's effect on Python. I attempt to make reference to ABC's impact since I'm obliged to all that I mastered during that undertaking and to individuals who chipped away at it." Guido van Rossum went on later in the same interview: " I recalled all my experience and a portion of my dissatisfaction with ABC.

I decided to try to create a straightforward scripting language with some of ABC's best features without its drawbacks. So I began composing. I made a straightforward virtual machine, a basic parser, and a basic runtime. I adapted the various ABC parts I liked into my own creation. I developed a basic syntax, substituted indentation for curly braces or begin-end blocks for statement grouping, and a small number of powerful data types: a hash table (or word reference, as we call it), a rundown, strings, and numbers."

**How to Install Python on Windows and Mac:**

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your System Requirements. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a Windows 64-bit operating system. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheat sheet here.The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

**Download the Correct version into the system:**

**Step 1:** Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: <https://www.python.org>

**Fig:6.1.1 Official website of Python**

Now, check for the latest and the correct version for your operating system.

**Step 2:** Click on the Download Tab.

****

**Fig 6.1.2:Download latest version**

**Step 3:** You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

****

**Fig 6.1.3: Click download on specific version**

**Step 4:** Scroll down the page until you find the Files option.

**Step 5:** Here you see a different version of python along with the operating system.



**Fig 6.1.4: Different versions of Python**

• To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

• To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

**Note:** To know the changes or updates that are made in the version you can click on the Release Note Option.

**Installation of Python**

**Step 1:** Go to Download and Open the downloaded python version to carry out the installationprocess.

**Fig 6.1.5: Installation**

**Step 2:** Before you click on Install Now, make sure to put a tick on Add Python 3.7 to PATH.



**Fig 6.1.6: Click install now**

**Step 3:** Click on Install NOW After the installation is successful. Click on Close.



**Fig 6.1.6: Close after setup**

With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

**Verify the Python Installation:**

**Step 1:** Click on Start

**Step 2:** In the Windows Run Command, type “cmd”



**Fig 6.1.8: Command prompt**

**Step 3:** Open the Command prompt option.

**Step 4:** Let us test whether the python is correctly installed. Type python –V and press Enter.



**Fig 6.1.9: Check Python version**

**Step 5:** You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

**Check how the Python IDLE works**

**Step 1**: Click on Start

**Step 2:** In the Windows Run command, type “python idle”



**Fig 6.1.10: Python IDLE**

**Step 3:** Click on IDLE (Python 3.7 64-bit) and launch the program

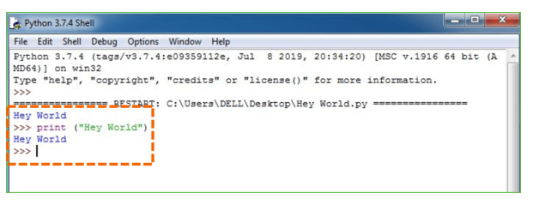
**Step 4:** To go ahead with working in IDLE you must first save the file. Click on File > Click on Save



**Fig 6.1.11 Save the file**

**Step 5**: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

**Step 6:** Now for e.g. enter print (“Hey World”) and Press Enter.



**Fig 6.1.12:Example for writing and executing**

You will see that the command given is launched. With this, we end our tutorial on how to install Python. You have learned how to download python for windows into your respective operating system.

Note: Unlike Java, Python doesn’t need semicolons at the end of the statements otherwise it won’t work.

**6.3 MACHINE LEARNING:**

Before we investigate the subtleties of different AI techniques, how about we start by seeing what AI is, and what it isn't. Although machine learning is frequently referred to as a subfield of artificial intelligence, I believe that categorization can frequently be initially misleading. The investigation of AI positively emerged from research in this specific circumstance, however in the information science use of AI techniques, it's more useful to consider AI for the purpose of building models of information.

In its most basic form, machine learning is the creation of mathematical models that aid in data comprehension. Learning" enters the conflict when we give these models tunable boundaries that can be adjusted to noticed information; The program can thus be regarded as "learning" from the data. These models can be used to predict and comprehend aspects of newly observed data once they have been fitted to data that has already been observed. The more philosophical discussion regarding the degree to which this kind of mathematical, model-based "learning" is comparable to the "learning" exhibited by the human brain will be left up to the reader. Understanding the issue setting in AI is crucial for utilizing these devices successfully, thus we will begin for certain general classifications of the kinds of approaches we'll examine here.

**Classes of Machine Learning:**

Machine learning can be broken down into two main categories at the most fundamental level: learning under supervision and unsupervised learning.

Modeling the relationship between the data's measured features and some label is one aspect of supervised learning. New, unidentified data can be labeled with the help of this model once it has been established. Regression and classification tasks are two more subcategories of this: The labels in classification are discrete categories, whereas the labels in regression are continuous quantities. In the following section, we will see examples of both kinds of supervised learning.

Unsupervised learning is often referred to as "letting the dataset speak for itself," and it involves modeling the features of a dataset without using any labels. These models incorporate undertakings like bunching and dimensionality decrease. Bunching calculations distinguish unmistakable gatherings of information, while dimensionality decrease calculations look for additional brief portrayals of the information. In the following section, we will see examples of both kinds of unsupervised learning.

**Need for AI**

Individuals, as of now, are the most keen and high level species on earth since they can think, assess and tackle complex issues. On the opposite side, computer based intelligence is still in its underlying stage and haven't outperformed human knowledge in numerous viewpoints. Then the inquiry that is the need to make machine learn? The most appropriate justification behind doing this is, "to decide, in view of information, with proficiency and scale".

Recently, associations are putting vigorously in fresher advancements like Man-made brainpower, AI and Profound Figuring out how to get the critical data from information to play out a few true errands and tackle issues. We can call it information driven choices taken by machines, especially to mechanize the cycle. Problems that cannot naturally be programmed can benefit from these data-driven decisions rather than programming logic. The truth of the matter is that we can't manage without human insight, however other perspective is that we as a whole need to take care of genuine issues with proficiency at a gigantic scope. Because of this, machine learning is required.

**Applications of Machines Learning:**

Researchers assert that we are in the golden age of AI and machine learning, with machine learning being the technology with the fastest growth rate. It is used to solve a lot of real-world complex problems that can't be solved using the traditional method. Emotion analysis, sentiment analysis, error detection and prevention, weather forecasting, stock market analysis and forecasting, speech synthesis, speech recognition, customer segmentation, object recognition, fraud detection and prevention, and product recommendation for online shoppers are just a few examples of ML's real-world applications.

1. **Terminologies of Machine Learning**

A model is a particular representation that is learned from data by using a machine learning algorithm. A model is likewise called a speculation.

• **Highlight** - A component is an individual quantifiable property of the information. A feature vector makes it simple to describe a collection of numerical features. The model receives input in the form of feature vectors. For instance, there may be characteristics like color, smell, taste, and so on that can be used to predict a fruit.

• **Target (Label):** The value that will be predicted by our model is referred to as a target variable or label. For the organic product model talked about in the component segment, the mark with each arrangement of information would be the name of the organic product like apple, orange, banana, and so on.

**• Training:** The idea is to provide a set of inputs (features) and expected outputs (labels), so that after training, a model (hypothesis) will map new data to one of the categories trained on.

**• Prediction:** When our model is ready, it can be given a set of inputs and a predicted output (a label) will be generated.

**(b) Types of Machine Learning**

• **Supervised Learning:** This involves using classification and regression models to learn from a training dataset with labeled data. Until the required level of performance is reached, this learning process continues.

• **Unsupervised Learning:** Using factor and cluster analysis models, this method uses unlabeled data to learn more about the data itself by locating its underlying structure.

• **Semi-supervised Learning:** This method uses a small amount of labeled data and unlabeled data, similar to unsupervised learning. Utilizing named information tremendously expands the learning precision and is likewise more savvy than Managed Learning.

**• Reinforcement Learning:** This method entails figuring out the best course of action through trial and error. Learning behaviors that are based on the current state and will maximize reward in the future determine the subsequent action.

**6.4 PACKAGES AND VERSIONS**:

Asgiref 3.7.2

Contourpy 1.1.1

cycler 0.12.1

Django 3.0.4

et-xmlfile 1.1.0

fonttools 4.50.0

importlib-resources 6.3.0

joblib 1.3.2

kiwisolver 1.4.5

matplotlib 3.7.5

mysql-connector-python 8.0.24

mysqlclient 2.0.3

numpy 1.21.3

openpyxl 3.1.2

packaging 24.0

pandas 2.0.3

pillow 10.2.0

pip 20.1.1

protobuf 5.26.0

pyparsing 3.1.2

python-dateutil 2.9.0.post0

pytz 2024.1

scikit-learn 0.22.2.post1

scipy 1.10.1

seaborn 0.10.0

setuptools 47.1.0

six 1.16.0

sqlparse 0.4.4

typing-extensions 4.10.0

tzdata 2024.1

xlwt 1.3.0

**CHAPTER – 7**

**SYSTEM STUDY AND TESTING**

**7.1 FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

**Three key considerations involved in the feasibility analysis are,**

* Economical Feasibility
* Technical Feasibility
* Social Feasibility

**7.1.1 ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### **7.1.2 TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**7.1.3 SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**7.2 SYSTEM TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### **7.2.1 TYPES OF TESTS**

**7.2.1.1 UNIT TESTING**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**7.2.1.2 INTEGRATION TESTING**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**7.2.1.3 FUNCTIONAL TEST**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input: identified classes of valid input must be accepted.

Invalid Input: identified classes of invalid input must be rejected.

Functions: identified functions must be exercised.

Output: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**7.2.1.4 SYSTEM TEST**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**7.2.1.5 WHITE BOX TESTING**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**7.2.1.6 BLACK BOX TESTING**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**7.2.1.7 UNIT TESTING**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**TEST STRATEGY AND APPROACH:**

Field testing will be performed manually and functional tests will be written in detail.

**TEST OBJECTIVES:**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**FEATURES TO BE TESTED:**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

**7.2.1.8 ACCEPTANCE TESTING**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

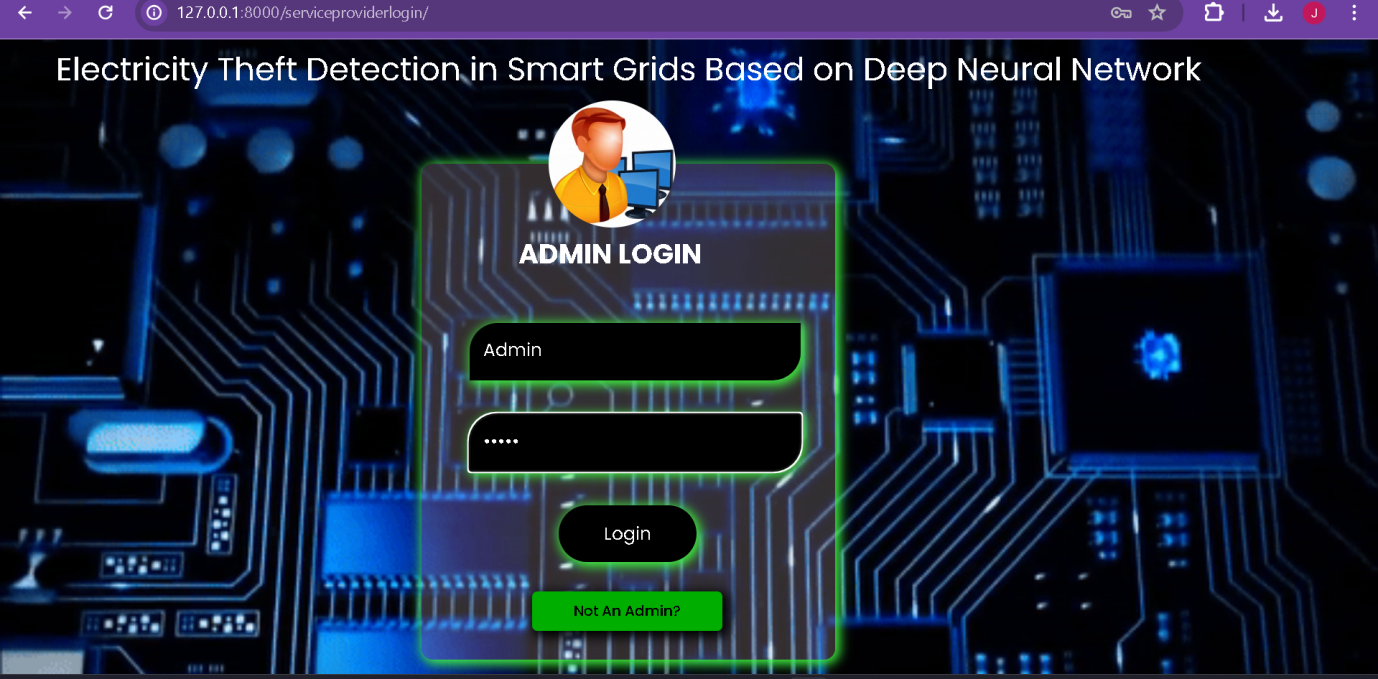
**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**CHAPTER – 8**

**OUTPUT SCREENS**

**8.1 For service provider:**

1. Below is the screen for Login Page of Service Provider. Service Provider has to login by using valid user name and password



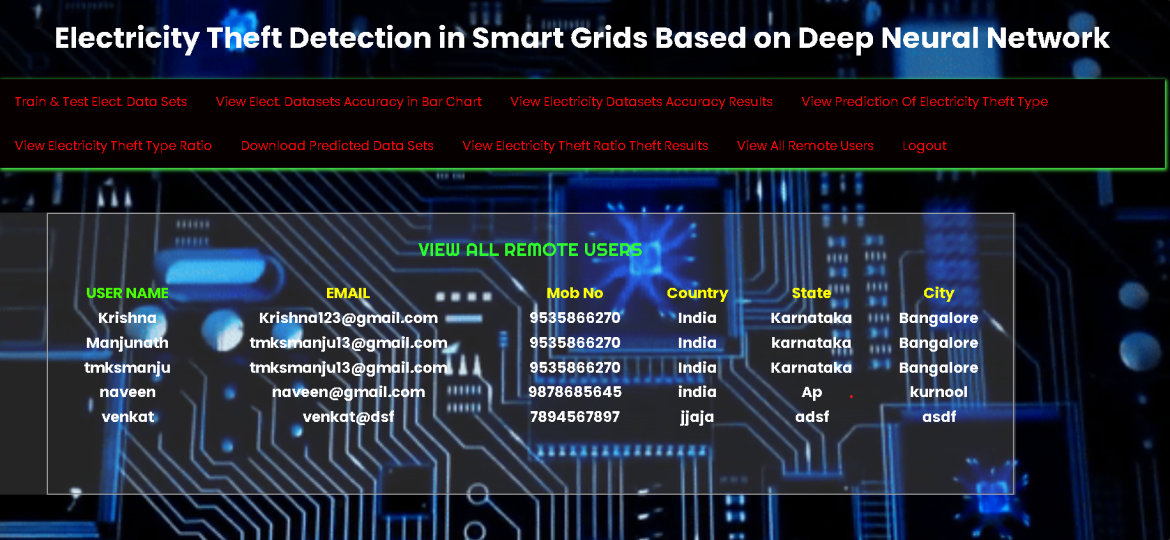
**Fig8.2.1: Login Pages Overview**

2. After login below screen is displayed.

In below screen after login the admin have the Access to datasets, Model accuracy visualization, Electricity theft prediction, User management.

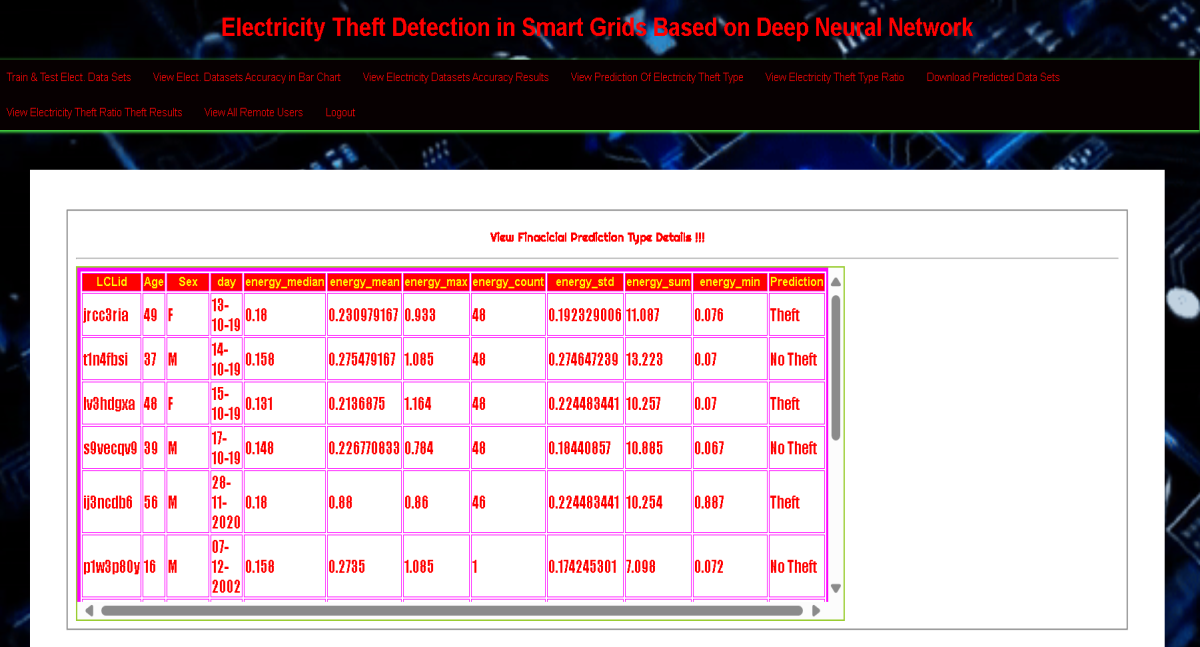
**Operations:**

* Train & Test Disaster Data Sets,
* View Electricity datasets Accuracy (Bar Chart),
* View Prediction of Electricity Theft Type,
* Download Predicted Data Sets,
* View Electricity theft Type Ratio,
* View All Remote Users.



**Fig 8.2.3: Main Menu**

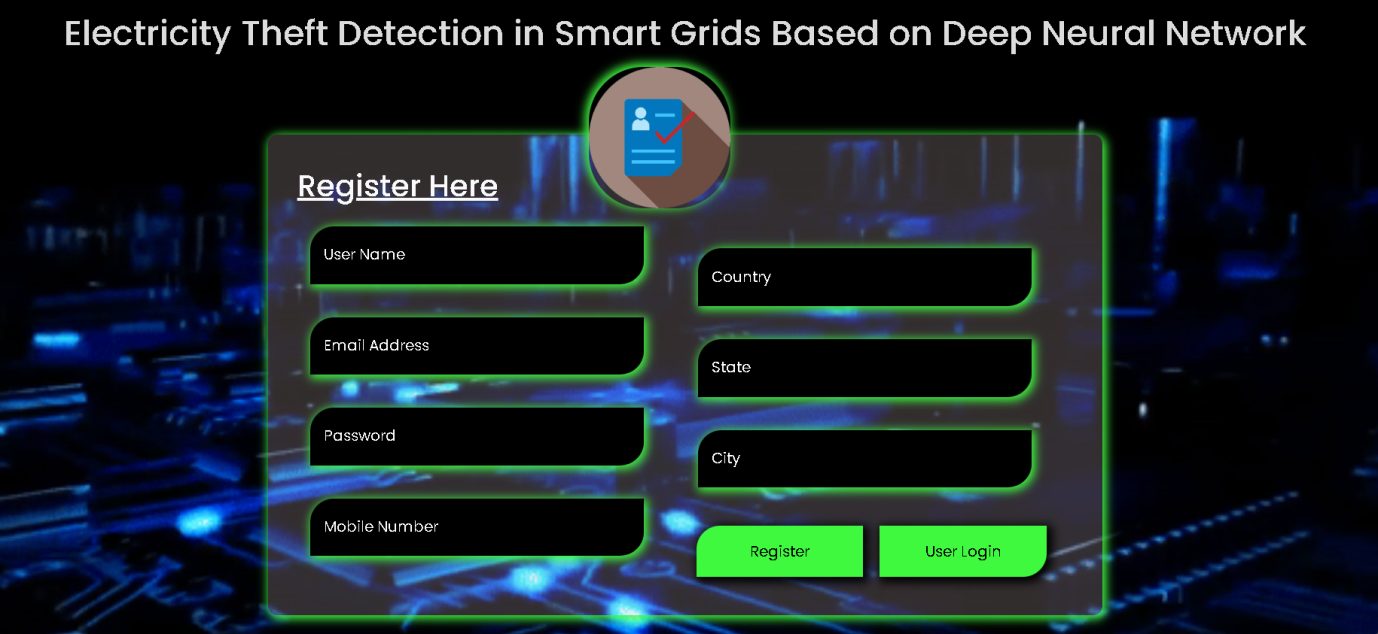
3. The below screen shows the Electricity Theft type Ratio



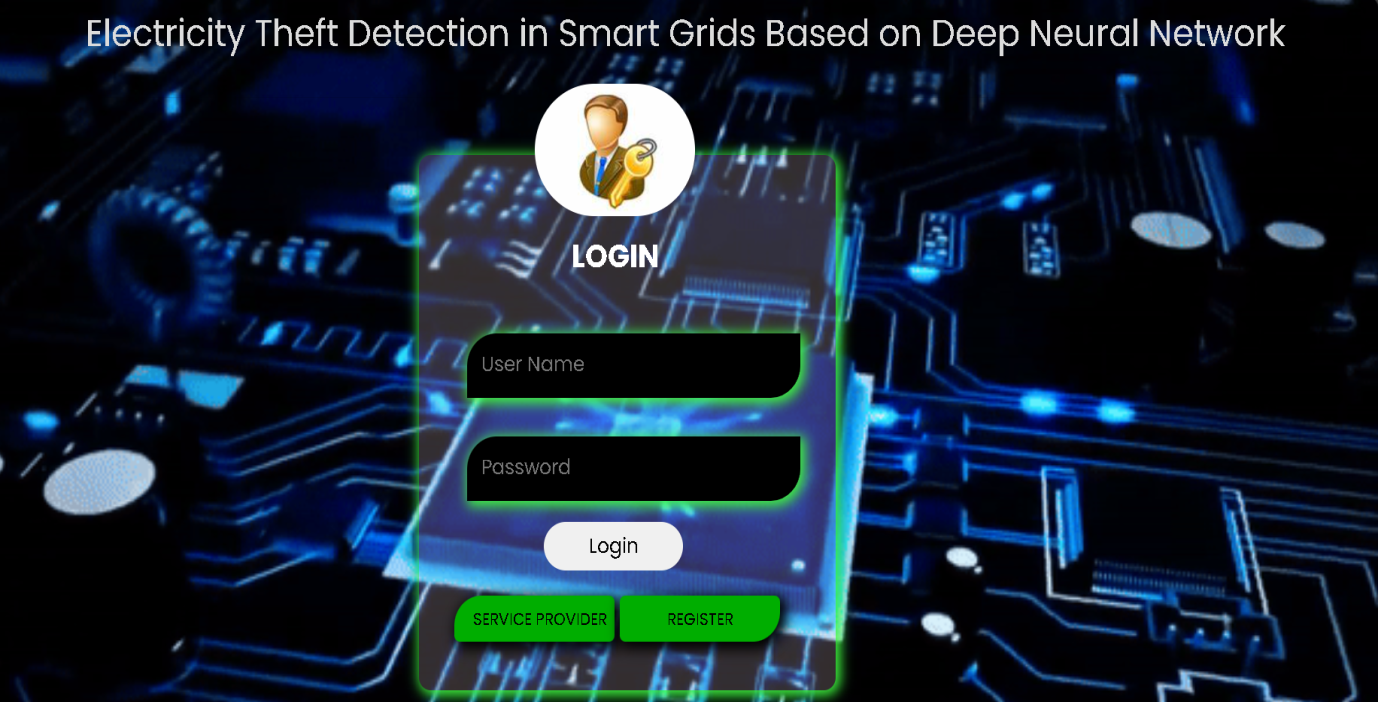
* 1. **For Remote Users:**

1. Below is the screen for **User Registration** which asks to fill the information such as:

* User name
* Email Address
* Password
* Contact Number
* Country
* State
* City

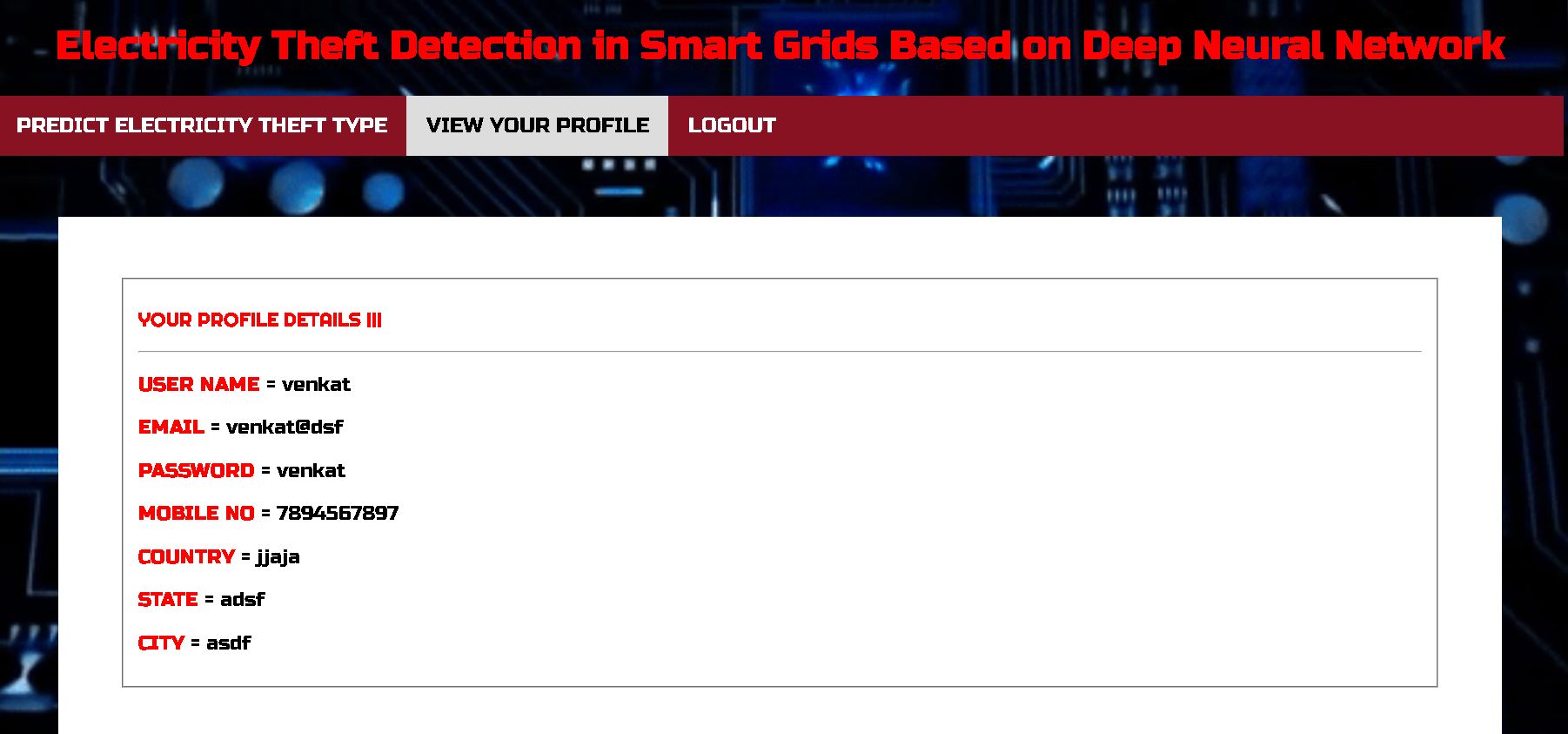


2. Below is the screen for User Login.After entering correct credentials the user get logged in into the system.



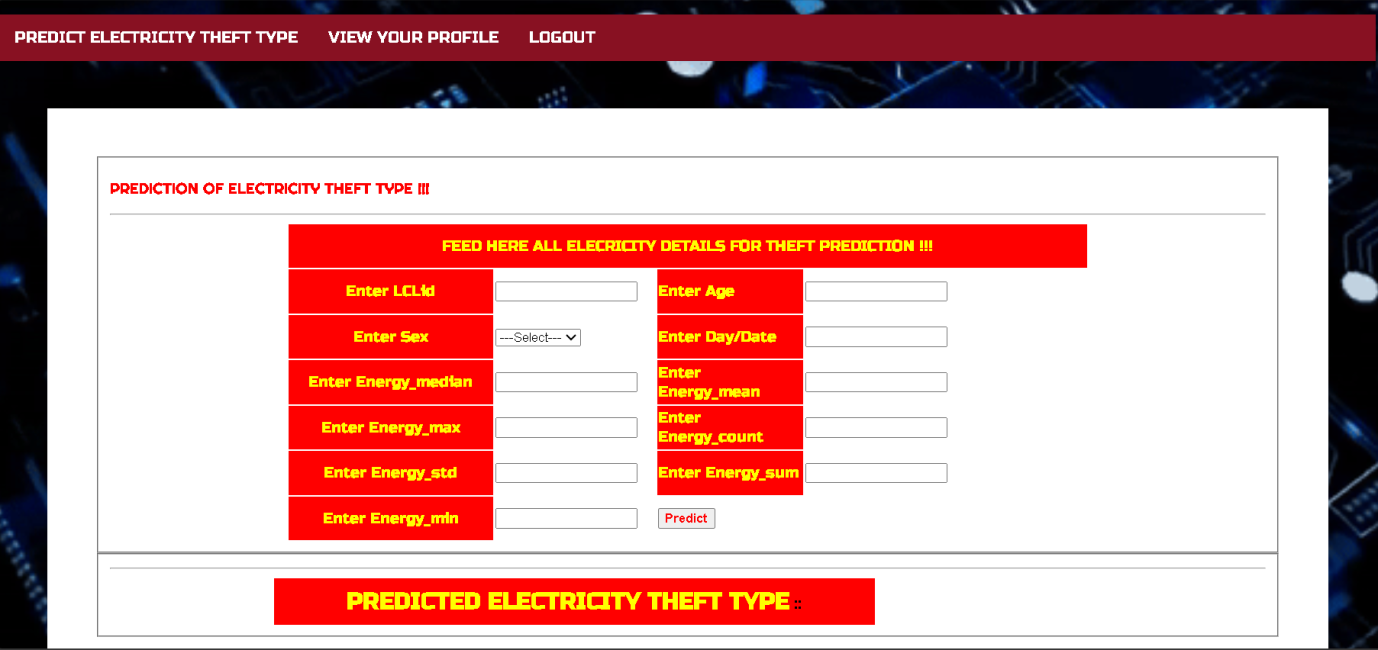
**Figure 8.2.1: User LOGIN page**

3.This is the screen which user gets, after successful login.This screen shows the Profileof the user. To train the model using the dataset, user has to click on the **Predict Electricity Theft Type** option from the main menu.

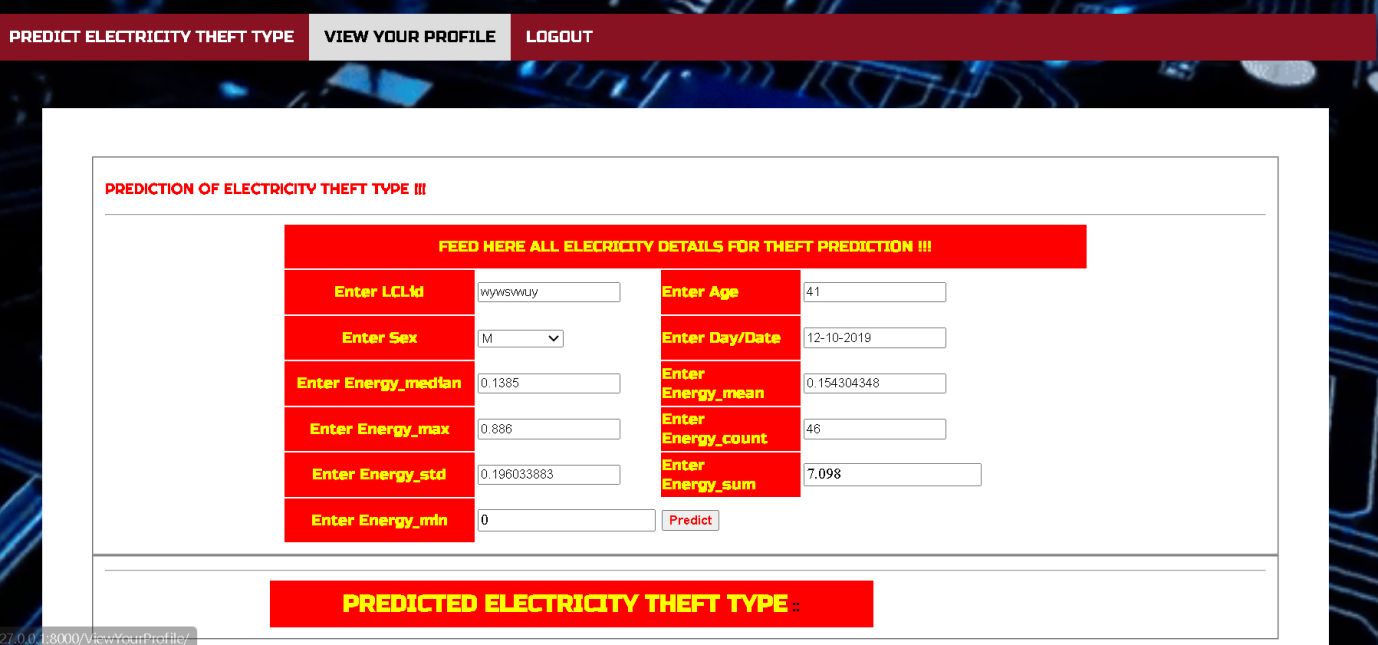


**Figure 8.2.2: User Profile**

4.This is the screen which contains a forms that asks to input the values from the dataset.



5.After entering all the values and clicking the Predict button the **Class Label** is predicted.(theft/no theft)





**Figure 8.2.4: Predict Electricity Theft type**

**CHAPTER – 9**

**CONCLUSION**

In this work, the detection of electricity theft in smart grids was investigated using time-domain and frequency-domain features in a DNN-based classification approach. Isolated classification tasks based on the time-domain, frequency domain and combined domains features were investigated on the same DNN network. Widely accepted performance metrics such as recall, precision, F1-score, accuracy, AUCROC and MCC were used to measure the performance of the model. We observed that classification done with frequency-domain features outperforms classification done with time-domain features, which in turn is outperformed by classification done with features from both domains.

The classifier was able to achieve 87.3% accuracy and 93% AUC-ROC when tested. We used PCA for feature reduction. With 7 out of 20 components used, the classifier was able to achieve 85.8% accuracy and 92% AUC-ROC when tested. We further analyzed individual features' contribution to the classification task and confirmed with the MRMR algorithm the importance of frequency-domain features over time-domain features towards a successful classification task. For better performance, a Bayesian optimizer was also used to optimize hyper parameters, which realized accuracy improvement close to 1%, on validation. Adam optimizer was incorporated and optimal values of key parameters were investigated.

In comparison with other data-driven methods evaluated on the same dataset, we obtained 97% AUC which is 1% higher than the best AUC in existing works, and 91.8% accuracy, which is the second-best on the benchmark. The method used here utilizes consumption data patterns. Our work brings a small contribution towards accurately detecting energy theft as we detect theft that only took place over time. We wish to extend our method to detect real-time electricity theft in the future. Since this method was evaluated based on consumption patterns of SGCC customers, it can further be validated against datasets from different areas to ensure its applicability anywhere.

**CHAPTER - 10**

**FUTURE ENHANCEMENT**

1. Integrate multi-model data including weather and customer behavior.
2. Apply advanced signal processing for Feature Extraction.
3. Handle missing data with robust imputation and address dynamic class imbalance through effective methods.
4. Enable real-time theft detection by ADAPTING MODELS WITH NEW PATTERNS THROUGH ONLINE LEARNING.
5. Continuously benchmark performance against either methods and incorporate ML/DL methods.

**CHAPTER -11**

**REFERENCES**

[1] S. Foster. (Nov. 2, 2021). Non-Technical Losses: A $96 BillionGlobal Opportunity for Electrical Utilities. [Online]. Available:https://energycentral.com/c/pip/ non-technical-losses-96-billion-globalopportunity-electrical-utilities

[2] Q. Louw and P. Bokoro, ``An alternative technique for the detection andmitigation of electricity theft in South Africa,'' SAIEE Afr. Res. J., vol. 110,no. 4, pp. 209\_216, Dec. 2019. [3] M. Anwar, N. Javaid, A. Khalid, M. Imran, and M. Shoaib, ``Electricitytheft detection using pipeline in machine learning,'' in Proc. Int. . Mobile Comput. (IWCMC), Jun. 2020, pp. 2138\_2142.

[4] Z. Zheng, Y. Yang, X. Niu, H.-N. Dai, and Y. Zhou, ``Wide and deepconvolutional neural networks for electricity-theft detection to securesmart grids,'' IEEE Trans. Ind. Informat., vol. 14, no. 4, pp. 1606\_1615,Apr. 2018.

[5] P. Pickering. (Nov. 1, 2021). E-Meters Offer Multiple Ways toCombat Electricity Theft andTampering. [Online]. Available:https://www.electronicdesign.com/technologies/meters

[6] X. Fang, S. Misra, G. Xue, and D. Yang, ``Smart grid The new and improved power grid: A survey,'' IEEE Commun. Surveys Tuts., vol. 14,no. 4, pp. 944\_980, 4th Quart., 2012.

[7] M. Ismail, M. Shahin, M. F. Shaaban, E.Serpedin, and K. Qaraqe, ``Efficient detection of electricity theft cyber attacks in AMI networks,'' in Proc. IEEE Wireless Commun. Netw. Conf. (WCNC), Apr. 2018, pp. 1\_6.

[8] A. Maamar and K. Benahmed, ``Machine learning techniques for energytheft detection in AMI,'' in Proc. Int. Conf. Softw. Eng. Inf. Manage.(ICSIM), 2018, pp. 57\_62.

[9] A. Jindal, A. Schaeffer-Filho, A. K. Marnerides, P. Smith, A. Mauthe, andL. Granville, ``Tackling energy theft in smart grids through data-drivenanalysis,'' in Proc. Int. Conf. Comput., Netw. Commun. (ICNC), Feb. 2020,pp. 410\_414.

[10] I. Diahovchenko, M. Kolcun, Z. .onka, V. Savkiv, and R. Mykhailyshyn,``Progress and challenges in smart grids: Distributed generation, smartmetering, energy storage and smart loads,'' Iranian J. Sci. Technol., Trans.Electr. Eng., vol. 44, no. 4, pp. 1319\_1333, Dec. 2020.

[11] M. Jaganmohan. (Mar. 3, 2022). Global Smart GridMarket Size by Region 2017\_2023.[Online].Available: <https://www.statista.com/statistics/246154/global-smart-grid-marketsize>by-region/

[12] Z. Zheng, Y. Yang, X. Niu, H.-N. Dai, and Y. Zhou.(Sep. 30, 2021). Electricity Theft Detection, [Online]. Available:https://github.com/henryRDlab/ ElectricityTheftDetection

[13] D. O. Dike, U. A. Obiora, E. C. Nwokorie, and B. C. Dike, ``Minimizinghousehold electricity theft in Nigeria using GSM based prepaid meter,''Amer. J. Eng. Res., vol. 4, no. 1, pp. 59\_69, 2015.

[14] P. Dhokane, M. Sanap, P. Anpat, J. Ghuge, and P. Talole, ``Power theftdetection &initimate energy meter information through SMS with autopower cut off,'' Int. J. Current Res. Embedded Syst. VLSI Technol., vol. 2,no. 1, pp. 1\_8, 2017.

[15] S. B. Yousaf, M. Jamil, M. Z. U. Rehman, A. Hassan, and S. O. G. Syed,``Prototype development to detect electric theft using PIC18F452 microcontroller,''Indian J. Sci. Technol., vol. 9, no. 46, pp. 1\_5, Dec. 2016.

[16] K. Dineshkumar, P. Ramanathan, and S. Ramasamy, ``Development ofARM processor based electricity theft control system using GSM network,''in Proc. Int. Conf. Circuits, Power Comput. Technol. (ICCPCT),Mar. 2015, pp. 1\_6.

[17] S. Ngamchuen and C. Pirak, ``Smart anti-tampering algorithm design forsingle phase smart meter applied to AMI systems,'' in Proc. 10th Int. Conf.Electr. Eng./Electron., Comput., Telecommun. Inf. Technol., May 2013,pp. 1\_6.

[18] B. Khoo and Y. Cheng, ``Using RFID for anti-theft in a Chinese electricalsupply company: A cost-bene\_t analysis,'' in Proc. Wireless Telecommun.Symp. (WTS), Apr. 2011, pp. 1\_6.

[19] J. Astronomo, M. D. Dayrit, C. Edjic, and E. R. T. Regidor, ``Developmentof electricity theft detector with GSM module and alarm system,'' in Proc.IEEE 12th Int. Conf. Humanoid, Nanotechnol., Inf. Technol., Commun.Control, Environ., Manage. (HNICEM), Dec. 2020, pp. 1\_5.

[20] P. Jokar, N. Arianpoo, and V. C. M. Leung, ``Electricity theft detection inAMI using customers' consumption patterns,'' IEEE Trans. Smart Grid,vol. 7, no. 1, pp. 216\_226, Jan. 2015.

[21] W. Han and Y. Xiao, ``A novel detector to detect colluded non-technicalloss frauds in smart grid,'' Comput. Netw., vol. 117, pp. 19\_31, Apr. 2017.

[22] S. Sahoo, D. Nikovski, T. Muso, and K. Tsuru, ``Electricity theft detectionusing smart meter data,'' in Proc. IEEE Power Energy Soc. Innov. SmartGrid Technol. Conf. (ISGT), Feb. 2015, pp. 1\_5.

[23] S. K. Singh, R. Bose, and A. Joshi, ``PCA based electricity theft detectionin advanced metering infrastructure,'' in Proc. 7th Int. Conf. Power Syst.(ICPS), Dec. 2017, pp. 441\_445.

[24] M. Di Martino, F. Decia, J. Molinelli, and A. Fernández, ``Improving electricfraud detection using class imbalance strategies,'' in Proc. ICPRAM,2012, pp. 135\_141.

[25] M. N. Hasan, R. N. Toma, A.-A. Nahid, M. M. M. Islam, and J.-M. Kim,``Electricity theft detection in smart grid systems: A CNN-LSTM basedapproach,'' Energies, vol. 12, no. 17, p. 3310, 2019.