IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Artificial Intelligence Based Phase Balancing in the Smart Grid

NISHANTHINI R (620322403008)

சுருக்கம் (ABSTRACT)

ஸ்மார்ட் கிரிட்டில் மேம்பட்ட அளவீட்டு உள்கட்டமைப்பு, கட்டுப்பாட்டு தொழில்நுட்பங்கள் மற்றும் தகவல் தொடர்பு தொழில்நுட்பங்களின் ஒருங்கிணைப்பு, மின்சார சக்தி கட்டம் செயல்பாடுகள் தொடர்பான உயர் பரிமாண மற்றும் பல வகை தரவுகளின் பரந்த தொகுதிகளை சேகரிக்க உதவுகிறது. ஆயினும்கூட, பாரம்பரிய மாதிரியாக்கம், தேர்வுமுறை மற்றும் கட்டுப்பாட்டு முறைகள் அத்தகைய தரவை திறம்பட கையாள்வதில் பல வரம்புகளை சந்திக்கின்றன. இதன் விளைவாக, ஸ்மார்ட் கிரிட்டில் செயற்கை நுண்ணறிவு (AI) நுட்பங்களைப் பயன்படுத்துவது முக்கியத்துவம் பெறுகிறது. சுமை முன்கணிப்பு, பவர் கிரிட் நிலைத்தன்மை மதிப்பீடு, தவறு கண்டறிதல் மற்றும் ஸ்மார்ட் கிரிட் மற்றும் பவர் சிஸ்டங்களில் உள்ள பாதுகாப்புச் சிக்கல்கள் ஆகியவற்றிற்குப் பயன்படுத்தப்படும் பல்வேறு AI நுட்பங்கள் குறித்த தற்போதைய ஆராய்ச்சியை இந்தக் கணக்கெடுப்பு முறையாக மதிப்பாய்வு செய்கிறது. மேலும், உண்மையான அறிவார்ந்த கட்ட அமைப்புகளை அடைவதற்கு AI தொழில்நுட்பங்களை செயல்படுத்துவதில் உள்ள ஆராய்ச்சி சவால்களை இது அடையாளம் காட்டுகிறது. இறுதியாக, இந்த கணக்கெடுப்பு ஸ்மார்ட் கிரிட் சவால்களை எதிர்கொள்வதில் AI இன் சாத்தியமான பயன்பாடுகளை எடுத்துக்காட்டுகிறது. இறுதியில், AI நுட்பங்களை ஒருங்கிணைப்பது ஸ்மார்ட் கிரிட் அமைப்புகளின் நம்பகத்தன்மை மற்றும் பின்னடைவை கணிசமாக மேம்படுத்தும் என்று காகிதம் முடிவு செய்கிறது.

ABSTRACT

The integration of advanced metering infrastructure, control technologies, and communication technologies within the smart grid facilitates the collection of vast volumes of high-dimensional and multi-type data concerning electric power grid operations. Nonetheless, traditional modeling, optimization, and control methods encounter numerous limitations in handling such data effectively. As a result, the utilization of artificial intelligence (AI) techniques within the smart grid is gaining prominence. This survey systematically reviews existing research on various AI techniques applied to load forecasting, power grid stability assessment, fault detection, and security issues within the smart grid and power systems. Furthermore, it identifies ongoing research challenges in implementing AI technologies to achieve truly intelligent grid systems. Finally, this survey highlights the potential applications of AI in addressing smart grid challenges. Ultimately, the paper concludes that integrating AI techniques can significantly enhance the reliability and resilience of smart grid systems.

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The conventional electric power grid is undergoing a transformation into a smart grid, transitioning from electromechanical control to electronic network management. As outlined in the US Department of Energy's Smart Grid System Report, smart grid systems integrate information management, control technologies, digital sensing, communication technologies, and field devices to coordinate electric processes effectively. These technologies revolutionize grid planning and operation by enabling real-time monitoring, data communication, automated response mechanisms, data sharing, and sophisticated data processing. Key challenges in smart grid operation include load forecasting (LF), power grid stability assessment, fault detection (FD), and security. These challenges necessitate the collection of vast amounts of high-dimensional data, highlighting the limitations of traditional modeling, optimization, and control methods. Consequently, the application of artificial intelligence (AI) techniques in the smart grid becomes increasingly essential.

AI techniques leverage extensive data to develop intelligent systems capable of tasks requiring human-like intelligence. While machine learning (ML) is a subset of AI, other approaches such as neural networks, robotics, expert systems (ES), fuzzy logic (FL), and natural language processing contribute to AI advancements. AI facilitates rapid and accurate decision-making, mimicking cognitive functions to achieve self-healing capabilities within smart grid systems. However, challenges persist, and AI may not entirely replace human operators due to complex constraints and requirements. The evolution of AI systems in the smart grid includes virtual AI aiding operators and physical AI optimizing grid operations autonomously or with minimal human intervention. These systems vary from artificial narrow intelligence (ANI) designed for specific tasks to artificial general intelligence (AGI) capable of autonomous learning and evolution akin to humans.

Research into AI for smart grid applications has surged in the past decade, with recent surveys providing insights into various AI techniques. This paper aims to complement existing reviews by exploring AI applications not covered comprehensively, discussing challenges, and outlining future opportunities in smart grid research. The systematic survey presented herein categorizes existing research, identifies challenges, and suggests future research directions.

1.2 ARTIFICIAL INTELLIGENCE TECHNIQUES

The modern power system is undergoing rapid evolution, incorporating a myriad of distributed smart grid components such as smart metering infrastructure, communication infrastructure, distributed energy resources, and electric vehicles. These components form a complex electrical power network intertwined with communication systems, generating vast amounts of data. This data plays a crucial role in automating and enhancing smart grid performance through applications like distributed energy management, system state forecasting, fault detection, and cyber-attack security. However, conventional computational techniques struggle to process the immense volume of data introduced by smart grid systems, prompting a surge in attention towards artificial intelligence (AI) techniques. Numerous research efforts have focused on leveraging AI to address these challenges, harnessing large-scale data to optimize smart grid performance.

AI techniques in the smart grid can be broadly classified into the following categories:

- Expert Systems (ES): Utilizing human expertise in solving specific problems within the grid.
- Supervised Learning: Analyzing input-output mappings to predict outputs for new inputs.
- Unsupervised Learning: Exploring unlabeled data to identify similarities and differences.
- Reinforcement Learning (RL): Employing intelligent agent strategies to maximize cumulative rewards.

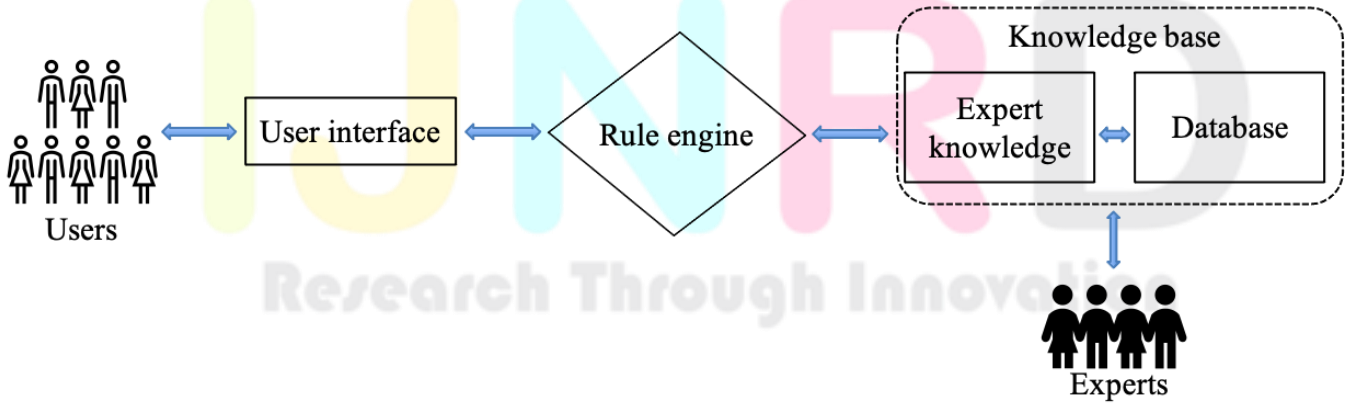
- Ensemble Methods: Integrating results from multiple AI algorithms to overcome individual limitations and achieve improved overall performance.

1.3 EXPERT SYSTEMS

The Expert System (ES), depicted in Figure 1, represents the initial generation of intelligent systems designed to supplant human experts in specific domains and solve particular problems using Boolean logic. Many smart grid challenges, including fault diagnosis, intelligent control, and energy router self-determination, continue to rely on ES techniques. The ES's knowledge base integrates domain expertise and databases, forming its core component. This knowledge base comprises rules expressed as if-then statements linked by logical operations, derived directly from domain experts or research findings. The ES draws conclusions by evaluating these rules against user-input data, facilitated by an intermediate rule engine.

Fuzzy Logic (FL) was introduced to handle partial truth concepts, contrasting with the binary logic of ES. FL, rooted in fuzzy set theory, assigns membership degrees ranging from 0 to 1, allowing for representation of partial truths or falsities. For instance, FL may assign 0 for false, 1 for true, and fractional values between 0 and 1 for partial truths or falsities. Fuzzy Inference Systems (FIS) convert crisp input variables into fuzzy variables, applying fuzzy operators to infer results from rule conditions and defuzzifying outputs to obtain crisp values. The Mamdani and Sugeno methods are popular FIS-based approaches, employing multiple rules to determine degrees of fulfillment.

Figure 1. ES diagram.



SUPERVISED LEARNING

Supervised learning involves the ML task of constructing general hypotheses by connecting labeled input and output pairs, allowing for predictions of future data after training. Over the past two decades, a plethora of supervised learning algorithms have been developed and widely implemented to enhance smart grid systems. Figure 2 showcases common supervised learning algorithms utilized

in the smart grid domain. Artificial neural networks (ANNs), inspired by the biological nervous system, have significantly impacted various fields over the last decade. Similar to many other ML techniques, ANNs do not require explicit programming but instead utilize algorithms to make predictions based on data. They efficiently tackle image processing and pattern recognition challenges that traditional methods struggle with.

Extreme learning machines (ELMs), employing a single hidden layer feed forward neural network, represent one such ANN algorithm and have found application in addressing smart grid problems such as power system stability assessment and fault detection. The back-propagation neural network (BPNN), proposed for the learning process of neural networks, adjusts network weights iteratively until the error between the output and ground truth reaches a specific threshold. Widely used in various neural network algorithms, BPNN is a cornerstone of the field. Multilayer perceptron, another feed forward neural network algorithm, is extensively employed in smart grid applications.

Probabilistic neural networks (PNNs), utilizing the parent probability distribution function of each class to estimate class probabilities, offer yet another robust feed forward neural network model.

Driven by burgeoning datasets and the necessity to address increasingly complex problems, a wave of new AI algorithms has emerged, propelling AI into what is termed the AI 2.0 stage. Deep learning (DL), a subset of ML initially developed for image processing, initially featured multilayer deep neural networks (DNNs). DL techniques have since undergone rapid development, with numerous successful architectures proposed to tackle smart grid challenges, including deep belief networks, convolutional neural networks (CNNs), recurrent neural networks (RNNs), generative adversarial networks, and auto encoders.

Beyond the mentioned algorithms, various AI methods are employed for classification and regression tasks. Support vector machine (SVM), a robust classification model proposed by Vapnik, stands out among them. The k-nearest neighbors (KNN) algorithm, known for its speedy training, finds application in classification and regression within smart grid systems. Decision tree learning models and logistic regression, prized for their interpretability and ease of implementation, have seen widespread adoption in smart grid systems. Regression methods like linear regression (LR), Gaussian process regression (GPR), support vector regression (SVR), and multivariate adaptive regression splines (MARS) offer solutions for forecasting, fault detection, demand response, and other smart grid challenges.

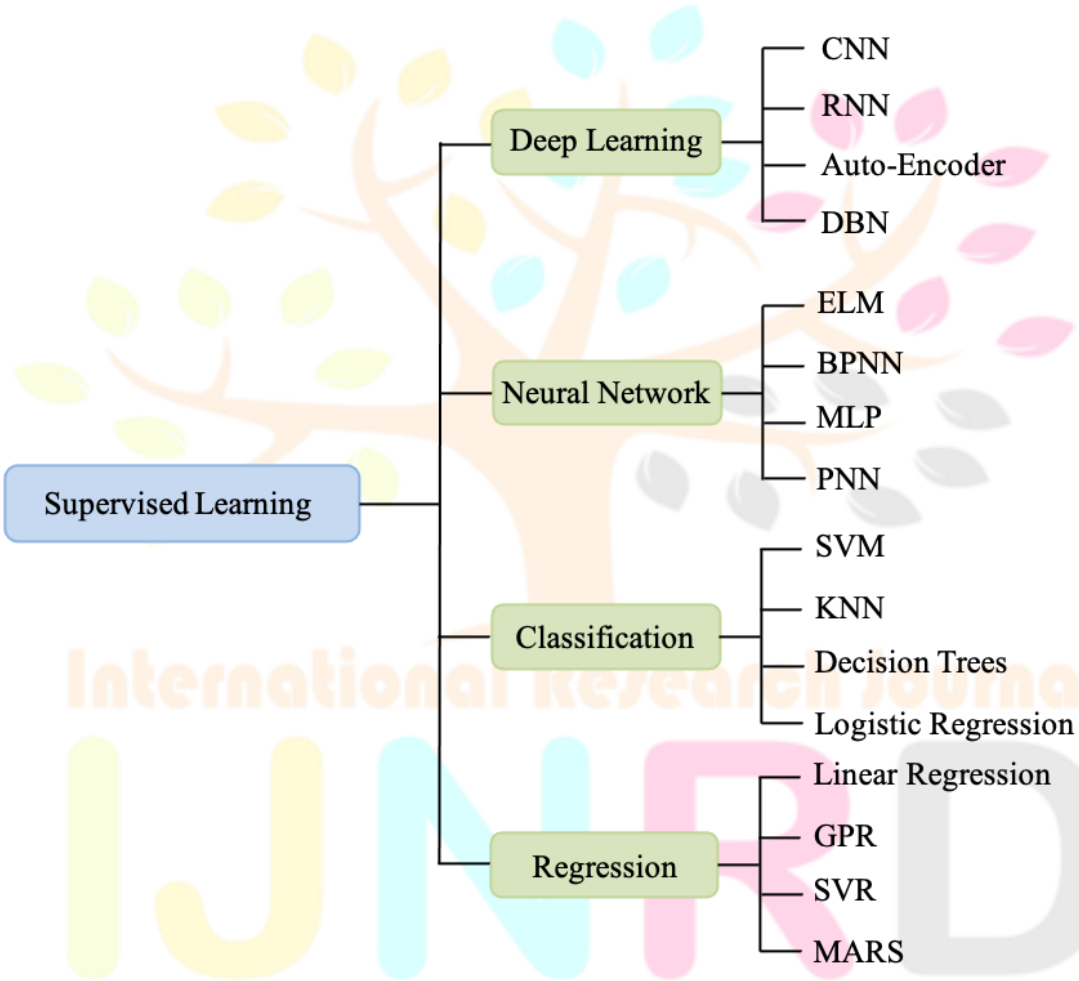


Figure 2. Supervised learning techniques in the smart grid.

REINFORCEMENT LEARNING

Reinforcement learning (RL) has gained significant traction in addressing smart grid challenges. Its fundamental components include the agent, environment, reward, and action. RL operates by continuously seeking to maximize cumulative rewards through a cycle of receiving feedback—both rewards and penalties—on each action taken. Despite limited knowledge of the environment and feedback on decision quality, RL adapts effectively to unforeseen circumstances. Figure 4 outlines commonly employed RL algorithms.

Q-learning and SARSA (state–action–reward–state–action) find applications in attack detection and energy management within smart grid systems. Deep reinforcement learning (DRL) merges the perceptual capabilities of deep learning (DL) with the decision-making prowess of RL. Notably, Alpha Go exemplifies the success of DRL through its adept utilization of rich perceptual input and policy control. Deep Q network and deep deterministic policy gradient stand out as popular DRL algorithms applied in smart grid contexts.

UNSUPERVISED LEARNING

While supervised learning algorithms have demonstrated remarkable performance over decades of development, their effectiveness relies on the availability of ground truth or predefined patterns, which may not always be assured in real-world scenarios. This limitation underscores the value of unsupervised learning, which can uncover latent information and detect hidden patterns in data without the need for labels. Common unsupervised learning algorithms are listed in Figure 3.

Unsupervised neural networks, including restricted Boltzmann machines, auto encoders, and variation auto encoders, find applications in anomaly detection, stability assessment, load forecasting, among others. Clustering, an unsupervised task, involves grouping data points or populations into clusters where items within the same cluster exhibit similarity. K-means, fuzzy c-means, hierarchical clustering, and DBSCAN (density-based spatial clustering of applications with noise) are frequently employed for fault detection and load forecasting.

Dimensionality reduction (DR) techniques are essential for transforming data from high-dimensional spaces to lower-dimensional ones, particularly in processing smart grid data to eliminate redundant features. Common DR methods used in the smart grid domain include principal component analysis (PCA), linear discriminant analysis, generalized discriminant analysis, and non-negative matrix factorization.

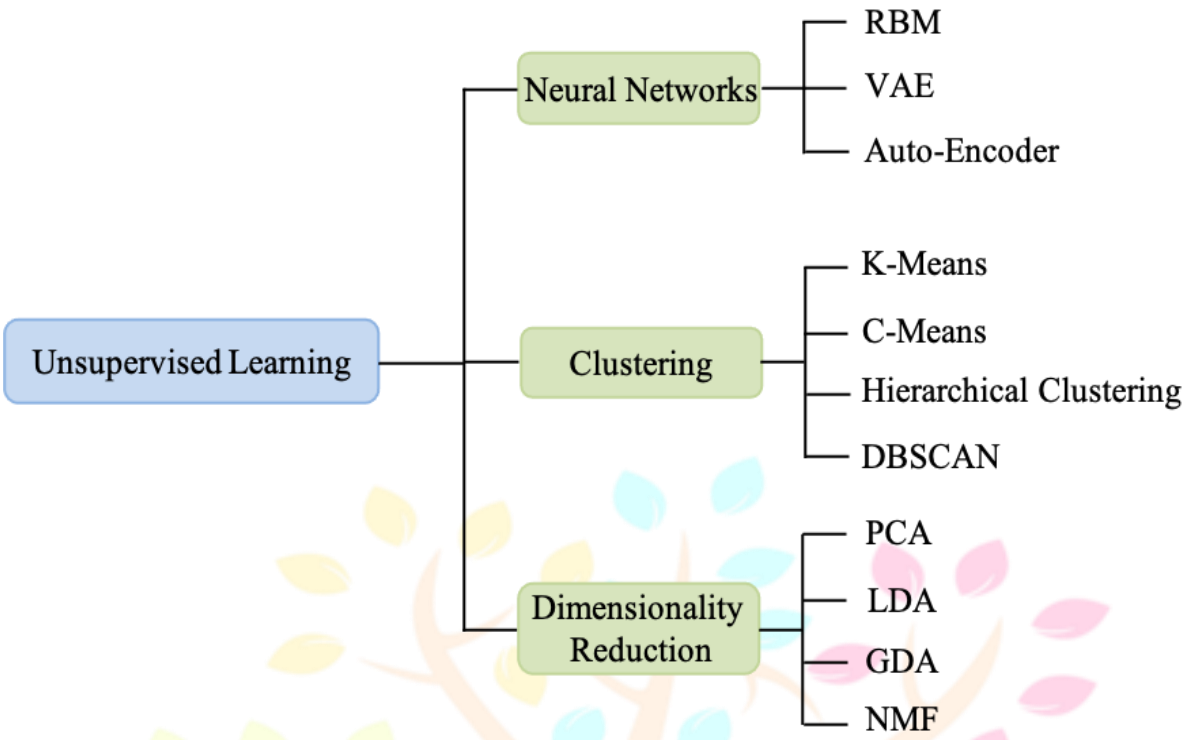


Figure 3. Unsupervised learning techniques diagram.

ENSEMBLE METHODS

Ensemble methods enhance overall performance by combining results from multiple learning algorithms or diverse initial datasets. Bootstrap aggregating, also known as bagging, assigns equal weight to each model in the ensemble, training them on random subsets of data. Random forest, a successful bagging model, merges random decision trees with a high-performance classification algorithm. Its applications extend to load forecasting, anomaly detection, and stability assessment. Boosting, another ensemble method, constructs a new model aimed at rectifying misclassifications made by the previous model, yielding promising outcomes in smart grid challenges. Stacking, an ensemble learning technique, amalgamates predictions from various classification or regression algorithms, demonstrating robust capabilities in load forecasting, anomaly detection, and cyberattack detection.

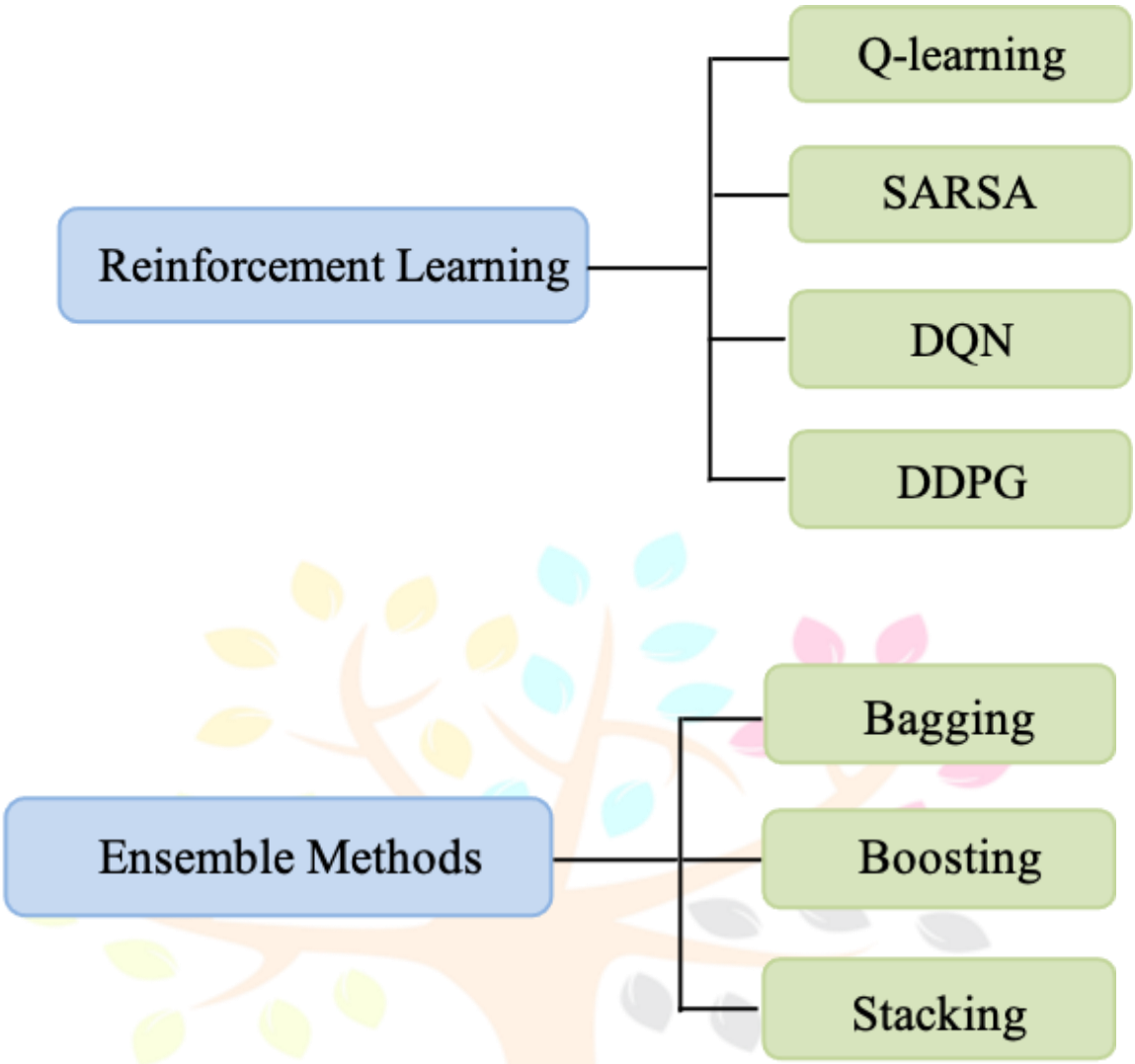


Figure 4. RL and ensemble methods.

ARTIFICIAL INTELLIGENCE TECHNIQUES IN SMART GRIDS

This section presents a review of AI techniques in smart grids.

Predictive Analytics

Predictive analytics is a method that gathers big data and presents t in a way that is well understood. The main aim of predictive data is to present current and historical data to plan for the future. It uses methods and tools that are either supervised, semi-supervised, or unsupervised. Predictive analytics is categorized into two; machine learning and regression techniques. To properly implement parallelism in machine learning, the authors utilize predictive knowledge from Neural Networks using Graphics Processing Unit Machine Learning Library (GPUMLib) by applying algorithms like Self-Organizing Maps and Multiple Backpropagation.

Importance of Predictive Analytics

Predictive analytics has been used in many companies and institutions due to its vast advantages. It improves customer service since all the complaints are analyzed, and proper responses are taken to action. The analytics are quick to offer recommendations; hence there is understanding. Predictive analytics systems can detect fraud and help the company to prevent it in the future. Other importance includes reducing risks, improving efficiency, reducing cost, and helping in the identification of opportunities that add value to the company.

Prescriptive Analytics

Prescriptive analytics utilize is a method that utilizes data from descriptive and predictive analytics to make decisions. It makes use of AI to collect big data that overwhelm humans. Prescriptive analytics may not be accurate; hence data scientists need to monitor the systems to ensure missing or incorrect data. Model overfitting can lead to wrong predictions. Data scientists should exercise machine learning algorithms and functionalities to build a predictive analytics system because different algorithms assume different data structures and completeness. For example, when using a linear regression model, it is assumed that the prediction variable can be represented as a weighted sum of the descriptive characteristics. In practice, however, not all data is linearly connected, and as a result, linear regression cannot be used to solve every data science problem in every situation.

Importance of Prescriptive Analytics

Prescriptive analytics make use of artificial intelligence in place of human intervention. Although we cannot downplay human power, IA can collect big data and organize it so that the algorithms can use it to make decisions. Decisions made from prescriptive analytics systems increase the performance of the business since all factors have been checked and predictions made right. If there is an error either in big

data collection or organization, the prescriptive analytics algorithms can recommend solutions immediately since it is automated.

Role of Finance and Accounting Departments in using Predictive and Prescriptive analytics

The two departments have the role of standardizing the most crucial information in the organization, that is, master data. They are mandated to prepare an excel sheet for data presentation used in decision making. The decision-making model operationalizes thinking to be used by accounting and finance managers. The two departments have a responsibility to unite other departments while using the two analytics. Other roles of finance and accounting departments include pinpointing outliers across the firm based on a continuous assessment, utilizing open-source components to be used in the supply chains like MySQL, end-to-end processing, and machine learning, and establishing and maintaining a list of approved external data sources to supplement existing internal data, with the list being updated regularly to reflect any new sources that have become available.

Technology and Skills Limitation in Finance Department

Organizations have introduced big data (BD) that is overwhelming employees in collecting data. A company can receive thousands of data each day that needs to be analyzed and related to other historical data. Most finance departments lack automated systems to collect and organize data. The limitation can be overcome by installing an automated system that will give employees an easy time processing data. Another limitation in the finance department is the lack of a strong data system for visual representation. Data is always presented visually in charts or graphs for easy understanding. Manual putting the data in the reporting tool may be tiresome and time-consuming. Traditionally, finance departments manually fit the data. The finance department can address the problem by having strong data systems to build reporting for easy decision-making. Employees from the entire organization face confusion and anxiety about changing from traditional data entry even with the knowledge of the importance of automation and analytics. The finance team should comprehensively educate the employees on the meaningfulness of using analytics.

Many finance departments face the problem of analyzing data due to a lack of skilled labor. A couple of employees may lack the capacity to deal with comprehensive analysis. The finance department can overcome this limitation by enhancing the hiring process and hiring according to competency and making all employees' analytics easy to use. Finally, finance departments are faced with the challenge of accessing necessary data for use. They should hence employ an effective database to counter such issues.

Real-life Application for Predictive Analytics

Predictive analytics is widely used in the healthcare industry. In 2019, Digital Health Cooperative Research Centre partnered with the Royal Melbourne Institute of Technology and came up with software to elevate clinical decision-making for the aged. The software predicts the health condition of the aged to avoid emergency hospitalizations using predictive analytics algorithms. Predictive analytics help the family, residents and healthcare providers to plan ahead before the worst happens.

Weather forecasting utilizes predictive analytics to predict weather patterns. By using historical data and satellite imagery, weather estimates are drawn even a month in advance. Additionally, the information from the estimates can be used to find the impact of global warming.

Real-life Application of Prescriptive Analytics

Prescriptive analytics is used in navigation. Prescriptive analytics is widely used in GPS technology to give suggestions on routes that can be used by giving the users a way to reach to their destination according to the during of the journey and road closures. Prescriptive analytic tools are used to calculate the distance between your starting point and the stopping/destination point and predict the shortest and the quickest way.

Another example is an insurance company that uses prescriptive analytics to determine when it is most beneficial for them to pay new claims versus appealing them. The insurance company collects data on incoming claims; it takes time to process the claim and the different outcomes of these claims. By using predictive analysis, the insurer can forecast whether a particular claim can be paid or appealed based on the data surrounding the event.

Back-end Code: (AI Algorithm)

```
using DashboardApp.Models;
using DashboardApp.Services;
using Microsoft.AspNetCore.Mvc;
using System;
using System.Diagnostics;

namespace DashboardApp.Controllers;

public class HomeController : Controller
{
    private readonly ILogger<HomeController> _logger;

    public HomeController(ILogger<HomeController> logger)
    {
        _logger = logger;
    }

    public async Task<IActionResult> Index()
    {
        try
        {
            DBService<ReportData> _dbService = new();
            List<ReportData> data = await _dbService.GetDataAsync();

            return View(data);
        }
        catch (Exception ex)
        {
            return RedirectToAction("Error", ex);
        }
    }

    public async Task<IActionResult> Users()
    {
        try
        {
            DBService<User> _dbService = new();
            List<User> users = await _dbService.GetDataAsync();

            return View(users);
        }
        catch (Exception ex)
        {
            return RedirectToAction("Error", ex);
        }
    }

    public async Task<IActionResult> AddOrEdit(int id)
    {
        try
        {
```

```
        if (id == 0) { return View(); }
        else
        {
            DBService<User> _dbService = new();
            List<User> users = await _dbService.GetDataAsync(id);

            return View(users.FirstOrDefault());
        }
    }
    catch (Exception ex)
    {
        return RedirectToAction("Error", ex);
    }
}

public async Task<IActionResult> Save(User userRequest)
{
    try
    {
        DBService<User> _dbService = new();
        List<User> users = await _dbService.GetDataAsync();
        userRequest.UserId = users.Count + 1;
        User user = await _dbService.InsertDataAsync(userRequest.UserId, userRequest);

        return RedirectToAction("Users");
    }
    catch (Exception ex)
    {
        return RedirectToAction("Error", ex);
    }
}

public async Task<IActionResult> Update(User userRequest)
{
    try
    {
        DBService<User> _dbService = new();
        User user = await _dbService.UpdateDataAsync(userRequest.UserId, userRequest);

        return RedirectToAction("Users");
    }
    catch (Exception ex)
    {
        return RedirectToAction("Error", ex);
    }
}

public async Task<IActionResult> Delete(int id)
{
    try
    {
        DBService<User> _dbService = new();
        bool isDeleted = await _dbService.DeleteDataAsync(id);
```

```
            return RedirectToAction("Users");
        }
        catch (Exception ex)
        {
            return RedirectToAction("Error", ex);
        }
    }

    public async Task<IActionResult> RunAIPoweredSimulator(double solarValue, double windValue,
double thermalValue)
    {
        try
        {
            DBService<ReportData> _dbService = new();

            List<ReportData> data = new();

            ReportData reportData = await _dbService.InsertDataAsync(0, new()
            {
                Type = "Solar",
                Value = solarValue,
                CreatedAt = DateTime.Now,
            });

            data.Add(reportData);

            reportData = await _dbService.InsertDataAsync(0, new()
            {
                Type = "Wind",
                Value = windValue,
                CreatedAt = DateTime.Now,
            });

            data.Add(reportData);

            reportData = await _dbService.InsertDataAsync(0, new()
            {
                Type = "Thermal",
                Value = thermalValue,
                CreatedAt = DateTime.Now,
            });

            data.Add(reportData);

            return Ok(data);
        }
        catch (Exception ex)
        {
            return BadRequest(ex);
        }
    }
```

```
    public async Task<IActionResult> AnalyzeAI(dynamic data)
    {
        try
        {
            bool isProccessed = await AnalyzePrccess(data);

            return RedirectToAction("Index");
        }
        catch (Exception ex)
        {
            return RedirectToAction("Error", ex);
        }
    }

    public async Task<IActionResult> AnalyzePrccess(dynamic data)
    {
        try
        {
            // AnalyzePrccess(data)

            return data;
        }
        catch (Exception ex)
        {
            return RedirectToAction("Error", ex);
        }
    }

    [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
    public IActionResult Error()
    {
        return View(new ErrorViewModel { RequestId = Activity.Current?.Id ??
HttpContext.TraceIdentifier });
    }
}
```

Front-end Code: (Webpage)

```
@model List<ReportData>

@{
    ViewData["Title"] = "Dashboard";
}

<style>
    .card-header {
        font-size: 1.2rem;
    }

    .card-body {
        padding: 20px;
    }

    canvas {
        height: 250px;
    }

    .double-input {
        display: flex;
        align-items: center;
    }

    .double-input input {
        flex: 1;
        border: 1px solid #ccc;
        padding: 5px;
        margin-right: 5px;
    }

    .input-with-percentage {
        position: relative;
        display: inline-block;
    }

    .percentage-symbol {
        position: absolute;
        right: 8px;
        top: 50%;
        transform: translateY(-50%);
    }

    #toast-container {
        position: fixed;
        bottom: 20px;
        left: 50%;
        transform: translateX(-50%);
        z-index: 9999;
    }
```

```
    .toast {
      background-color: #333;
      color: #fff;
      padding: 10px 20px;
      border-radius: 5px;
      box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);
      opacity: 0;
      transition: opacity 0.3s ease-in-out;
    }

    .show {
      opacity: 1;
    }
</style>

<div class="container">
    <div class="row">
        <div class="col-md-6 mb-4">
            <div class="card">
                <div class="card-header bg-secondary text-white">
                    Load Power Factor
                </div>
                <div class="card-body">
                    <label for="loadType">Select Season:</label>
                    <select id="loadType" class="form-control" onchange="loadTypeChanged()">
                        <option value="summer">Summer</option>
                        <option value="winter">Winter</option>
                        <option value="spring">Spring</option>
                        <option value="rainy">Rainy</option>
                    </select>

                    <label for="powerFactorInput">Power factor:</label>
                    <input type="number" id="powerFactorInput" max="0.99" value="0.99" class="form-
control" placeholder="Enter a number" onchange="validateInput('powerFactorInput')">

                    <label for="loadPowerInput">Load Power:</label>
                    <div class="input-with-percentage p-1">
                        <input type="number" id="loadPowerInput" max="15" value="15" class="form-
control" placeholder="Enter a number" onchange="validateInput('loadPowerInput')">
                        <span class="percentage-symbol">kw</span>
                    </div>

                    <br />

                    <label for="lossInput">Loss:</label>
                    <div class="input-with-percentage p-1">
                        <input type="number" id="lossInput" max="30" value="0" class="form-control"
placeholder="Enter a number" onchange="validateInput('lossInput')">
                        <span class="percentage-symbol">%</span>
                    </div>

                    <div class="double-input p-1 d-none">
                        <label for="solarLoadMinInput">Solar Min load:</label>
```

```
                <input type="number" id="solarLoadMinInput" max="0.99" value="0.90" class="form-control" placeholder="Enter a number" oninput="validateInput('solarLoadMinInput')">

                <label for="solarLoadMaxInput">Solar Max load:</label>
                <input type="number" id="solarLoadMaxInput" max="0.99" value="0.9" class="form-control" placeholder="Enter a number" oninput="validateInput('solarLoadMaxInput')">
            </div>

            <div class="double-input p-1 d-none">
                <label for="windLoadMinInput">Wind Min load:</label>
                <input type="number" id="windLoadMinInput" max="0.99" value="0.70" class="form-control" placeholder="Enter a number" oninput="validateInput('windLoadMinInput')">

                <label for="windLoadMaxInput">Wind Max load:</label>
                <input type="number" id="windLoadMaxInput" max="0.99" value="0.9" class="form-control" placeholder="Enter a number" oninput="validateInput('windLoadMaxInput')">
            </div>

            <div class="double-input p-1 d-none">
                <label for="thermalLoadMinInput">Thermal Min load:</label>
                <input type="number" id="thermalLoadMinInput" max="0.99" value="0.60" class="form-control" placeholder="Enter a number" oninput="validateInput('thermalLoadMinInput')">

                <label for="thermalLoadMaxInput">Thermal Max load:</label>
                <input type="number" id="thermalLoadMaxInput" max="0.99" value="0.6" class="form-control" placeholder="Enter a number" oninput="validateInput('thermalLoadMaxInput')">
            </div>
        </div>
        <div class="card-body text-center">
            <button id="updateButton" onclick="updateLoadInput()" class="btn btn-primary">Update</button>
        </div>
    </div>
</div>
<div class="col-md-6 mb-4">
    <div class="card">
        <div class="card-header bg-primary text-white">
            Consolidated Status
        </div>
        <div class="d-flex">
            <div class="card-body" style="height: 300px; overflow: hidden;">
                <canvas id="pieChart" style="height: 100%;"></canvas>
            </div>
            <div class="card-body" style="height: 300px; overflow: hidden;">
                <canvas id="pieChart1" style="height: 100%;"></canvas>
            </div>
        </div>
    </div>
</div>
<div class="col-md-6 mb-4">
    <div class="card">
        <div class="card-header bg-success text-white">
            Power Factor Chart
```

```
            </div>
            <div class="card-body">
              <canvas id="powerFactorChart"></canvas>
            </div>
          </div>
        </div>
        <div class="col-md-6 mb-4">
          <div class="card">
            <div class="card-header bg-success text-white">
              Loss Chart
            </div>
            <div class="card-body">
              <canvas id="lossChart"></canvas>
            </div>
          </div>
        </div>
        <div class="col-md-6 mb-4">
          <div class="card">
            <div class="card-header bg-success text-white">
              Solar Chart
            </div>
            <div class="card-body">
              <canvas id="lineChart"></canvas>
            </div>
          </div>
        </div>
        <div class="col-md-6 mb-4">
          <div class="card">
            <div class="card-header bg-warning text-white">
              Wind Chart
            </div>
            <div class="card-body">
              <canvas id="barChart"></canvas>
            </div>
          </div>
        </div>
        <div class="col-md-6 mb-4">
          <div class="card">
            <div class="card-header bg-info text-white">
              Thermal Chart
            </div>
            <div class="card-body">
              <canvas id="lineChart1"></canvas>
            </div>
          </div>
        </div>
      </div>
    </div>

    <div id="toast-container"></div>

    @section Scripts {
      <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
```

```
<script>
    var lineChart, barChart, lineChart1, pieChart, pieChart1, powerFactorChart, lossChart;
    var model = @Html.Raw(Json.Serialize(Model));

    var solarMinLoad = 0.80;
    var windMinLoad = 0.60;
    var thermalMinLoad = 0.20;

    var solarMaxLoad = 0.90;
    var windMaxLoad = 0.70;
    var thermalMaxLoad = 0.60;

    var powerFactorInput = 0.99;
    var lossInput = 0;
    var loadPowerInput = 15;

    var today = new Date();
    today.setHours(0);
    today.setMinutes(0);
    today.setSeconds(0);
    today.setMilliseconds(0);

    var presetDate = [];
    presetDate.push(new Date().toLocaleTimeString());

    var powerFactorData = [0.99];
    var lossData = [0];

    var yesterday = new Date(today);
    yesterday.setDate(today.getDate() - 1);

    function genRandom(minLoad, maxLoad) {
        var decimalPart = Math.floor(Math.random() * (maxLoad * 100)) / 100;

        decimalPart = decimalPart < minLoad ? minLoad : decimalPart;

        return decimalPart;
    }

    function runAIPoweredSimulator() {
        var solarRandom = genRandom(solarMinLoad, solarMaxLoad);
        var windRandom = genRandom(windMinLoad, windMaxLoad);
        var thermalRandom = genRandom(thermalMinLoad, thermalMaxLoad);

        $.ajax({
            url: '/Home/RunAIPoweredSimulator?solarValue=' + solarRandom + "&windValue=" +
windRandom + "&thermalValue=" + thermalRandom,
            type: 'GET',
            dataType: 'json',
            success: function (data) {
                model = model.concat(data);

                var todayData = model.filter(obj => new Date(obj.createdAt) > today);
```

```
var todaySolar = todayData.filter(obj => obj.type == "Solar").map(function (obj) {
  return obj["value"];
});

var todaySolarLable = todayData.filter(obj => obj.type == "Solar").map(function (obj) {
  return new Date(obj["createdAt"]).toLocaleTimeString();
});

var todayWindmill = todayData.filter(obj => obj.type == "Wind").map(function (obj) {
  return obj["value"];
});

var todayWindmillLable = todayData.filter(obj => obj.type == "Wind").map(function
(obj) {
  return new Date(obj["createdAt"]).toLocaleTimeString();
});

var todayThermal = todayData.filter(obj => obj.type == "Thermal").map(function (obj) {
  return obj["value"];
});

var todayThermalLable = todayData.filter(obj => obj.type == "Thermal").map(function
(obj) {
  return new Date(obj["createdAt"]).toLocaleTimeString();
});

lineChart.data.datasets[0].data = todaySolar.slice(-25);
lineChart.data.labels = todaySolarLable.slice(-25);
lineChart.update();

barChart.data.datasets[0].data = todayWindmill.slice(-25);
barChart.data.labels = todayWindmillLable.slice(-25);
barChart.update();

lineChart1.data.datasets[0].data = todayThermal.slice(-25);
lineChart1.data.labels = todayThermalLable.slice(-25);
lineChart1.update();

var todaySolarSum = todayData.filter(obj => obj.type == "Solar").reduce(function (acc,
obj) {
  return acc + obj["value"];
}, 0);

var todayWindmillSum = todayData.filter(obj => obj.type == "Wind").reduce(function
(acc, obj) {
  return acc + obj["value"];
}, 0);

var todayThermalSum = todayData.filter(obj => obj.type == "Thermal").reduce(function
(acc, obj) {
  return acc + obj["value"];
}, 0);
```

```
            pieChart1.data.datasets[0].data = [todaySolarSum, todayWindmillSum,
todayThermalSum];
            pieChart1.update();

            var oddOrEven = Math.floor(Math.random() * 3);

            presetDate.push(new Date().toLocaleTimeString());
            powerFactorData.push(oddOrEven % 3 == 1 ? (parseFloat(powerFactorInput) - 0.5 + 1) :
(oddOrEven % 3 == 2 ? (parseFloat(powerFactorInput) - 0.5) : parseFloat(powerFactorInput)));

            powerFactorChart.data.datasets[0].data = powerFactorData.slice(-25);
            powerFactorChart.data.labels = presetDate;
            powerFactorChart.update();

            oddOrEven = Math.floor(Math.random() * 3);

            lossData.push(oddOrEven % 3 == 1 ? (parseInt(lossInput) + 0.5) : (oddOrEven % 3 == 2 ?
(parseInt(lossInput) - 0.5) : parseInt(lossInput)));

            lossChart.data.datasets[0].data = lossData.slice(-25);
            lossChart.data.labels = presetDate;
            lossChart.update();

            if (parseInt(lossData.slice(-1)[0]) > 30){
               showToast('Need manual support: LOSS');
            }

            if (loadPowerInput > 15 || loadPowerInput < 7.5) {
               showToast('Need manual support: LOAD');
            }
         },
         error: function (xhr, textStatus, errorThrown) {
            console.log('Error:', errorThrown);
         }
      });
   }

   function reload(data) {
      var powerFactorChartCtx = document.getElementById('powerFactorChart').getContext('2d');
      powerFactorChart = new Chart(powerFactorChartCtx, {
         type: 'line',
         data: {
            labels: presetDate,
            datasets: [{
               label: 'Today Power Factor',
               data: powerFactorData,
               backgroundColor: 'rgba(255, 99, 132, 0.2)',
               borderColor: 'rgba(255, 99, 132, 1)',
               borderWidth: 1
            }]
         },
         options: {
```

```
            scales: {
                yAxes: [{
                    ticks: {
                        beginAtZero: true
                    }
                }]
            }
        }
    });

    var lossChartCtx = document.getElementById('lossChart').getContext('2d');
    lossChart = new Chart(lossChartCtx, {
        type: 'line',
        data: {
            labels: presetDate,
            datasets: [{
                label: 'Today Loss',
                data: lossData,
                backgroundColor: 'rgba(255, 99, 132, 0.2)',
                borderColor: 'rgba(255, 99, 132, 1)',
                borderWidth: 1
            }]
        },
        options: {
            scales: {
                yAxes: [{
                    ticks: {
                        beginAtZero: true
                    }
                }]
            }
        }
    });

    var todayData = data.filter(obj => new Date(obj.createdAt) > today);
    var yesterdayData = data.filter(obj => new Date(obj.createdAt) < today && new
Date(obj.createdAt) > yesterday);

    var todaySolar = todayData.filter(obj => obj.type == "Solar").slice(-25).map(function (obj) {
        return obj["value"];
    });

    var yesterdaySolar = yesterdayData.filter(obj => obj.type == "Solar").map(function (obj) {
        return obj["value"];
    });

    var todaySolarLable = todayData.filter(obj => obj.type == "Solar").slice(-25).map(function
(obj) {
        return new Date(obj["createdAt"]).toLocaleTimeString();
    });

    var lineChartCtx = document.getElementById('lineChart').getContext('2d');
    lineChart = new Chart(lineChartCtx, {
```

```
            type: 'line',
            data: {
                labels: todaySolarLable,
                datasets: [{
                    label: 'Today Solar Plant',
                    data: todaySolar,
                    backgroundColor: 'rgba(255, 99, 132, 0.2)',
                    borderColor: 'rgba(255, 99, 132, 1)',
                    borderWidth: 1
                }, {
                    label: 'Yesterday Solar Plant',
                    data: yesterdaySolar,
                    backgroundColor: 'rgba(54, 162, 235, 0.2)',
                    borderColor: 'rgba(54, 162, 235, 1)',
                    borderWidth: 1
                }]
            },
            options: {
                scales: {
                    yAxes: [{
                        ticks: {
                            beginAtZero: true
                        }
                    }]
                }
            }
        });

        var todayWindmill = todayData.filter(obj => obj.type == "Wind").slice(-25).map(function (obj)
{
            return obj["value"];
        });

        var yesterdayWindmill = yesterdayData.filter(obj => obj.type == "Wind").map(function (obj) {
            return obj["value"];
        });

        var todayWindmillLable = todayData.filter(obj => obj.type == "Wind").slice(-25).map(function
(obj) {
            return new Date(obj["createdAt"]).toLocaleTimeString();
        });

        var barChartCtx = document.getElementById('barChart').getContext('2d');
        barChart = new Chart(barChartCtx, {
            type: 'bar',
            data: {
                labels: todayWindmillLable,
                datasets: [{
                    label: 'Today Wind Plant',
                    data: todayWindmill,
                    backgroundColor: 'rgba(255, 99, 132, 0.2)',
                    borderColor: 'rgba(255, 99, 132, 1)',
                    borderWidth: 1
```

```
            }, {
                label: 'Yesterday Wind Plant',
                data: yesterdayWindmill,
                backgroundColor: 'rgba(54, 162, 235, 0.2)',
                borderColor: 'rgba(54, 162, 235, 1)',
                borderWidth: 1
            }]
        },
        options: {
            scales: {
                yAxes: [{
                    ticks: {
                        beginAtZero: true
                    }
                }]
            }
        }
    });

    var todayThermal = todayData.filter(obj => obj.type == "Thermal").slice(-25).map(function
(obj) {
        return obj["value"];
    });

    var yesterdayThermal = yesterdayData.filter(obj => obj.type == "Thermal").map(function (obj)
{
        return obj["value"];
    });

    var todayThermalLable = todayData.filter(obj => obj.type == "Thermal").slice(-
25).map(function (obj) {
        return new Date(obj["createdAt"]).toLocaleTimeString();
    });

    var lineChart1Ctx = document.getElementById('lineChart1').getContext('2d');
    lineChart1 = new Chart(lineChart1Ctx, {
        type: 'line',
        data: {
            labels: todayThermalLable,
            datasets: [{
                label: 'Today Thermal Plant',
                data: todayThermal,
                backgroundColor: 'rgba(255, 99, 132, 0.2)',
                borderColor: 'rgba(255, 99, 132, 1)',
                borderWidth: 1
            }, {
                label: 'Yesterday Thermal Plant',
                data: yesterdayThermal,
                backgroundColor: 'rgba(54, 162, 235, 0.2)',
                borderColor: 'rgba(54, 162, 235, 1)',
                borderWidth: 1
            }]
        },
```

```
            options: {
                scales: {
                    yAxes: [{
                        ticks: {
                            beginAtZero: true
                        }
                    }]
                }
            }
        });

        var todaySolarSum = todayData.filter(obj => obj.type == "Solar").reduce(function (acc, obj) {
            return acc + obj["value"];
        }, 0);

        var yesterdaySolarSum = yesterdayData.filter(obj => obj.type == "Solar").reduce(function (acc,
obj) {
            return acc + obj["value"];
        }, 0);

        var todayWindmillSum = todayData.filter(obj => obj.type == "Wind").reduce(function (acc,
obj) {
            return acc + obj["value"];
        }, 0);

        var yesterdayWindmillSum = yesterdayData.filter(obj => obj.type == "Wind").reduce(function
(acc, obj) {
            return acc + obj["value"];
        }, 0);

        var todayThermalSum = todayData.filter(obj => obj.type == "Thermal").reduce(function (acc,
obj) {
            return acc + obj["value"];
        }, 0);

        var yesterdayThermalSum = yesterdayData.filter(obj => obj.type ==
"Thermal").reduce(function (acc, obj) {
            return acc + obj["value"];
        }, 0);

        var pieChart1Ctx = document.getElementById('pieChart1').getContext('2d');
        pieChart1 = new Chart(pieChart1Ctx, {
            type: 'pie',
            data: {
                labels: ['Today Solar Plant', 'Today Wind Plant', 'Today Thermal Plant'],
                datasets: [{
                    label: '# of Volts',
                    data: [todaySolarSum, todayWindmillSum, todayThermalSum],
                    backgroundColor: [
                        'rgba(255, 99, 132, 0.5)',
                        'rgba(54, 162, 235, 0.5)',
                        'rgba(255, 206, 86, 0.5)',
                    ],
```

```
            borderColor: [
                'rgba(255, 99, 132, 1)',
                'rgba(54, 162, 235, 1)',
                'rgba(255, 206, 86, 1)',
            ],
            borderWidth: 1
        }]
    },
    options: {
        scales: {
            yAxes: [{
                ticks: {
                    beginAtZero: true
                }
            }]
        }
    }
});

var pieChartCtx = document.getElementById('pieChart').getContext('2d');
pieChart = new Chart(pieChartCtx, {
    type: 'pie',
    data: {
        labels: ['Yesterday Solar Plant', 'Yesterday Wind Plant', 'Yesterday Thermal Plant'],
        datasets: [{
            label: '# of Volts',
            data: [yesterdaySolarSum, yesterdayWindmillSum, yesterdayThermalSum],
            backgroundColor: [
                'rgba(255, 99, 132, 0.5)',
                'rgba(54, 162, 235, 0.5)',
                'rgba(255, 206, 86, 0.5)',
            ],
            borderColor: [
                'rgba(255, 99, 132, 1)',
                'rgba(54, 162, 235, 1)',
                'rgba(255, 206, 86, 1)',
            ],
            borderWidth: 1
        }]
    },
    options: {
        scales: {
            yAxes: [{
                ticks: {
                    beginAtZero: true
                }
            }]
        }
    }
});
}

$(document).ready(function () {
```

```
        reload(model);
        setInterval(runAIPoweredSimulator, 3000);
    });

    function updateLoadInput() {
        solarMinLoad = $('#solarLoadMinInput').val() == '' ? 0.01 : $('#solarLoadMinInput').val();
        windMinLoad = $('#windLoadMinInput').val() == '' ? 0.01 : $('#windLoadMinInput').val();
        thermalMinLoad = $('#thermalLoadMinInput').val() == '' ? 0.01 :
$('#thermalLoadMinInput').val();

        solarMaxLoad = $('#solarLoadMaxInput').val() == '' ? 0.01 : $('#solarLoadMaxInput').val();
        windMaxLoad = $('#windLoadMaxInput').val() == '' ? 0.01 : $('#windLoadMaxInput').val();
        thermalMaxLoad = $('#thermalLoadMaxInput').val() == '' ? 0.01 :
$('#thermalLoadMaxInput').val();

        lossInput = $('#lossInput').val() == '' ? 0 : $('#lossInput').val();
        powerFactorInput = $('#powerFactorInput').val() == '' ? 0.01 : $('#powerFactorInput').val();
        loadPowerInput = $('#loadPowerInput').val() == '' ? 15 : $('#loadPowerInput').val();

        showToast('Updated Successfully');
    }

    function validateInput(inputId) {
        var input = document.getElementById(inputId);

        if (inputId == 'lossInput') {
            if (input.value > 100) {
                input.value = 100;
            }

            if (input.value < 0) {
                input.value = 0;
            }
        } else if (inputId == 'loadPowerInput'){
        } else {
            if (input.value > 0.99) {
                input.value = 0.99;
            }

            if (input.value < 0.01) {
                input.value = 0.01;
            }
        }
    }

    function loadTypeChanged() {
        var loadType = document.getElementById("loadType").value;
        switch (loadType) {
            case "summer":
                $('#solarLoadMinInput').val(0.80);
                $('#windLoadMinInput').val(0.60);
                $('#thermalLoadMinInput').val(0.20);
```

```
            $('#solarLoadMaxInput').val(0.90);
            $('#windLoadMaxInput').val(0.70);
            $('#thermalLoadMaxInput').val(0.60);

            break;
        case "winter":
            $('#solarLoadMinInput').val(0.40);
            $('#windLoadMinInput').val(0.80);
            $('#thermalLoadMinInput').val(0.60);

            $('#solarLoadMaxInput').val(0.50);
            $('#windLoadMaxInput').val(0.90);
            $('#thermalLoadMaxInput').val(0.80);

            break;
        case "spring":
            $('#solarLoadMinInput').val(0.70);
            $('#windLoadMinInput').val(0.70);
            $('#thermalLoadMinInput').val(0.35);

            $('#solarLoadMaxInput').val(0.85);
            $('#windLoadMaxInput').val(0.80);
            $('#thermalLoadMaxInput').val(0.60);

            break;
        case "rainy":
            $('#solarLoadMinInput').val(0.50);
            $('#windLoadMinInput').val(0.80);
            $('#thermalLoadMinInput').val(0.50);

            $('#solarLoadMaxInput').val(0.60);
            $('#windLoadMaxInput').val(0.90);
            $('#thermalLoadMaxInput').val(0.70);

            break;
        default:
            break;
    }
}

function showToast(msgContent) {
    var toast = document.createElement('div');
    toast.classList.add('toast');
    toast.classList.add('text-center');
    toast.textContent = msgContent;

    document.getElementById('toast-container').appendChild(toast);

    setTimeout(function () {
        toast.classList.add('show');
    }, 100);

    setTimeout(function () {
```

```
            toast.classList.remove('show');
            setTimeout(function () {
                toast.remove();
            }, 300);
        }, 3000);
    }
</script>
}
```

CHAPTER 2
LITERATURE SURVEY

Conducting load flow analysis is essential for determining the steady-state operating parameters of a power system, including node voltages, line currents, angles, and power losses under specific load conditions. Power utilities regularly analyze these variables to prepare for potential critical scenarios such as system failures or fault occurrences. Additionally, load flow analysis aids in the planning and expansion of existing power systems. The integration of distributed energy resources (DERs) like electric vehicles (EVs), wind, solar, distributed generators (DGs), and storage introduces significant challenges to the distribution system, including variability, uncertainty in control, infrastructure, and grid operation. Therefore, performing power flow analysis becomes imperative to understand the impacts of integrating various entities into the grid.

Illustrates the interconnectedness between different approaches of artificial intelligence (AI) and data science. Machine learning (ML) is a subset of AI where machines learn objectives from historical data, enabling them to act on new data patterns without explicit programming instructions. Deep learning (DL), another subset of ML, offers higher accuracy, leveraging larger datasets and more complex neural networks. ML finds numerous applications in power system data analytics, including load forecasting, renewable energy forecasting, state estimation, fault detection, grid security analysis, and power flow studies. Recent advancements in ML, particularly deep learning, play a significant role in addressing the challenges posed by the increasing uncertainty of distributed energy resources (DERs).

One crucial application is power flow analysis, necessary for analyzing the steady-state condition of the network. Various machine learning methodologies have been applied to perform load flow analysis, including support vector regression (SVR), neural networks (NN), and support vector machines (SVM), among others.

Deep neural networks (DNN) or deep learning (DL) represent another approach in machine learning, particularly effective in image recognition, prediction, and audio recognition. With the advancement of smart grids, the interconnection of smart grid and green buildings introduces bidirectional power flow, necessitating accurate energy consumption prediction over extended periods. Supervised and unsupervised energy consumption prediction models can assist building owners in planning energy consumption effectively.

Deep learning has emerged as a critical tool in addressing data-driven challenges in power systems. However, ensuring network security from data attacks remains a primary concern. Deep learning techniques, while powerful, are susceptible to attacks on input data, potentially compromising model performance. Power system restoration and fault diagnosis are areas where deep learning has been extensively applied, leveraging data collected from SCADA measurements for improved fault localization and state estimation.

In this work, we propose utilizing deep neural network architectures like Convolutional Neural Networks (CNN) to predict power flow accurately, replacing traditional iteration-based load flow techniques.

Our trained neural networks can predict branch currents, node voltages, angles, and power losses with high accuracy. Leveraging models such as Radial Basis Function (RBF), Multi-Layer Perceptron (MLP), and CNN, we aim to enhance load flow prediction accuracy and efficiency, particularly in large and unbalanced distribution systems. While deep learning offers significant advantages, concerns regarding network security and data attacks necessitate careful consideration and mitigation strategies.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Existing systems for managing power plants often rely on manual or rule-based approaches for phase balancing, leading to suboptimal performance, inefficiencies in power distribution and also unbalancing of power grid.

These systems lack the capability to adapt to dynamic changes in demand and generation, resulting in imbalances and potential grid instability.

DISADVANTAGE

- Manual or rule-based approaches lack adaptability to dynamic conditions.
- Lack of predictive capabilities to anticipate grid instability.
- Inefficiencies in power distribution due to static control mechanisms.
- Limited ability to optimize phase balancing in real-time.

3.2 PROPOSED SYSTEM

- The proposed system introduces an AI-powered dashboard for managing power plants within the smart grid.
- By utilizing advanced machine learning algorithms, the system dynamically optimizes phase balancing based on real-time data, demand forecasts, and grid conditions.
- This enables proactive management of power generation and distribution, leading to improved efficiency, stability, and reliability of the smart grid.

3.2.1 ADVANTAGES

- The proposed system has a broad scope, applicable to utility companies, grid operators, and energy management entities involved in smart grid deployments.
- It can be integrated into existing grid management systems or deployed as a standalone solution for optimizing phase balancing and improving grid performance.

3.3 SYSTEM STUDY

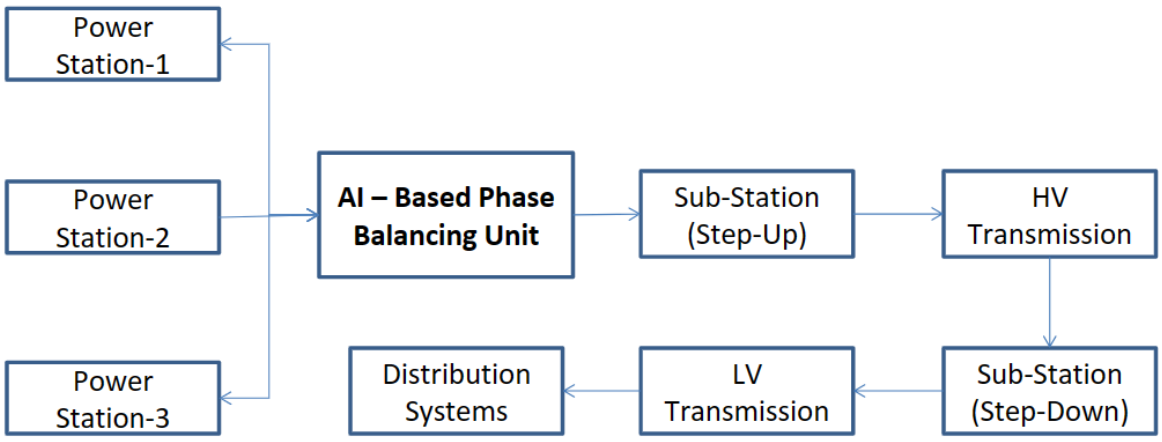


Figure 5.System Architecture.

- Real-time data from power plants and grid sensors are collected and processed by AI algorithms to optimize phase balancing.
- The dashboard provides visualization of grid performance metrics and enables manual intervention when necessary.

3.3.1 STEP-UP SUBSTATION

This facility elevates the voltage of electricity for efficient transmission over long distances. It plays a crucial role in the initial stages of the power distribution process, where electricity generated at various sources is collected and transformed to higher voltage levels for transmission through the high-voltage (HV) transmission lines.

3.3.2 STEP-DOWN SUBSTATION

Serving as an intermediary between the transmission and distribution systems, the step-down substation reduces the voltage of electricity received from the transmission lines to levels suitable for distribution to end-users. It ensures that electricity is delivered safely and efficiently to homes, businesses, and other consumers.

3.3.3 HV TRANSMISSION

High-voltage (HV) transmission lines are the backbone of the power grid, carrying electricity over long distances from power plants or step-up substations to various regions and cities. These lines operate at high voltages to minimize energy losses during transmission and ensure reliable delivery of electricity.

3.3.4 LV TRANSMISSION

Low-voltage (LV) transmission lines distribute electricity from step-down substations to local distribution networks, connecting to residential, commercial, and industrial areas. LV transmission lines operate at lower voltages compared to HV lines and are responsible for delivering electricity to end-users within a specific area.

3.3.5 POWER GRID

The power grid comprises interconnected networks of transmission lines, substations, and distribution systems that facilitate the generation, transmission, and distribution of electricity. It ensures the reliable and efficient delivery of electricity from power plants to consumers across various regions.

3.3.6 DISTRIBUTION SYSTEM

The distribution system delivers electricity from the transmission and substation facilities to end-users, including homes, businesses, and industries. It consists of distribution lines, transformers,

meters, and other equipment designed to regulate voltage levels and manage the flow of electricity within local communities.

3.4 SYSTEM REQUIREMENTS

3.4.1 HARDWARE REQUIREMENTS:-

Server infrastructure for hosting the dashboard, communication modules for data exchange.

3.4.2 SOFTWARE REQUIREMENTS:-

AI and machine learning frameworks (e.g., TensorFlow, PyTorch), programming languages (e.g., Python), data visualization tools (e.g., Tableau), database management systems.

CHAPTER 4

SYSTEM DESIGN

4.1INPUT DESIGN

LOAD FORECASTING

The escalating integration of renewable energy sources like solar, wind, and tidal power into the modern power grid is intensifying the challenges of scheduling and operating the smart grid. Load forecasting (LF), a pivotal component in maintaining the stability and intelligence of the power system, assumes critical importance for modern power system planning and operation. LF plays a vital role in predicting electricity demand accurately, thus aiding in cost reduction and power conservation. However, forecasting becomes exceedingly complex when dealing with non-stationary loads.

LF can be categorized into three levels based on the duration of the forecast: short-term LF (STLF), mid-term LF (MTLF), and long-term LF (LTLF). STLF predicts load variations from minutes to hours, while MTLF and LTLF extend the forecast period to weeks and years, respectively. Furthermore, LF is influenced by various factors such as weather conditions, time of day, season, events, customer types, and academic schedules.

SHORT-TERM LOAD FORECASTING

Several approaches have been proposed to enhance the accuracy of STLF. For instance, a hybrid incremental learning approach integrating discrete wavelet transform, empirical mode decomposition, and random vector functional link network has shown promising results. Additionally, deep learning-based methods, such as deep neural networks (DNNs), have been effective in capturing patterns from historical data. However, challenges like overfitting and computational overhead persist.

MID-TERM LOAD FORECASTING

While STLF addresses short-term fluctuations, MTLF and LTLF are essential for long-term planning and system dynamics understanding. These forecasts are influenced not only by historical load data but also by demographic factors and weather conditions. Various models, including dynamic Bayes networks (DBNs) and deep learning algorithms, have been proposed for MTLF, offering improved accuracy and efficiency.

LONG-TERM LOAD FORECASTING

LTLF plays a crucial role in system planning and expansion. ML and AI techniques like multivariate adaptive regression splines (MARS), long short-term memory (LSTM) networks, and hybrid models have shown promising results in predicting long-term load demand accurately. These models leverage historical data and environmental variables to forecast power consumption over extended periods.

POWER GRID STABILITY ASSESSMENT

Assessing power grid stability, including transient stability, frequency stability, small-signal stability, and voltage stability, is vital for ensuring grid reliability. Traditional methods rely heavily on real-time dynamic power system models, but AI-based approaches have emerged as viable alternatives. ML algorithms such as decision trees, support vector machines (SVMs), and artificial neural networks (ANNs) offer efficient solutions for stability assessment, leveraging data collected from phasor measurement units (PMUs) and wide-area measurement systems (WAMS).

TRANSIENT STABILITY ASSESSMENT

Determining whether a system will maintain synchronization after a significant perturbation is the essence of transient stability assessment (TSA). Traditional methods like time domain simulations and direct methods face challenges in making reliable decisions due to the increasing complexity of power systems. However, the emergence of AI technologies offers new perspectives by leveraging the vast amount of data collected by Phasor Measurement Units (PMUs) and Wide-Area Measurement Systems (WAMS).

Various machine learning (ML) algorithms, including decision trees, support vector machines (SVMs), and artificial neural networks (ANNs), have been compared for online TSA, exhibiting similar performance with variations based on dataset quality. Additionally, innovative approaches such as deep neural classifiers, trajectory fitting combined with Extreme Learning Machines (ELM), and Recurrent Neural Network-Long Short-Term Memory (RNN-LSTM) models have shown effectiveness and reliability in TSA.

FREQUENCY STABILITY ASSESSMENT

Power grid frequency stability assessments (FSAs) focus on the system's ability to maintain a stable frequency range following severe perturbations causing an imbalance between generation and load. AI technologies have been employed to develop hybrid models integrating frequency response models with extreme learning ML models for FSA, showing promising results in maintaining system stability.

SMALL-SIGNAL STABILITY ASSESSMENT

Small-signal stability, synonymous with oscillatory stability assessment (OSA), pertains to the system's ability to maintain synchronism under minor disturbances. ML-based methods, including Convolutional Neural Network (CNN) and multivariate random forest regression (MRFR) algorithms, have demonstrated robustness and accuracy in OSA, even as the system scales up.

VOLTAGE STABILITY ASSESSMENT

Voltage collapse poses a significant threat to power system stability, necessitating timely voltage stability assessments (VSAs) for prevention. AI-based models, such as artificial neural networks (ANNs), support vector machines (SVMs), and decision trees, offer effective solutions for VSA, achieving low misclassification rates and providing insights into voltage magnitude changes in real-time scenarios. Feature selection models and moment-based spectrum estimation methods further enhance the accuracy and efficiency of voltage stability assessment.

FAULTS DETECTION

An Extreme Learning Machine (ELM)-based method was employed for fault location detection in the system, utilizing features extracted through wavelet transform (WT). A comparison was made with Support Vector Regression (SVR) and Artificial Neural Network (ANN) models. Additionally, a Generalized Likelihood Ratio Test (GLRT) based on Gaussian Process Regression (GPR) was presented to enhance fault detection (FD) performance in photovoltaic (PV) systems. Two ensemble methods, combining supervised and unsupervised classifiers, were utilized to detect stealthy false data injections.

An ensemble framework integrating five machine learning (ML) algorithms was built for analyzing power grid frequency disturbances, capable of detecting faults with varying degrees of severity. High-Impedance Fault Detection (HIFD) in power systems was addressed with an ANN-based method, achieving high accuracy (98.67%).

ELM was also applied to HIFD, typically based on wavelet packet transform. A method for predicting line trip faults in power systems was proposed, employing Long Short-Term Memory (LSTM) networks and Support Vector Machines (SVM). ML-based techniques, including discrete wavelet transform and double-channel extreme learning machines, were suggested for fault localization and classification in transmission lines. Furthermore, a stacked sparse autoencoder-based network combined with SVM and Principal Component Analysis (PCA) was proposed to improve the accuracy of line trip fault prediction using real-world data.

Despite the development of microgrids as effective power solutions for integrating renewable sources, fault detection (FD) in microgrids remains a challenge. A hybrid approach combining S-transform and feedforward neural networks was utilized for distribution grid FD. ANN-based methods were also evaluated, demonstrating effectiveness in detecting fault time and location. To address labeled and unlabeled data, a semisupervised ML model comprising a K-Nearest Neighbors (KNN) model and a decision tree model was proposed for FD in transmission and distribution microgrid systems. Additionally, SVM-based algorithms were developed to solve islanding and grid FD problems, exhibiting superior performance compared to traditional methods.

In 2017, a Probabilistic Neural Network (PNN) classifier was employed for FD and fault diagnosis in the DC side of a PV system, while in 2020, an ANN-based fault detection algorithm for PV achieved 97% overall accuracy. Evaluation of deep ANNs in wind turbine FD highlighted their effectiveness, and the application of ensemble methods for energy theft detection was presented.

CHAPTER 5

CONCLUSION

The traditional electric grid system is evolving into a smart grid system, presenting challenges for conventional power system methods to process and analyze the vast amounts of data inherent to a smart grid. Consequently, AI techniques are being developed and deployed across various applications in smart grid systems, showing promising results. This paper offers a survey of recent AI applications in four critical areas—load forecasting, power grid stability assessment, fault detection, and security issues—not previously explored in existing studies. It also addresses current challenges, opportunities, and the future prospects of employing AI techniques to realize a truly smart grid.

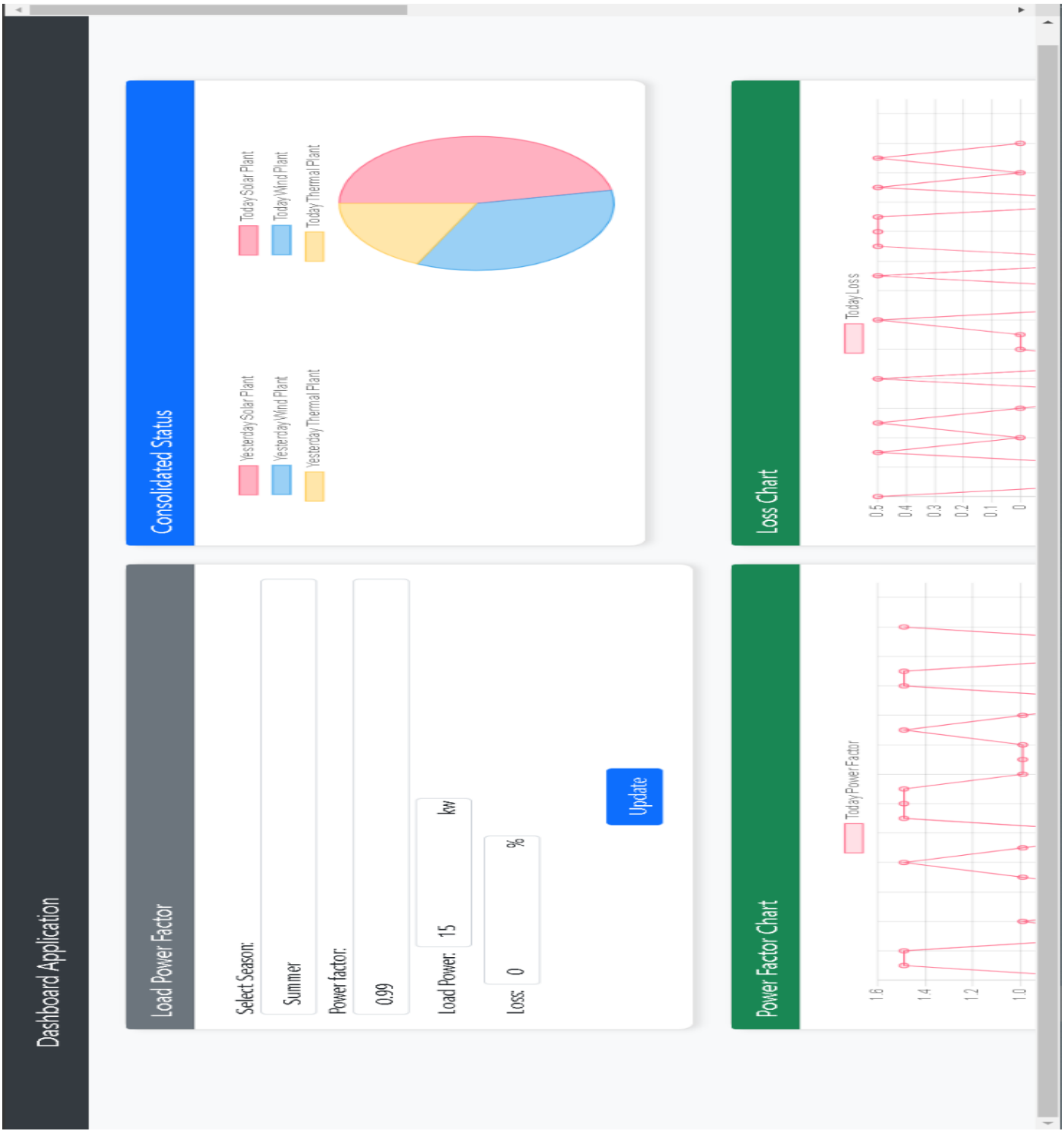
Based on this survey, our conclusions can be summarized as follows: (i) AI techniques have been successfully applied to several critical areas crucial for enhancing the reliability and resilience of smart grids; (ii) However, challenges remain, notably in data privacy, security, and addressing the "black box"

nature of certain AI techniques to ensure a human-centered approach to AI solution design; and (iii) This survey aims to stimulate discussions in the surveyed application areas, fostering the exchange of ideas to further strengthen smart grid development. In essence, the utilization of AI techniques is bolstering efforts to enhance the reliability and resilience of smart grid systems.

Our future research in this domain will focus on exploring the implications of the "black box" nature of AI techniques on smart grid operations. Specifically, we will investigate how smart grid operators have addressed this challenge, aiming to inform the design of more human-centered AI solutions.

CHAPTER 6

OUTPUT:




Wind Chart
Today Wind Plant
Yesterday Wind Plant
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
10:47:46 PM
10:47:40 PM
10:47:34 PM
10:47:29 PM
10:47:22 PM
10:47:16 PM
10:47:10 PM
10:47:04 PM
10:46:59 PM
10:46:53 PM
10:46:47 PM
10:46:41 PM
10:46:35 PM
Solar Chart
Today Solar Plant
Yesterday Solar Plant
0.820
0.818
0.816
0.814
0.812
0.810
0.808
0.806
0.804
0.802
0.800
10:47:46 PM
10:47:40 PM
10:47:34 PM
10:47:28 PM
10:47:22 PM
10:47:16 PM
10:47:10 PM
10:47:04 PM
10:46:58 PM
10:46:52 PM
10:46:46 PM
10:46:40 PM
10:46:34 PM
Thermal Chart
Today Thermal Plant
Yesterday Thermal Plant
0.60
0.55
0.50
0.45
0.40
0.35
0.30