Energy management System using Multi Agent Systems with IoT and Machine learning

Leo Raju, Vishwa Raj V, Vishal E and Vimalan K M, 
Department of Electrical and Electronics Engineering 
SSN College of Engineering 
Chennai, India 
leor@ssn.edu.in, vishwarajv14@gmail.com, vishal.easwaramoorthy@gmail.com, vimalan.mani@gmail.com

Abstract—The topic of this study is a Smart Microgrid with Dynamic, Distributed Energy Management. At the moment, a SCADA system is used for energy management (Supervisory Control and Data Acquisition). A distributed energy system (DES) called a "multi-agent system" (MAS) is made up of several software agents that form a loosely connected network and cooperate to solve issues that are more complex than they can handle alone. By selecting the least expensive renewable resource that is readily available at the time, economic and environmental optimization is performed. As a result, dynamic energy management has been put into practice. The purpose of this research is to demonstrate how an energy management system based on MAS may be utilized to optimize a microgrid. Machine learning has been used to forecast load, increase power grid efficiency, and establish a stable operating environment.

Keywords—Arduino; Multi Agent System; Internet of Things; Machine Learning; Radial Basis Function; Random Forest Regressor; Dynamic Energy Management system; Agent oriented programming; Energy management system; Java Agent DEvelopment Framework

I. INTRODUCTION

When compared to the power that is used by customers, the initial power created is substantially more. This is due to the fact that most of the power generated is lost as heat-producing discharges. As a result, there is an imbalance in the ratio of demand to consumption. Since there will frequently be blackouts and power interruptions in the existing systems, we are unable to continuously monitor the energy usage. These difficulties are what drives the requirement for an effective power management system. In order to assure efficiency in the system's storage and transmission components, we would have addressed managing the demand for power in our paper using smart grid and micro grid technologies. The data is continuously collected and stored in the cloud using IoT concepts for additional analysis.

The evolution of smart grids as the new generation of electric power system and the introduction of agent-based energy management system is discussed in [1]. The entire architecture of JADE (Java Agent DEvelopment Framework) and the available functions and classes of primitive agents are discussed in this book [2]. These classes and functions are widely being used in the development of MAS for Smart grids. The usage of agents to develop an energy system for the Smart grids is briefed in [3]. The primary objective of sustainable development is to increase the proportion of renewable energy in the world's total energy consumption [4]. In order to integrate the production of renewable energy sources alongside those from conventional energy sources, microgrids are the most crucial component of a power supply network. For microgrids to maintain their power's dependability, quality, and affordability, they must be connected to larger networks [5]. Analysis of various machine learning algorithms for classifying the smart grid system as stable or unstable is discussed in [6]. Every agent in MAS has its own local authority and responsibility. Additionally, these agents can work together to maintain the entire grid network. This characteristic of MAS allows it to continue to function even in the event of network failure and malfunction [7]. MAS and cloud systems can be combined, and it can manage IoT features [8]. A working simulation for the hierarchical control of smart grid with various other sources such as wind, solar and battery has been introduced from [9]. The transactive organization of the power system is briefed in [10]. The software simulation of this model is discussed in [11]. Use of different Machine learning methods to predict the load value beforehand is discussed in [12]. Understanding of the use of control system to arrive at better results for the smart grid-oriented system has been arrived from [13].

By using machine learning algorithms to predict the short-term load, we have also optimized the current power grid system and by allowing for a smooth contingency analysis process. By averaging consumption from day to night, this prediction offers increased efficiency, increases the production profit, and avoid peak consumption periods.

Organization of the paper is elucidated below. The key technical concepts (multi-Agent energy management Systems and Internet of Things) are covered in Section II. The additional ML feature is covered in Section III. The action plan is presented in Section IV. Prototype simulation is presented in Section V. Machine learning models are discussed in Section VI.
II. MULTI AGENT SYSTEM AND IoT BASED ENERGY MANAGEMENT IN MICROGRID

Building an IoT-based energy management system requires a number of different components. Here is a section where they are explained.

A. Microgrid:

The foundation of a smart grid is a microgrid. A smart grid can be created by multiplying these grids by a large number. The microgrids are connected to one another and together they form a network that provides continuous power supply. It can seamlessly switch between the various supplies and provide constant energy when connected with both a traditional power source and a renewable energy source. The power can be rerouted through other microgrids even if one grid is undergoing maintenance. It is possible to prevent the power supply interruption.

B. Smart grid:

A smart grid is a system of power plants, substations, transmission lines, transformers, and other devices that is in charge of distributing electricity to businesses and residences. It serves as a middleman between power generation and consumption, taking care of any power outages and network failures. This smart grid, which uses both conventional and unconventional power sources, manages all the microgrids in a given area. By lowering the cost of production, it thus addresses overloading, peak demand, and equally distributes the available energy to the consumers.

C. Internet of Things (IoT):

IoT is the process of gathering data online, interacting with it, and manipulating it to produce the desired result. We use an IoT platform and a multi-agent system in this project. The concepts presented in this paper are implemented using JADE (Java Agent DEvelopment Framework). For the purposes of online storage and data communication, a cloud data base (like UBIDOTS) is utilized.

D. Java Agent DEvelopment Framework:

JADE is an API (application programming interface) that enables allowing programming of the necessary agents, also acquire the information that need to be processed to determine variables like peak load time, cost-effective distribution, etc. JADE is a Java-based software framework that is FIPA (Foundation for Intelligent Physical Agents) compliant and used to create multi-agent systems. Through a middle-ware that complies with the FIPA specifications and a set of graphical tools, it makes it easier to implement multi-agent systems. It aids in the phases of debugging. JADE can offer users a practical distributed platform so they can concentrate on creating agents for controlling and monitoring power balance while operating microgrids.

E. Ubidots:

A cloud service that allows users of any internet-connected device to send or receive data. Here, all the information gathered from users and the smart grid is stored and used by the JADE environment to discover different effective ways to distribute power. Remote servers are available for data handling on this IoT platform. Real-time data can be used to configure actions and alerts, and visual tools can be used to maximize the value of your data.

F. Arduino:

The smallest component of a grid network that manages sensors and measures load is called an Arduino. Its Wi-Fi module facilitates communication with the data server and transmits the data that has been gathered to the cloud. It permits seamless plug-and-play features so that any new production source (like solar) can be plugged in or disconnected from the grid without having an immediate impact on the power supply network. It is a processing component of the entire smart grid network.
G. Distribution Energy System:

Distributed Energy Systems (DES) describes the entire power system, i.e., from generation to control strategies. These systems are can be tailored to needs of the application that the user intends to create efficiency, and security of supply and carbon reduction.

III. MACHINE LEARNING IN SMART GRID

The potential of computers are ever increasing with not a plateau in sight anytime soon. This is being exploited in modernization of many age old, robust systems like the current power system, with the advent of the Smart grid system where as discussed in the previous sections [5]. Due to the highly intermittent nature of the renewable power sources the use of ML algorithms is even more appreciated and direly required. The ability to swallow humongous volumes of data and based on appropriate training spit out an immensely accurate model of what the required price of the load should be at a particular time of the day is of unimaginable potential use. Precision in the field of planning in terms of resources and production of energy can be achieved using this technology. Creating a ML model requires huge amounts of data, as training is proportional to accuracy which is proportional to the amount of data available. Features which support the prediction of the required data have to be highly correlative and non-disruptive in nature.

A. Radial Basis Function

Radial basis even though being extremely complex possess a sense of intuition, by performing regression based on similarities on between the various datapoints available in the datasets [5]. To break it down simply, each value is stored in a neuron and spaced based on magnitude of value almost on a cartesian plane in case of a 2-D system, and the Euclidian distance between the points is measured in order to calculate a measure of similarity between points to create a correlation, based on which sorting is performed. The intuition comes into play here as the namesake is because of the calculation of the ‘Radial’ distance between data to achieve predictions. This model is prone to overfitting.

B. Random Forest Method

The model represents the naming convention used, where regression is achieved using combination of multiple single decision trees which are trained for a different subset of datasets, which gives a pseudo-continuous prediction for an otherwise limited model [10]. The underlying issue of over-fitting is solved using the powerful Stochastic discrimination tool of the regression concept. Each tree has leaves and nodes, nodes represent a condition of greater than or less than a value, and the leaves move onto other conditions so on and so forth until the final layer of leaves is achieved to narrow down the exact value to be predicted. Values of multiple trees are aggregated in a voting model and the final answer is achieved based on majority prediction.

IV. ACTION PLAN

- The proposed energy management system is implemented using the library called JADE, and utilizing an Arduino mega microcontroller and serial communications.
- The actions of the agents (such as load, solar, and wind agents, among others) are practically validated.
- A testbed is created by attaching potentiometers and LEDs to the Arduino board. Potentiometers reflect the micro-environmental grid's factors, such as its loads, batteries, grid, and diesel generators, while LEDs show the actuator signals. After receiving environmental values from an Arduino board, JADE agents collaborate to make the tactical choices depicted in Fig. 4.
- The LEDs serve as indicators for the command signals that the agents ultimately send to the Arduino board, which can then be utilized to activate or power the microgrid actuators in the field. Three micro-grids are connected to create a smart micro-grid testbed, and the features of the smart grid are validated in a dynamic environment (Fig. 5).
- The development of a transactive energy market allows for the cost-effective purchase or sale of power across the network by anyone. Future smart grids could be formed by scaling this technique.
- Through Ubidot, Node-MCU (an open source WiFi module) is utilized to connect to the internet (TCP/IP). It essentially makes it possible for our...
Arduino Board to send data to or receive data from the Internet.

Fig. 4. Multi Agent System Block Diagram

Fig. 5. Structure of the entire MAS

V. IMPLEMENTATION OF MAS

A. Creating Agents in Jade Framework

The Java IDE can be expanded with the JADE library (jade.jar). The MAS is built on a microgrid-based paradigm, with each microgrid consisting of agents with load and source agents like solar, wind, batteries, and grid agents that act as a common element for all microgrids. Agents implementation in Fig. 6.

Snippet of code for agent initialization:

```java
import jade.core.Agent

public class loadA extends agentProcessor {
    public void setup() {
        // all the initialization
        // of the agents
        // happens here
    }
}
```

B. Implementing Agents Communication

The ability of agents to communicate is the most basic feature of MAS. Agents communicate with one another by exchanging information-containing messages. Depending on the agent development framework chosen, a suitable agent communication language must be utilized to implement the communication between agents. A message contains a variety of auxiliary data in addition to the content information, such as the message type, sender information, recipient information, etc. The process is straightforward and frequently involves maintaining the text as a "String" with application-specific meaning.

```java
ACLMessage msg = new ACLMessage(ACLMessage.INFORM)
msg.setContent("power values will be given here");
msg.addReceiver(new AID("load1",AID.ISLOCALNAME )
send(msg);
```
Multiple other types of communication signals are available in the JADE library (Agent class), however those are too sophisticated and out of scope for the current implementations. Fig. 7 shows a sample of ACLMessages being communicated between the agents.

![Communication between different Agents](image)

**Fig. 7. Communication between different Agents**

C. Outputs

The primary goal of this management system, which was developed using agents, is to ensure that the agent that corresponds to load receives the power from the source agent with the best pricing and capacity for that time. Each agent is ranked by the algorithm from best to worst, and it switches from one to another as it runs out of power.

1) Case 1:

Load demand is set to 20KW and the solar1 agent provides power at Rs 1/KW, all other agents provide at Rs 2/KW.

The load agent accepts the proposal of the solar1 agent since it is the most cost efficient given its minimal cost and maximal power, other agents' proposals for power supply are rejected as already the demand is being met. Figures 8 and 9 depict the results of the current scenario.

```
load1 is ready for power consumption and needs 20 KW
load1 approaching wind1
load1 approaching battery1
load1 approaching grid
load1 approaching solar1
load1 tapped 20 KW of power from solar1
load1 rejected the proposal of grid
load1 rejected the proposal of battery1
load1 rejected the proposal of wind1
```

**OUTPUT:**

```
Load1 demand:20
Power Available in solar1:130
Power tapped from solar1:120
Price per unit KW: Rs1
Amount spent for solar1: Rs20
Power Available in grid:1000
Power tapped from grid:0
Price per unit KW: Rs2
Amount spent for grid: Rs0
Power Available in battery1:201
Power tapped from battery1:0
Price per unit KW: Rs3
Amount spent for battery1: Rs0
Power Available in wind1:20
Power tapped from wind1:0
Price per unit KW: Rs3
Amount spent for wind1: Rs0
Total Price: Rs20
```

**Fig. 8. Screenshot of Workspace 1**

![Screenshot of Workspace 1](image)

**Fig. 9. Distribution for case 1**

2) Case 2:

Load demand is set to 500KW, wind1 agent supplies power at Rs 1/KW, solar & battery1 agent supplies at Rs 2/KW each respectively and the grid agent supplies at Rs 3/KW

Since there is a large demand for the load in this situation, the load agent draws power from all the sources with varying prices, but in the way the final price is the
least price possible. Figures 10 and 11 exhibit the results of the current circumstance.

```
load1 is ready for power consumption and needs 500 KW
load1 approaching solar1
load1 approaching wind1
load1 approaching grid
load1 approaching battery1
load1 tapped 20 KW of power from wind1
load1 tapped 200 KW of power from battery1
load1 tapped 100 KW of power from solar1
load1 tapped 180 KW of power from grid

OUTPUT:
-------
Load1 demand: 500
Power Available in wind1: 20
Power tapped from wind1: 20
Price per unit KW: Rs 1
Amount spent for wind1: Rs 20
Power Available in battery1: 200
Power tapped from battery1: 200
Price per unit KW: Rs 1
Amount spent for battery1: Rs 400
Power Available in solar1: 100
Power tapped from solar1: 100
Price per unit KW: Rs 1
Amount spent for solar1: Rs 200
Power Available in grid: 1000
Power tapped from grid: 1000
Price per unit KW: Rs 1
Amount spent for grid: Rs 540
Total Price: Rs 1160
```

---

**Fig. 10. Console Output 2**

**Fig. 11. Sniffer Output for case 2**

3) **Case 3:**

Load demand is set to 400KW, solar1 & wind1 agent supplies power at Rs 1/KW each, grid and battery1 agent supplies at Rs 2/KW each.

In this scenario, two sets of source agents (wind1 & solar1, grid & battery1) will offer power at the same price, thus the load agent will search for the agent with the biggest power capacity. The agent with the maximum capacity at the same price is considered the best source agent, and if demand needs to be satisfied, it taps from that agent before moving to the next best agent. Figures 12 and 13 depict the results of the current scenario.

```
load1 is ready for power consumption and needs 400 KW
load1 approaching solar1
load1 approaching wind1
load1 approaching grid
load1 approaching battery1
load1 tapped 100 KW of power from solar1
load1 tapped 20 KW of power from wind1
load1 tapped 100 KW of power from grid
load1 rejected the proposal of battery1

OUTPUT:
-------
Load1 demand: 400
Power Available in solar1: 100
Power tapped from solar1: 100
Price per unit KW: Rs 1
Amount spent for solar1: Rs 100
Power Available in wind1: 20
Power tapped from wind1: 20
Price per unit KW: Rs 1
Amount spent for wind1: Rs 20
Power Available in grid: 1000
Power tapped from grid: 1000
Price per unit KW: Rs 1
Amount spent for grid: Rs 560
Power Available in battery1: 200
Power tapped from battery1: 0
Price per unit KW: Rs 2
Amount spent for battery1: Rs 0
Total Price: Rs 660
```

---

**Fig. 12. Console Output 3**

**Fig. 13. Sniffer Output 3**

VI. **MACHINE LEARNING MODELS**

Pandas, Scikit-learn and TensorFlow packages are essential requirements for the project and is downloaded using Anaconda. The snippet is implemented on Jupyter notebook due to its advantageous web-based compiling of the code enabling easy collaborating and scalability [10]. The required data for creating the machine learning model is collected from the Electrical and Electronics department of SSN college for a cumulative time period of one year in order to achieve the required volume of data to ensure accurate results. The data collected is of a load curve, which is based on the demand(load) at every point of time of the day. The data is split in 80:20 ratio in order to assign the majority share to train and optimize the model and the remaining minority is used to check
and tweak the model further to perfect predictions. The first basic model used to get a base performance of prediction was the linear regression model, which is inbuilt in the sci-kit learn package, this marks the benchmark performance based on which other models are compared and contrasted. The load is the predictor here and the various parameters, initially just time is the feature [10]. The general rule of thumb is to have several highly corelative parameters in order to allow for a better modelling of an algorithm. TensorFlow has the MLR package which allows to create the necessary model. The linear regression model works by passing the most suitable straight line with appropriate slope through the data, optimizing parameters based on sum of least squares methodology, but since the load curve tends to follow a natural bell curve model, this provides a vastly inaccurate modelling of the dataset and hence produces poor results, this led to the experimenting and idea to use a Multi-layer perceptron based Neural Network model which works mimicking the functioning of a brain based on connecting paths through weights assigned to them and the RBF model also seems to be an extensively complex yet supremely accurate way of modelling the dataset. Python has a powerful tool in the form of TensorFlow which has all the useful AI and ML models built in creating abstraction so that the user creates powerful models without knowing the ins and outs of the working of the model. Visualization tools also makes it a wonderful overall package to work. Classification, Regression, Neural Networks, Deep Learning, are easily implemented using the TensorFlow packages. Having powerful models may turn out to be completely useless if the datasets are not clean and appropriate, the Pandas package is used for this purpose along with NumPy to perform mathematical manipulation. Prior discussion talks about the importance of the data manipulation and the number of predictors hence we go with an additional parameter which is the pricing of the units of power during different load points.

I. Simulation results

The base setup and importing of the required files and modules are depicted in Figure14. NumPy allows use of MSE (Mean Squared Error) and R2 test in order to assess the fitness of the dataset.

On analysis the dataset appears to be very good based on the very small value of MSE which can be neglected. Figure 15 depicts the results achieved on the R2 test for the RBF model which is 98.2% which depicts the accuracy of the results which means the % of correct predictions for the test dataset.

![Figure 15. RBF model analysis](image)

The results are graphically represented in the above diagram for the RBF kernel which plots the actual and predicted value from the linear trend and immensely high R2 score of 0.99 is a testament to the accuracy of the model.

![Figure 16. RFR model analysis](image)

The result are graphically represented in the above diagram for the RFR (Random Forest Regressor) kernel which plots the actual and predicted value from the linear trend and immensely high R2 score of 0.989, is a testament to the accuracy of the model and proves that a nonlinear dataset can be approached from multiple angles and still be predicted with extremely high accuracy.

Visualization is an extremely useful and key process to understanding ML models and its accuracy and the same applies to the overall dataset available, in this project the versatile Matplotlib package is used which allows us to view the data and analyze the models in multiple visual models just by importing the tools using bare minimum lines of code.
VII. CONCLUSION
This research highlights the Energy transactions based on MA using Machine learning for optimized demand side management for a microgrid consisting of 3 sources, a load and a battery. An entire microgrid has been simulated using JADE with real-world data collected from accessible sources. Distributed energy systems and a battery were successfully combined to form a MAS in the JADE IDE with the following features:

- Multiple microgrids were combined and made to co-exist using MAS, in the JADE IDE
- The fundamental communication between the multiple agents to enable autonomous operation was achieved.
- Based on responsibilities of the sources each agent represents it was made to conform to the rules of the energy system as a whole
- Realistic data as well as scenarios were used to check the integrity and robustness of the system.

For analyzing and providing demand side management Machine Learning is used, MLR, RBF and Random Forest Regressor models are opted, out of which due to the odd trend of the data the RBF and latter models succeed compared to the MLR mode. The scope of this model includes addition of multiple highly corelative features such as weather conditions to further increase quality of data and accuracy of predictions. Deep learning is a potent alternative when features exceed a certain threshold. With the advent of smart meters and clustering data is becoming more and more available in order to optimize demand response management and to categorize users based on patters on consumption to further improve the whole power system. Real time pricing of power based on active consumption values is the end goal of the power market, which is moved to a step closer by this project. The power of computers and its ability to compute complex problems are used to solve the current problems of optimization. ML will soon be a useful and necessary companion to create a precise smart grid system.

VIII. REFERENCES
[1] G. H. Merabet et al., “Applications of Multi-Agent Systems in Smart Grids: A survey,” 2014 International Conference on Multimedia Computing and Systems (ICMCS), Marrakech, Morocco, 2014, pp. 1088-1094, doi: 10.1109/ICMCS.2014.6911384.
[2] Fabio Luigi Bellifemine, Giovanni Caire and Dominic Greenwood, Developing Multi-Agent Systems with JADE. Wiley; February 2007
[3] O. P. Mahela et al., “Comprehensive overview of multi-agent systems for controlling smart grids,” in CSEE Journal of Power and Energy Systems, vol. 8, no. 1, pp. 115-131, Jan. 2022, doi: 10.17775/CSEEJPS.2020.03390.
[4] P. A. Owusu, S. Asumadu-Sarkodie, and S. Dubey, "A review of renewable energy sources, sustainability issues and climate change mitigation,"Cogent Engineering, vol. 3, no. 1, 2016.
[5] H. Han, X. Hou, J. Yang, J. Wu, M. Su, and J. M. Guerrero, "Review of Power Sharing Control Strategies for Islanding Operation of AC Microgrids," IEEE Transactions on Smart Grid, vol. 7, no. 1, pp. 200-215, 2016.
[6] B. Hangun, O. Eyecioglu and M. Beken, "Forecasting the Stability of A 4-node Architecture Smart Grid Using Machine Learning," 2022 10th International Conference on Smart Grid (icSmartGrid), Istanbul, Turkey, 2022, pp. 440-442, doi: 10.1109/icSmartGrid55722.2022.9846635.
[7] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures," in IEEE Signal Processing Magazine, vol. 26, no. 1, pp. 98-117, Jan. 2009, doi: 10.1109/MSP.2008.930649.
[8] Leo Raju, P.R Muthukumar, S. Gokulakrishnan, S. Jagannathan, "Autonomous Demand Side Management of a Micro-Grid Using Arduino and Multi-agent System," Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM), pp. 159-164, 2017.
[9] M. Rafik, A. Bahnasse, A. Khiat, O. Bouattane, H. Ouaiji, and M. E. Khaili, "Towards a Smart Energy Sharing in "Micro Smart Grid Adopting SDN Approach," Procedia Computer Science, vol. 151, pp. 717-724, 2019.
[10] W. Liu et al., "Smart Micro-grid System with Wind/PV/Battery," Energy Procedia, vol. 152, pp. 1212-1217, 2018.
[11] S. Woltinann, M. Zarte, J. Kittel, and A. Pechmann, "Agent Based Simulation Model of Virtual Power Plants for Greener Manufacturing," Procedia CIRP, vol. 69, pp. 377-382, 2018.
[12] Leo Raju, E. Vishal, V. Vishwaraj and K. M. Vimalan, "Application of Machine Learning Algorithms for Short term Load Prediction of Smart grid," 2020 International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2020, pp. 371-376.
[13] S. Vasu, R. K. P and J. E. A, "Realizing Autonomous and Intelligent Smart Grid Using Multi-Agent Based Control System," 2020 International Conference on Power Electronics and Renewable Energy Applications (PEREA), Kannur, India, 2020, pp. 1-6, doi: 10.1109/PEREAS1218.2020.9339784.