ENHANCED MACHINE-LEARNING MODEL FOR ANALYZING AND PREDICTING ENERGY CONSUMPTION IN SMART HOMES

Report submitted to the SASTRA Deemed to be University

as the requirement for the course

CSE300 - MINI PROJECT

Submitted by

THOTAKURA ROHIT SAI NAREN

(Reg. No.: 124003249, B.Tech CSE)

VADLA KALYAN

(Reg. No.: 124003128, B.Tech CSE)

BHUMIREDDY HARINATHREDDY

(Reg. No.: 124003385, B.Tech CSE)

MAY 2023

SCHOOL OF COMPUTING

THANJAVUR, TAMIL NADU, INDIA – 613 401

SCHOOL OF COMPUTING

THANJAVUR – 613 401

Bonafide Certificate

This is to certify that the report titled "**ENHANCED MACHINE-LEARNING MODEL FOR ANALYSING AND PREDICTING ENERGY CONSUMPTION IN SMART HOMES**" submitted as a requirement for the course, CSE300 : **MINI PROJECT** for B.Tech. is a bonafide record of the work done by **Mr. Thotakura Rohit Sai Naren (Reg. No.: 124003249, B.Tech CSE), Mr. Vadla Kalyan (Reg. No.: 124003128, B.Tech CSE), Mr. BhumiReddy Harinath Reddy (Reg. No.: 124003385, B.Tech CSE)** during the academic year 2022-23, in the School of Computing, under my supervision.

Signature of Project Supervisor :

Name with Affiliation : **Ms. Kavitha R** , Asst. Professor-III, SOC

Date : **/05/20223**

Mini Project *Viva voce* held on /05/2023

Examiner 1 **Examiner 2**

Acknowledgements

We would like to thank our honourable Chancellor **Prof. R. Sethuraman** for providing us with an opportunity and the necessary infrastructure for carrying out this project as a part of our curriculum.

We would like to thank our honourable Vice-Chancellor **Dr. S.Vaidhyasubramaniam** and **Dr. S. Swaminathan,** Dean**,** Planning & Development, for the encouragement and strategic support at every step of our college life.

We extend our sincere thanks to **Dr. R. Chandramouli,** Registrar, SASTRA Deemed to be University for providing the opportunity to pursue this project.

We extend our heartfelt thanks to **Dr. A. Umamakeswari**, Dean, School of Computing, **Dr. V. S. Shankar Sriram,** Associate Dean, Department of Computer Science and Engineering and **Dr. R. Muthaiah,** Associate Dean, Department of Information Technology and Information & Communication Technology for their motivation and support offered in materializing this project.

Our guide **Ms.Kavitha R** , Asst. Professor-III, School of Computing was the driving force behind this whole idea from the start. Her deep insight in the field and invaluable suggestions helped us in making progress throughout our project work. We also thank the project review panel members for their valuable comments and insights which made this project better.

We would like to extend our gratitude to all the teaching and non-teaching faculties of the School of Computing who have either directly or indirectly helped us in the completion of the project.

We gratefully acknowledge all the contributions and encouragement from our family and friends resulting in the successful completion of this project.

List of Figures

List of Tables

Abbreviations

XGBoost	Extreme Gradient Boost
RSME	Root Square Mean Error
MAE	Mean Absolute Error
RF	Random Forest
DT	Decision Tree
IoT	Internet of Things

Abstract

Smart homes are becoming increasingly popular, with a range of devices that can automate processes and improve our lives. However, these devices also consume a considerable amount of energy, leading to an increase in carbon footprint and demand for supply. Therefore, it is essential to monitor and reduce energy consumption in smart homes. This paper uses an Ensemble Model based on machine learning techniques to predict power usage in smart homes. The model uses Decision Trees, Random Forest, and eXtreme Gradient Boosting techniques to create an ensemble approach that can better predict energy usage. We evaluate the performance of the Ensemble Model on multiple datasets, and our results demonstrate that it outperforms baseline algorithms in terms of MSE, R-squared, RMSE, and MAE. Our study highlights the importance of monitoring energy consumption in smart homes and emphasizes the potential of ensemble machine learning techniques to improve efficiency and reduce carbon footprint. By reducing energy consumption in smart homes, we can lower costs and improve sustainability. Overall, our study provides a useful framework for analyzing energy consumption in smart homes, which can help homeowners make informed decisions and optimize energy usage.

KEYWORDS: Random forest regression , XGBoost regression , Decision Tree, DT-RF-XGBoost Ensemble Model

Table of Contents

CHAPTER 1
SUMMARY OF THE BASE PAPER

Title: Enhanced Machine Learning Model For Analyzing and Predicting Energy Consumption in Smart Homes

Journal : Internet of Things

Authors : Ishani Priyadarshini, Sandipan Sahu, Raghvendra Kumar, David Taniar

Published : 2022

Base Paper Link : https://doi.org/10.1016/j.iot.2022.100636

1.1 INTRODUCTION :

Smart homes are the current trend in the global world as they provide comfort, user-friendly and energy efficient. For remote monitoring and administration of appliances, which are managed by networked or smart home software, these homes rely on internet-connected gadgets. On the Internet of Things (IoT) platform, smart homes are constructed and include sensors, speakers, smart lighting, cameras, locks, door openers, and other gadgets.

Although energy savings from smart homes can be significant, the effectiveness can still be raised. More connected cameras and smart speakers need more electricity, and poorly built buildings can also result in inefficient power use. Methods are being actively investigated to solve this problem.

The rising demand for energy consumption, which raises the danger of climate change and expands carbon footprints, is one of the main causes of inefficient power usage. For load-balancing power plants and carrying out load studies, which are crucial components of energy monitoring, monitoring power consumption is required. Many machine learning techniques have been used to track energy usage, including Linear Regression, Support Vector Machines, and Long-Short Term Memory. However, ensemble approaches can overcome the shortcomings of standard machine learning models, such as overfitting and cost.

The eXtreme Gradient Boosting, Decision Trees, and Random Forests are used in an ensemble-based method for predicting power usage in smart homes. Gradient Boosting, K Nearest Neighbours, Decision Trees, Random Forests, and other baseline models are used to assess how well our suggested ensemble method performs. Multiple metrics such as Mean Square Error, R-squared Error, Root Mean Square Error, and Mean Absolute Error, are used to evaluate the performance of the models. We conduct a thorough study on two distinct datasets that include readings of household applications in kilowatts taken from smart metres during a time period of one minute.

1.2 LITERATURE SURVEY :

Sl.No.	Paper Title	Year	Technique	Advantages
1	[1]A systematic review of the smart energy conservation system : From smart homes to sustainable smart cities	2021	MANFIS	Significant reduction in cost of electricity.
2	[2]Smart Home Energy saving system	2020	Big data and machine learning	Reduces energy consumption
3	[3]Effective energy management for smart buildings	2021	Deep Learning and IOT based approach (YOLOv3)	Enhanced decision making about energy consumption
4	[4]Optimal household appliance scheduling	2021	Fuzzy Logic and Machine Learning	Reduction in electricity, cost, optimal scheduling
5	[5]Enhanced Load Monitoring in Smart Homes	2021	Artificial bee colony algorithm	Efficient Load Monitoring

1.2 Literature Survey

1.3 METHODOLOGY AND WORKPLAN

1.3.1. DATA COLLECTION:

We have collected two datasets from *‘smart-home-dataset-with-weather-information1’* from Kaggle. Each data set has 32 features and around 5 lakh records.

The datasets are:

1.HOME.csv

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	time	use [kW]	gen [kW]	House ove	Dishwashe	Furnace 1	Furnace 2	Home offic	Fridge [kV	Wine cella	Garage dc	Kitchen 12	Kitchen 14	Kitchen 38
2	1.45E+09	0.932833	0.003483	0.932833	3.33E-05	0.0207	0.061917	0.442633	0.12415	0.006983	0.013083	0.000417	0.00015	0
3	1.45E+09	0.934333	0.003467	0.934333	0	0.020717	0.063817	0.444067	0.124	0.006983	0.013117	0.000417	0.00015	0
4	1.45E+09	0.931817	0.003467	0.931817	1.67E-05	0.0207	0.062317	0.446067	0.123533	0.006983	0.013083	0.000433	0.000167	1.67E-05
5	1.45E+09	1.02205	0.003483	1.02205	1.67E-05	0.1069	0.068517	0.446583	0.123133	0.006983	0.013	0.000433	0.000217	0
6	1.45E+09	1.1394	0.003467	1.1394	0.000133	0.236933	0.063983	0.446533	0.12285	0.00685	0.012783	0.00045	0.000333	0
7	1.45E+09	1.391867	0.003433	1.391867	0.000283	0.50325	0.063667	0.447033	0.1223	0.006717	0.012433	0.000483	0.000567	0
8	1.45E+09	1.366217	0.00345	1.366217	0.000283	0.4994	0.063717	0.443267	0.12205	0.006733	0.012417	0.000517	0.00055	0
9	1.45E+09	1.4319	0.003417	1.4319	0.00025	0.477867	0.178633	0.444283	0.1218	0.006783	0.01255	0.000483	0.00045	0
10	1.45E+09	1.6273	0.003417	1.6273	0.000183	0.44765	0.3657	0.441467	0.121617	0.00695	0.012717	0.000467	0.0003	1.67E-05
11	1.45E+09	1.735383	0.003417	1.735383	1.67E-05	0.17155	0.6825	0.438733	0.121633	0.007233	0.01335	0.000367	5.00E-05	0
12	1.45E+09	1.585083	0.003417	1.585083	5.00E-05	0.0221	0.678733	0.4402	0.12145	0.007433	0.013583	0.00035	0.000117	3.33E-05
13	1.45E+09	1.510317	0.003433	1.510317	3.33E-05	0.021967	0.620667	0.43695	0.12125	0.007317	0.013533	0.000333	0.0001	0
14	1.45E+09	1.459867	0.00345	1.459867	5.00E-05	0.021883	0.577467	0.43995	0.121033	0.007233	0.013517	0.000367	8.33E-05	1.67E-05
15	1.45E+09	0.840583	0.003433	0.840583	0	0.02095	0.1448	0.444783	0.035017	0.007033	0.013183	0.00065	0.000183	1.67E-05
16	1.45E+09	0.7032	0.003433	0.7032	1.67E-05	0.020733	0.061967	0.443833	0.004783	0.006967	0.013117	0.000733	0.000233	0
17	1.45E+09	0.571883	0.00345	0.571883	0	0.02065	0.06365	0.307783	0.004917	0.00705	0.0131	0.000733	0.00015	0
18	1.45E+09	0.485733	0.00345	0.485733	1.67E-05	0.020617	0.063433	0.22045	0.004983	0.007033	0.013117	0.00075	8.33E-05	0
19	1.45E+09	0.523167	0.003433	0.523167	0	0.020633	0.062117	0.26005	0.00495	0.007	0.013083	0.000733	0.0001	1.67E-05
20	1.45E+09	0.5362	0.00345	0.5362	0	0.020683	0.062917	0.272067	0.00495	0.007033	0.01315	0.000733	0.000117	0

	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	Barn [kW]	Well [kW]	Microwave	Living roo	Solar [kW	temperatu	icon	humidity	visibility	summary	apparentT	pressure
2	0.03135	0.001017	0.004067	0.001517	0.003483	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91
3	0.0315	0.001017	0.004067	0.00165	0.003467	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91
4	0.031517	0.001	0.004067	0.00165	0.003467	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91
5	0.0315	0.001017	0.004067	0.001617	0.003483	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91
6	0.0315	0.001017	0.004067	0.001583	0.003467	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91
7	0.03145	0.001017	0.004067	0.001583	0.003433	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91
8	0.03155	0.001033	0.004117	0.001533	0.00345	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91
9	0.031733	0.001033	0.0042	0.00155	0.003417	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91
10	0.031767	0.001017	0.0042	0.001567	0.003417	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91
11	0.031667	0.001017	0.0042	0.001617	0.003417	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91
12	0.031667	0.001	0.0042	0.001567	0.003417	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91
13	0.03175	0.001	0.0042	0.001567	0.003433	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91
14	0.031783	0.001	0.004217	0.001583	0.00345	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91
15	0.031783	0.001017	0.004217	0.001617	0.003433	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91
16	0.03175	0.001017	0.004233	0.00155	0.003433	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91
17	0.031733	0.001	0.004217	0.001583	0.00345	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91
18	0.031833	0.001	0.004233	0.001617	0.00345	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91
19	0.03185	0.001017	0.00425	0.001717	0.003433	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91
20	0.031867	0.001017	0.004233	0.00165	0.00345	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91

Fig 1.3.1 Snapshot of Sample HomeC dataset

2.HOME1.csv

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	time	use [kW]	gen [kW]	House ove	Dishwashe	Furnace 1	Furnace 2	Home offic	Fridge [kV	Wine cella	Garage dc	Kitchen 12	Kitchen 14	Kitchen 38
2	1.45E+09	0.932833	0.003483	0.932833	3.33E-05	0.0207	0.061917	0.442633	0.12415	0.006983	0.013083	0.000417	0.00015	0
3	1.45E+09	0.934333	0.003467	0.934333	0	0.020717	0.063817	0.444067	0.124	0.006983	0.013117	0.000417	0.00015	0
4	1.45E+09	0.931817	0.003467	0.931817	1.67E-05	0.0207	0.062317	0.446067	0.123533	0.006983	0.013083	0.000433	0.000167	1.67E-05
5	1.45E+09	1.02205	0.003483	1.02205	1.67E-05	0.1069	0.068517	0.446583	0.123133	0.006983	0.013	0.000433	0.000217	0
6	1.45E+09	1.1394	0.003467	1.1394	0.000133	0.236933	0.063983	0.446533	0.12285	0.00685	0.012783	0.00045	0.000333	0
7	1.45E+09	1.391867	0.003433	1.391867	0.000283	0.50325	0.063667	0.447033	0.1223	0.006717	0.012433	0.000483	0.000567	0
8	1.45E+09	1.366217	0.00345	1.366217	0.000283	0.4994	0.063717	0.443267	0.12205	0.006733	0.012417	0.000517	0.00055	0
9	1.45E+09	1.4319	0.003417	1.4319	0.00025	0.477867	0.178633	0.444283	0.1218	0.006783	0.01255	0.000483	0.00045	0
10	1.45E+09	1.6273	0.003417	1.6273	0.000183	0.44765	0.3657	0.441467	0.121617	0.00695	0.012717	0.000467	0.0003	1.67E-05
11	1.45E+09	1.735383	0.003417	1.735383	1.67E-05	0.17155	0.6825	0.438733	0.121633	0.007233	0.01335	0.000367	5.00E-05	0
12	1.45E+09	1.585083	0.003417	1.585083	5.00E-05	0.0221	0.678733	0.4402	0.12145	0.007433	0.013583	0.00035	0.000117	3.33E-05
13	1.45E+09	1.510317	0.003433	1.510317	3.33E-05	0.021967	0.620667	0.43695	0.12125	0.007317	0.013533	0.000333	0.0001	0
14	1.45E+09	1.459867	0.00345	1.459867	5.00E-05	0.021883	0.577467	0.43995	0.121033	0.007233	0.013517	0.000367	8.33E-05	1.67E-05
15	1.45E+09	0.840583	0.003433	0.840583	0	0.02095	0.1448	0.444783	0.035017	0.007033	0.013183	0.00065	0.000183	1.67E-05
16	1.45E+09	0.7032	0.003433	0.7032	1.67E-05	0.020733	0.061967	0.443833	0.004783	0.006967	0.013117	0.000733	0.000233	0
17	1.45E+09	0.571883	0.00345	0.571883	0	0.02065	0.06365	0.307783	0.004917	0.00705	0.0131	0.000733	0.00015	0
18	1.45E+09	0.485733	0.00345	0.485733	1.67E-05	0.020617	0.063433	0.22045	0.004983	0.007033	0.013117	0.00075	8.33E-05	0
19	1.45E+09	0.523167	0.003433	0.523167	0	0.020633	0.062117	0.26005	0.00495	0.007	0.013083	0.000733	0.0001	1.67E-05
20	1.45E+09	0.5362	0.00345	0.5362	0	0.020683	0.062917	0.272067	0.00495	0.007033	0.01315	0.000733	0.000117	0
21	1.45E+09	0.53415	0.00345	0.53415	1.67E-05	0.020667	0.06265	0.270067	0.00495	0.0071	0.01315	0.000733	0.0001	0
22	1.45E+09	0.533817	0.00345	0.533817	0	0.020633	0.062967	0.270033	0.00495	0.007033	0.013117	0.00075	0.0001	0
23	1.45E+09	0.523633	0.00345	0.523633	1.67E-05	0.02055	0.063283	0.259817	0.00495	0.007033	0.013083	0.000733	0.000117	0
24	1.45E+09	0.57725	0.003417	0.57725	0	0.020683	0.10975	0.257	0.004983	0.007067	0.013133	0.000733	8.33E-05	0
25	1.45E+09	0.679567	0.003433	0.679567	1.67E-05	0.020867	0.194083	0.2571	0.005017	0.007117	0.0132	0.0007	6.67E-05	1.67E-05
26	1.45E+09	1.293167	0.0034	1.293167	1.67E-05	0.107433	0.622467	0.2541	0.0052	0.00725	0.01335	0.00065	0.0001	3.33E-05
27	1.45E+09	1.546667	0.003433	1.546667	1.67E-05	0.220567	0.68005	0.253933	0.00555	0.007267	0.013183	0.0007	6.67E-05	1.67E-05
28	1.45E+09	1.82235	0.003417	1.82235	6.67E-05	0.49665	0.677	0.254533	0.004917	0.0071	0.012783	0.000717	0.000117	1.67E-05
29	1.45E+09	2.642517	0.002033	2.642517	8.33E-05	0.49875	0.575567	0.255533	0.004567	0.007033	0.012667	0.0004	0.000583	1.67E-05
30	1.45E+09	1.73935	0.003133	1.73935	0.0001	0.49445	0.48835	0.255817	0.00475	0.007017	0.012717	0.000617	0.000317	1.67E-05
31	1.45E+09	1.1872	0.003083	1.1872	0.00025	0.453817	0.062883	0.256033	0.004567	0.024717	0.0125	0.0007	0.0006	0
32	1.45E+09	1.19095	0.003083	1.19095	0.000217	0.3874	0.061883	0.256867	0.004933	0.101783	0.012533	0.0007	0.000583	1.67E-05
33	1.45E+09	0.833083	0.003067	0.833083	1.67E-05	0.02065	0.063067	0.254583	0.004817	0.097183	0.013017	0.000667	0.000217	1.67E-05
34	1.45E+09	0.834283	0.003067	0.834283	0	0.020617	0.062033	0.25655	0.004817	0.100833	0.013033	0.000667	0.00025	1.67E-05
35	1.45E+09	0.840433	0.003067	0.840433	1.67E-05	0.0206	0.06265	0.256017	0.004817	0.10915	0.013033	0.00065	0.00025	0
36	1.45E+09	0.869733	0.00305	0.869733	1.67E-05	0.02055	0.061917	0.256183	0.004817	0.113067	0.012983	0.000667	0.000233	0
37	1.45E+09	0.853883	0.00305	0.853883	0	0.020567	0.063733	0.2553	0.005217	0.114267	0.013	0.000667	0.00025	0
38	1.45E+09	0.8491	0.003083	0.8491	1.67E-05	0.0206	0.063483	0.256683	0.004817	0.116667	0.013033	0.000667	0.00025	1.67E-05
39	1.45E+09	1.013333	0.00305	1.013333	0	0.020933	0.198367	0.2542	0.004883	0.119133	0.013117	0.00065	0.000183	0
40	1.45E+09	1.3159	0.003033	1.3159	1.67E-05	0.0214	0.415217	0.2478	0.004983	0.120567	0.013317	0.000583	0.0001	1.67E-05

	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
1	Barn [kW]	Well [kW]	Microwave	Living roo	Solar [kW	temperatu	icon	humidity	visibility	summary	apparentT	pressure	windSpee	cloudCove	windBeari	precipInte	dewPoint
2	0.03135	0.001017	0.004067	0.001517	0.003483	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
3	0.0315	0.001017	0.004067	0.00165	0.003467	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
4	0.031517	0.001	0.004067	0.00165	0.003467	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
5	0.0315	0.001017	0.004067	0.001617	0.003483	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
6	0.0315	0.001017	0.004067	0.001583	0.003467	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
7	0.03145	0.001017	0.004067	0.001583	0.003433	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
8	0.03155	0.001033	0.004117	0.001533	0.00345	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
9	0.031733	0.001033	0.0042	0.00155	0.003417	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
10	0.031767	0.001017	0.0042	0.001567	0.003417	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
11	0.031667	0.001017	0.0042	0.001617	0.003417	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
12	0.031667	0.001	0.0042	0.001567	0.003417	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
13	0.03175	0.001	0.0042	0.001567	0.003433	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
14	0.031783	0.001	0.004217	0.001583	0.00345	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
15	0.031783	0.001017	0.004217	0.001617	0.003433	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
16	0.03175	0.001017	0.004233	0.00155	0.003433	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
17	0.031733	0.001	0.004217	0.001583	0.00345	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
18	0.031833	0.001	0.004233	0.001617	0.00345	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
19	0.03185	0.001017	0.00425	0.001717	0.003433	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
20	0.031867	0.001017	0.004233	0.00165	0.00345	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
21	0.0319	0.001	0.004233	0.00175	0.00345	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
22	0.031817	0.001017	0.004233	0.001567	0.00345	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
23	0.031733	0.001017	0.004217	0.001667	0.00345	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
24	0.031683	0.001017	0.004233	0.001517	0.003417	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
25	0.031717	0.001017	0.004217	0.00165	0.003433	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
26	0.031567	0.001	0.0042	0.001567	0.0034	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
27	0.031067	0.001	0.004167	0.00165	0.003433	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
28	0.030967	0.001017	0.010067	0.0016	0.003417	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
29	0.088067	0.0015	1.040767	0.001367	0.002033	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
30	0.113517	0.001167	0.004383	0.001583	0.003133	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
31	0.031767	0.001067	0.0045	0.001167	0.003083	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
32	0.03175	0.00105	0.0045	0.00115	0.003083	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
33	0.03175	0.001067	0.0045	0.001167	0.003067	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
34	0.031733	0.00105	0.004517	0.0012	0.003067	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
35	0.031683	0.001067	0.0045	0.001183	0.003067	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
36	0.031833	0.001067	0.004533	0.001233	0.00305	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
37	0.03175	0.001067	0.004517	0.00115	0.00305	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
38	0.0318	0.00105	0.004517	0.001167	0.003083	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
39	0.0317	0.001067	0.0045	0.001217	0.00305	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4
40	0.086217	0.00115	0.00445	0.00135	0.003033	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCove	282	0	24.4

Fig 1.3.2 Snapshot of Sample HomeC1 dataset

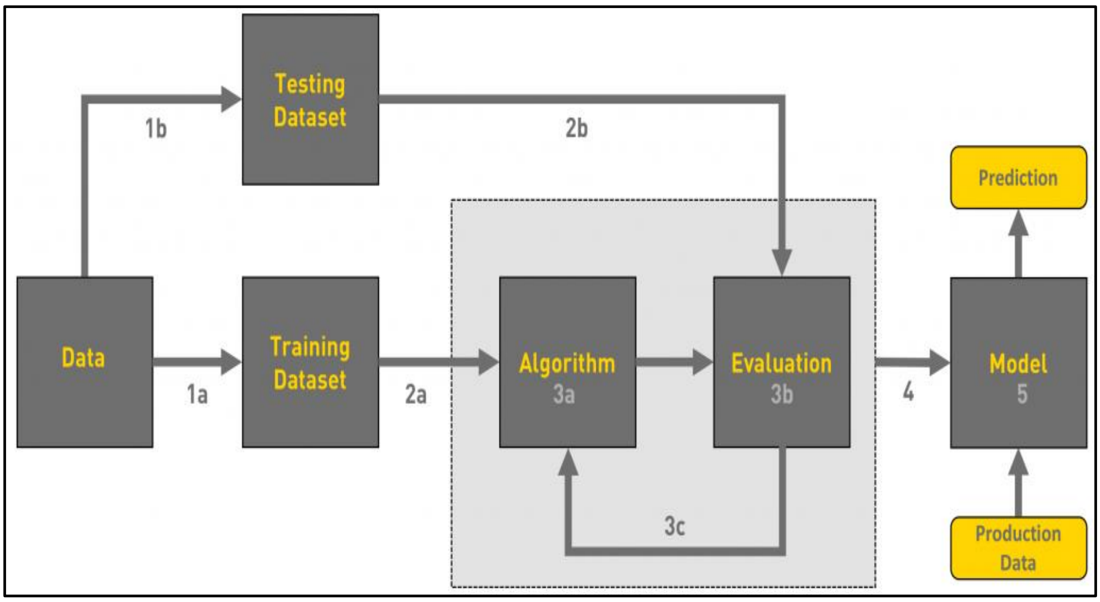

Fig 1.3 Work Flow of the project

1.3.2 DATA PREPROCESSING:

The dataset is containing 32 features, in those features time is taken as index. Factors that are affecting energy consumption that are weather parameters are taken into consideration.

Energy consumption in various components is given in Kw which is replaced and made into a numeric. In this dataset, time is recorded in the unix time format, we converted this large time stamp value to a readable data converted all the data type features into float. Removed the duplicate columns Using time series prediction data has been columnised and then divided dataset to training (80%) and testing(20%).

1.3.3: Train Test Split

Splitting the data set into the training and testing sets. Generally, data sets are divided into 80:20 ratio which is the most practiced split. After this splitting we will train the model using the Training set and test the Model using Testing set.

1.3.4 MACHINE LEARNING TECHNIQUES:

The machine learning models are constructed after the features are extracted from the dataset. The models which are used are:

1. Decision Tree
2. Random Forest Regression
3. XGBoost Regression
4. DT-RF-XGBoost Ensemble Model

1.3.4.1 Decision Tree

The decision tree algorithm is a popular and simple machine learning technique that is us for both classification and regression tasks. It uses set of rules and decision nodes.

The decision tree algorithm works by recursively splitting the data into smaller subsets based on the most significant attribute or feature. The splitting is done based on a set of rules or decision nodes, which are learned from the training data. The algorithm constructs the decision tree by following a top-down approach where each internal node of the tree represents a test on an attribute or feature, each branch represents the outcome of the test, and each leaf node represents a class label or a prediction value.

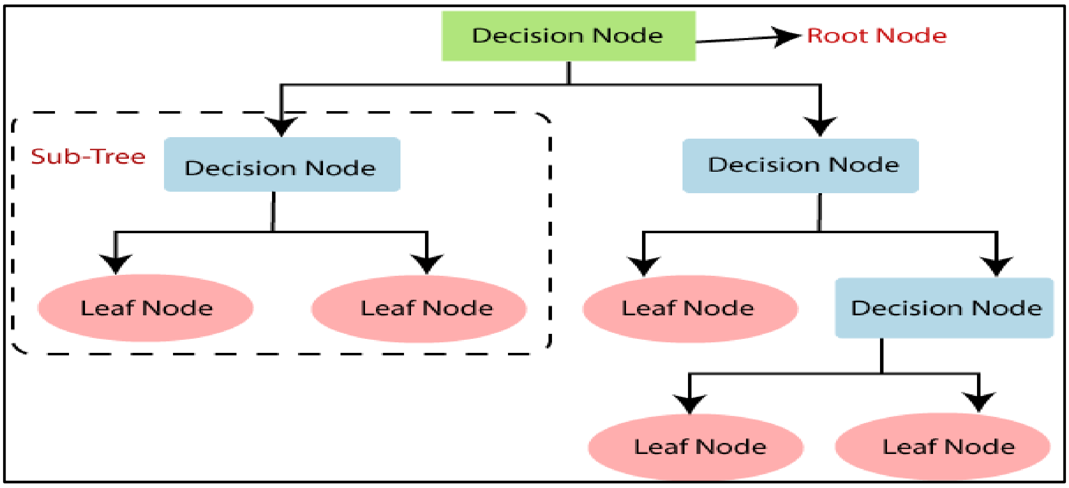

Fig 1.3.4.1.1 Decision Tree

1.3.4.2 Random Forest

Random Forest Regressor is machine learning technique that combines multiple decision trees to produce a more accurate prediction model. It is widely used for various regression tasks, such as stock price prediction, weather forecasting, and medical diagnosis.

The Random Forest Regressor creates an ensemble of decision trees that operate independently to generate the final output. It uses a technique called bootstrapping to randomly select samples from the training data to construct each decision tree. Additionally, at each split point in each decision tree, a random subset of the available features is considered to find the best split. This process is called feature bagging. Once the ensemble of decision trees is created, the Random Forest Regressor averages the predictions from all trees to generate the final output.

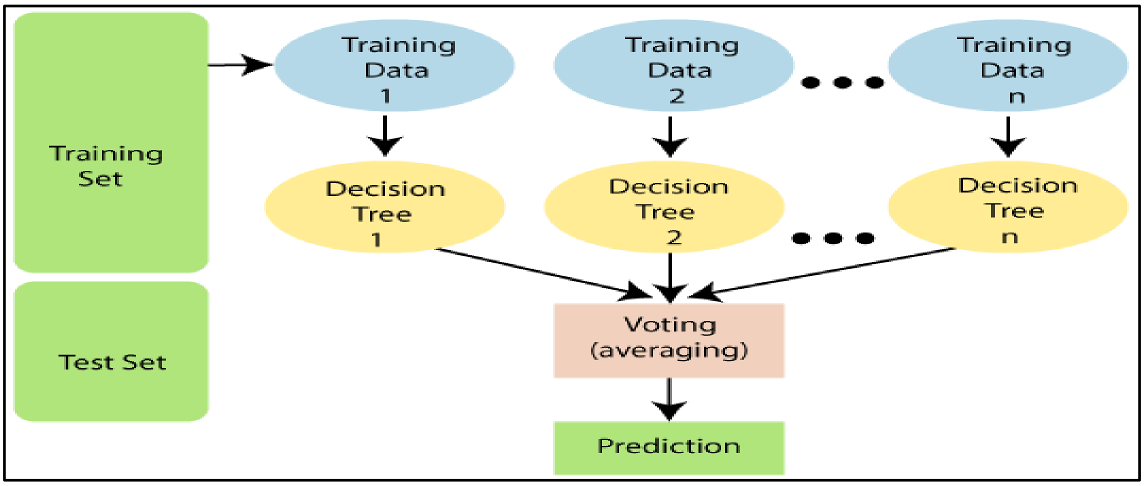

Fig 1.3.4.1.2 Random Forest

1.3.4.3 XGBoost

XGBoost works by building an ensemble of decision trees, where each tree is trained to make predictions based on a subset of the available data. The trees are grown sequentially, with each tree learning from the mistakes of the previous tree. The final prediction is made by taking the average of the predictions from all of the trees in the ensemble. One of the key advantages of XGBoost is its ability to handle missing data and large datasets efficiently.

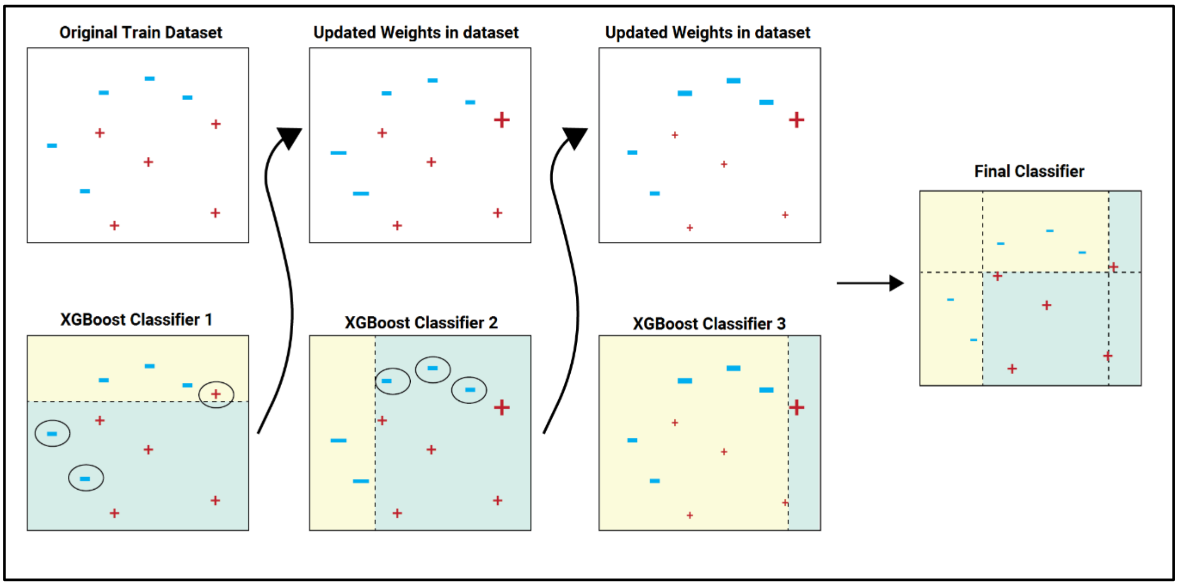

Fig 1.3.4.1.3 XGBoost

1.3.4.4 DT-RF-XGBoost Ensemble Model

The idea is combining many weak learners into strong learner The proposed ensemble model in project is boosting ensemble model Boosting : It is the ensemble model in which the models are fit and added to ensemble sequentially such that the second model will try to correct the predictions of first model, and third will correct the predictions of second model. In the proposed ensemble model , decision tree, random forest, XGBoost algorithms are executed sequentially giving the output predicted values of first model as the input to train the second model by adding these predicted values as the new feature to the dataset.

Here, DT is first trained with the dataset, and the training dataset is modified by considering the predicted values as the new column in the training dataset. The new training dataset is used to train the RF model, now the new predicted values using RF are considered as another new column in the training dataset. The final modified dataset is used to train the XGBoost model, the final predicted values are noted as the values predicted by the proposed ensemble model.

The decision tree is capable of handling both numeric and categorical data and work well with multi output problems, the limitations of DT model lie in its over fitting and unstable nature which is taken care by the RF. RF works well with large datasets and provide high accuracy. The limitations associated with the RF is training time, which is handled by the XGBoost model which provides direct root to minimum error and lower the computational cost.In this way the overall ensemble model will handle the limitations of each other and making it as strong learner.

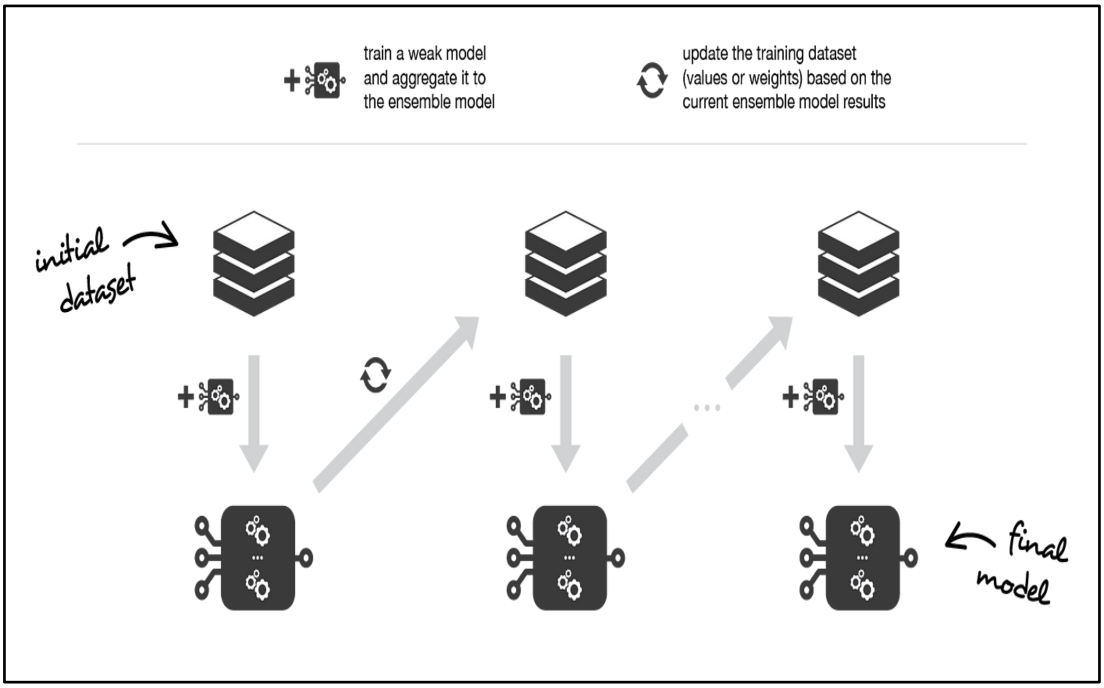

Fig 1.3.4.1.4 Ensemble Model

.

1.3.5 EVALUATION:

The trained model is then tested with the testing data and different evaluation metrics are calculated i.e., mean absolute error, mean squared error, R2 Score, root mean squared error.

The Ensemble Model performed well almost in all cases when compared to the other models.

Evaluation Metrics:

1. Mean Square Error: It is the average of square of the difference between the actual and the predicted values.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_{i-}\hat{y}_i)^2$$

where , y_i = actual value of the target of ith record,
$\hat{y}_i$ = predicted value of the target of ith record
N = total no of records

2. R-Squared Score: It is the statistical measure which shows how the predicted values are close to the actual values

$$R^2 = 1 - \frac{RSS}{TSS}$$

Where , $R^2 = coefficient\ of\ determination$
RSS= sum of squares of residuals
TSS=total sum of squares

3. Root Mean Square Error: Applying square root to the Mean Square Error gives the Root Mean Square.

$$RMSE = \sqrt{(\sum_{i=1}^{N}\frac{(Predicted_i - Actual_i)^2}{N}}$$

Where, N = total no of records,
$Predicted_i$= predicted value of the ith record target
$Actual_i$ = actual value of the ith record target

4. Mean Absolute Error: It is the mean of the absolute difference between the predicted value and the actual value

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_{i-}\hat{y}_i|$$

CHAPTER 2

MERITS AND DEMERITS OF THE BASE PAPER

2.1 MERITS :

- The base paper gave the detailed explanation of the features in the datasets
- It has the detailed explanation of ensemble model and its working principle
- The comparison of performance of different machine learning models was good.
- The findings are relevant and applicable to real world energy prediction because they make use of dataset of real world from a retail business.
- The energy prediction accuracy is improved with the Ensemble models. Because it make use of decision trees , random forests, regression which are supervised machine learning algorithms.
- It provides practical insights into the data pre-processing , feature selection which are applications of machine learning.

2.2 DEMERITS :

- The base paper didn’t given any details about the feature selection algorithm about the selection of the features that are efficient in predicting the energy consumption.
- It only focuses on the comparison of the three different algorithms with the Ensemble model that uses the three algorithms as models
- It does not focus on the individual efficiency of the algorithms.

CHAPTER 3

SOURCE CODE

3.1 DATA PRE-PROCESSING

#importing libraries

import numpy as np

import pandas as pd

import matplotlib

import matplotlib.pyplot as plt

import seaborn as sns ***# for data visualization***

 # copy the dataset to modify

da = df.copy()

da.head()

da.tail()

da = da[0:-1] ***# removing the last invalid tuple***

 # column names cleaned by removing ' [kW]'

da.columns = [col.replace(' [kW]','') for col in da.columns]

 # it is better to have an aggregated result

```
# so let's sum up the Furnace 1 and Furnace 2 to 'FurnaceSum'

# and Kitchen 12 , Kitchen 14 and Kitchen 38 to 'KitchenSum'

# so new features FurnaceSum and KitchenSum are added to dataset

da['FurnaceSum'] = da[['Furnace 1','Furnace 2']].sum(axis=1)

da['KitchenSum'] = da[['Kitchen 12','Kitchen 14','Kitchen 38']].sum(axis=1)

# let's keep the aggregated features 'FurnaceSum' and 'KitchenSum' in dataset

# and drop the old features

da = da.drop(['Furnace 1','Furnace 2'],axis=1)

da = da.drop(['Kitchen 12','Kitchen 14','Kitchen 38'],axis=1)

db = da.copy()

# then use and houseoverall are same

db = db.drop(['use'],axis=1)

# there are columns 'summary', 'icon' and 'cloudCover' which are with different datatypes

db['summary'].unique

db= db.drop(['summary'],axis=1)

db['icon'].unique

# icon is also not numeric type

db = db.drop(['icon'],axis=1)

db['cloudCover'].unique()

# 'cloudCover' seems to be numeric type, but there is an invalid value cloudCover

# instead of removing the column here we can replace the value with next valid observation

# bfill method replaces the NULL values with the values from the next row

db['cloudCover'].replace(['cloudCover'], method='bfill', inplace=True)

db['cloudCover'] = db['cloudCover'].astype('float')

# all features are of same datatype

# lets check for null values
```

db.isnull().sum()

 # it is observed that there are no null values in the dataset

#copying the dataset

dc=db.copy()

dc.info()

3.2 TRAIN-TEST SPLIT

x = dc[['temperature', 'humidity', 'visibility', 'apparentTemperature', 'pressure', 'windSpeed', 'cloudCover', 'windBearing', 'precipIntensity','dewPoint','precipProbability']]

house = dc['House overall'] ***# for House overall***

home = dc['Home office'] ***# for Home office***

 # splitting the dataset to 80-20 for House Overall

from pandas.core.common import random_state

from sklearn.model_selection import train_test_split

 # splitting the datasets to train test for house overall

x_train,x_test,house_train,house_test = train_test_split (x, house,test_size=0.2, random_state=2)

3.3 MACHINE LEARNING TECHNIQUES

3.3.1 Random Forest

from sklearn.ensemble import RandomForestRegressor

RF_reg = RandomForestRegressor(n_estimators = 10,random_state=0)

RF_reg.fit(x_train,house_train)

 # predictiong 'House overall' energy consumption using Random forest

house_pred_RF = RF_reg.predict(x_test)

3.3.2 Decision Tree

```
from sklearn.tree import DecisionTreeRegressor

from sklearn.model_selection import train_test_split

import pandas as pd

dec_tree_reg = DecisionTreeRegressor(random_state=0)

dec_tree_reg.fit(x_train,house_train)

house_pred_DT = dec_tree_reg.predict(x_test)
```

3.3.3 XGBoost Regression :

```
import xgboost as xgb

from sklearn.metrics import mean_squared_error

xg_reg = xgb.XGBRegressor(objective ='reg:squarederror',colsample_bytree = 0.3, learning_rate =
0.1,max_depth = 5, alpha = 10, n_estimators = 100 )

xg_reg.fit(x_train,house_train)

house_pred_xg=xg_reg.predict(x_test)
```

3.3.4 DT-RF-XGBoost Ensemble Model :

```
from sklearn.tree import DecisionTreeRegressor

from sklearn.ensemble import RandomForestRegressor

import xgboost as xgb

tree_model = DecisionTreeRegressor(random_state=42)

rf_model = RandomForestRegressor(random_state=42)

xgb_model = xgb.XGBRegressor(random_state=42)


#Train decision tree and make predictions on test set

tree_model.fit(x_train, house_train)

tree_preds = tree_model.predict(x_test)
```

#Add decision tree predictions as a new feature to train and test sets

x_train['tree_preds'] = tree_model.predict(x_train)

x_test['tree_preds'] = tree_preds

#Train random forest and make predictions on test set (using decision tree predictions as input feature)

rf_model.fit(x_train, house_train)

rf_preds = rf_model.predict(x_test)

#Add random forest predictions as a new feature to train and test sets

x_train['rf_preds'] = rf_model.predict(x_train)

x_test['rf_preds'] = rf_preds

#Train XGBoost and make predictions on test set (using decision tree and random forest predictions as input features)

xgb_model.fit(x_train, house_train)

final_preds = xgb_model.predict(x_test)

3.4 EVALUATION PARAMETERS

3.4.1 R-square, Mean Square Error, Mean Absolute Error, Root Mean Square Error

from sklearn.metrics import r2_score

from sklearn.metrics import mean_squared_error

from sklearn.metrics import mean_absolute_error

r2=r2_score(house_test,final_preds)

mse = mean_squared_error(house_test, final_preds)

mae = mean_absolute_error(house_test,final_preds)

rmse = np.sqrt(mse)

print("r2 : ",r2)

print("mse : ",mse)

print("mae : ",mae)

print("rmse : ",rmse)

3.5 PLOTS

3.5.1 Scatter Plot

time = house_test.index

import matplotlib.pyplot as plt

 # Create scatter plot

plt.scatter(time,house_test, color='blue', label='Actual',marker=r'$\plus$')

plt.scatter(time,final_preds, color='red', label='Predicted',marker=r'$\star$')

 # Add labels and title

plt.xlabel('Index')

plt.ylabel('Value')

plt.title('Predicted vs Actual')

 # Add legend

plt.legend()

 # Show plot

plt.show()

3.5.2 Bar Graph

import matplotlib.pyplot as plt

 # Assuming you have defined 'house_test' and 'house_pred_DT' with the actual and predicted value

 # Create index for the bars

index = house_test.index

 # Set the width of the bars

bar_width = 0.35

Plot the actual values as blue bars

plt.bar(index, house_test, bar_width, color='blue', label='Actual')

 # Plot the predicted values as red bars shifted by the bar width

plt.bar(index , house_pred_DT, bar_width, color='red', label='Predicted')

 # Add labels and title

plt.xlabel('Index')

plt.ylabel('House overall')

plt.title('Predicted vs Actual House overall')

plt.legend()

 # Show plot

plt.show()

CHAPTER 4

SNAPSHOTS

For HOUSEOVER ALL

4.1 Random Forest

The scatter plot and bar graph are plotted to quantify the predicted outcomes does hold good or not. Plots are plotted with predicted and actual values as Y-axis and index(time) as X-axis when random forest algorithm has been implemented for predicting house overall energy consumption.

From the scatter plot we can infer that the scattering of predicted and actual values is more, actual values are given in blue and predicted are given in red. From bar graph the predicted values are very slightly matched with the actual values.

These values are evaluated with some evaluation metrics and the results are as follows mean square error(MSE) is ,mean absolute error(MAE) is ,root mean square error(RMSE) is ,R-squared error(RSE) is

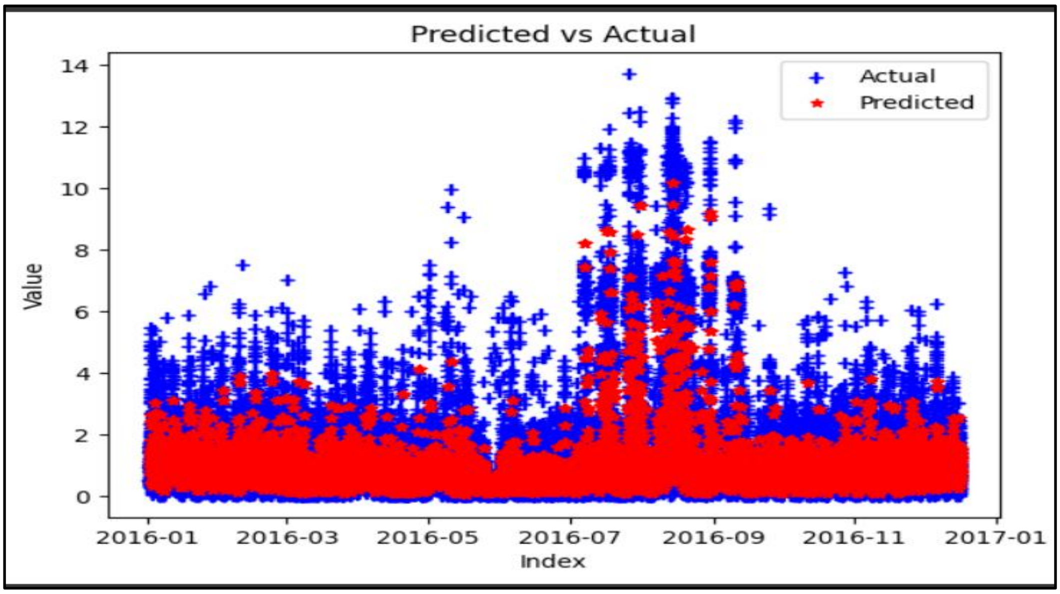

Fig 4.1.1 Scatter Plot for Random Forest

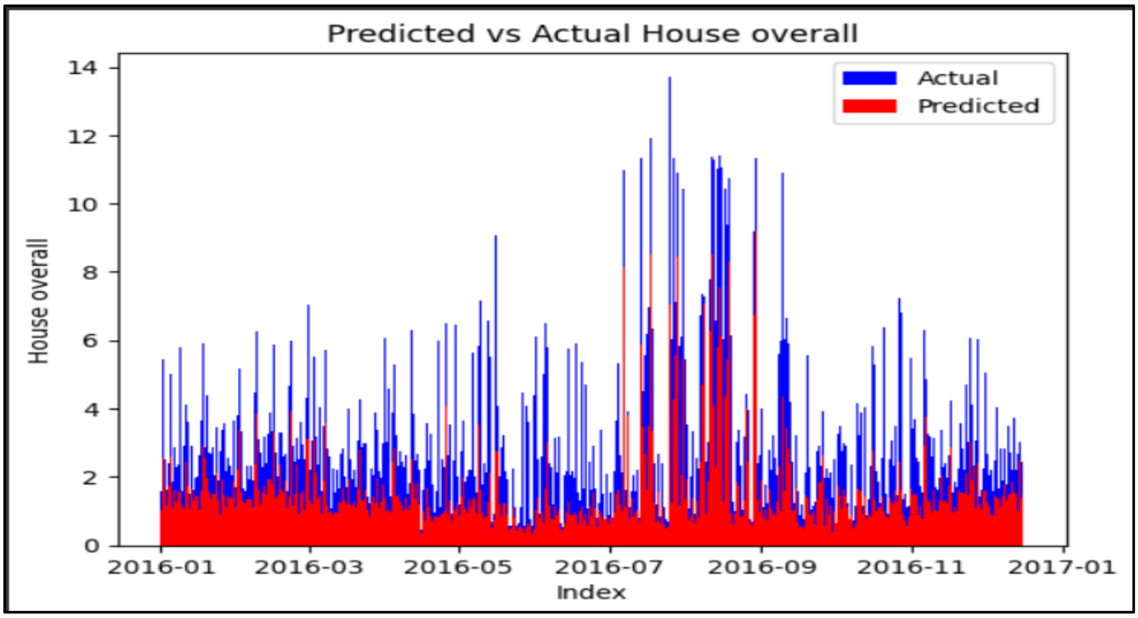

Fig 4.1.2 Bar Graph for Random Forest

4.2 Decision Tree

Here the actual and predicted values of house overall energy consumption after using decision tree algorithm has been taken and plotted with time as index. The house overall energy consumption values are taken in Y-axis and time is taken in X-axis.The plotted results are shown in Fig 4.2.1 and 4.2.2

From the scatter plot we can infer that the predicted and actual values are more scattered, actual values are given in blue and predicted are given in red. From bar graph the predicted values are slightly matched with the actual values.

These values are evaluated with some evaluation metrics and the results are as follows mean square error(MSE) is ,mean absolute error(MAE) is ,root mean square error(RMSE) is ,R-squared error(RSE) is

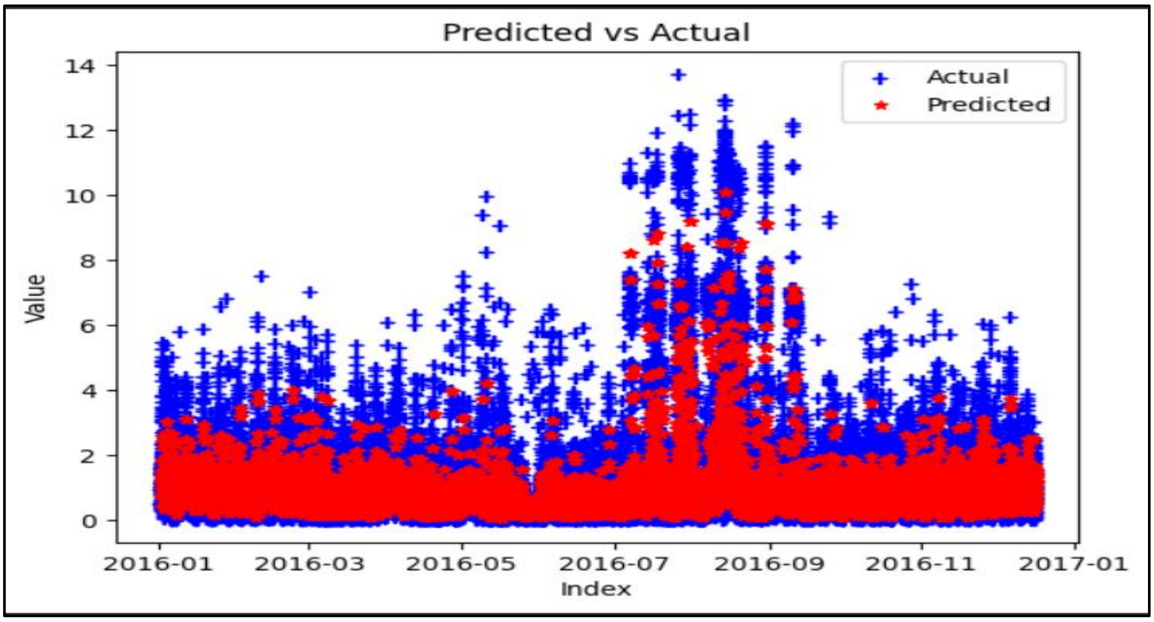

Fig 4.2.1 Scatter Plot for Decision Tree

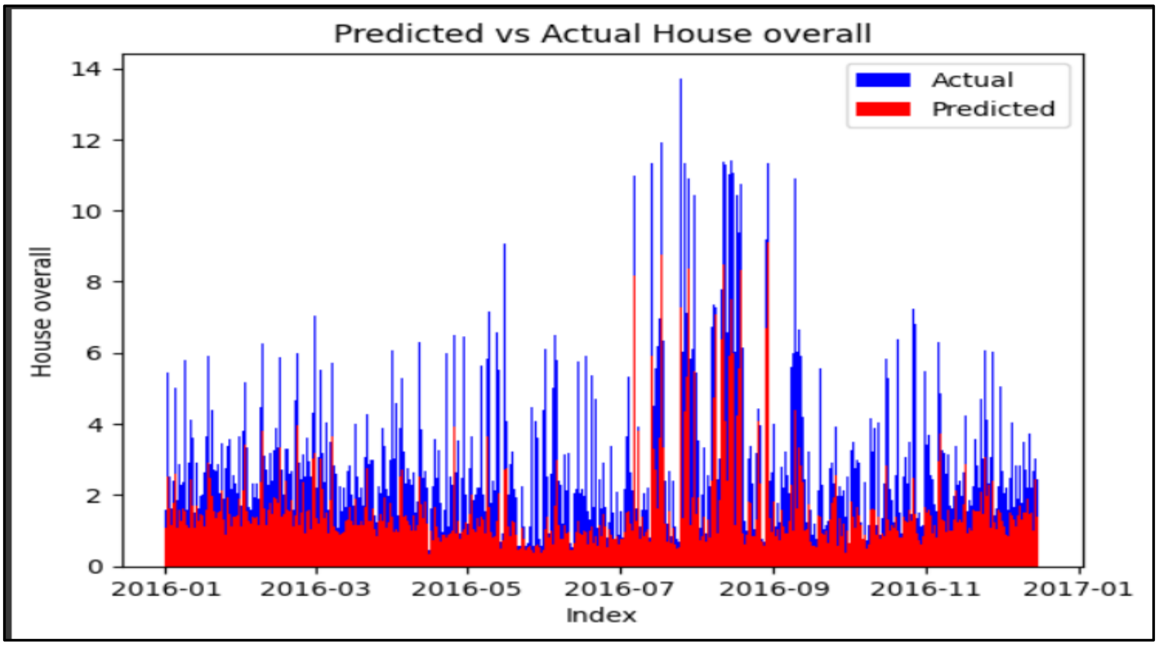

Fig 4.2.2 Bar Graph for Decision Tree

4.3 XGBoost

Here the actual and predicted values of house overall energy consumption after using XGboost algorithm has been taken and plotted with time as index. The house overall energy consumption values are taken in Y-axis and time is taken in X-axis.The plotted results are shown in Fig 4.3.1 and 4.3.2

From the scatter plot we can infer that the predicted and actual values are more scattered, actual values are given in blue and predicted are given in red. From bar graph the predicted values are slightly matched with the actual values.

These values are evaluated with some evaluation metrics and the results are as follows mean square error(MSE) is ,mean absolute error(MAE) is ,root mean square error(RMSE) is ,R-squared error(RSE) is

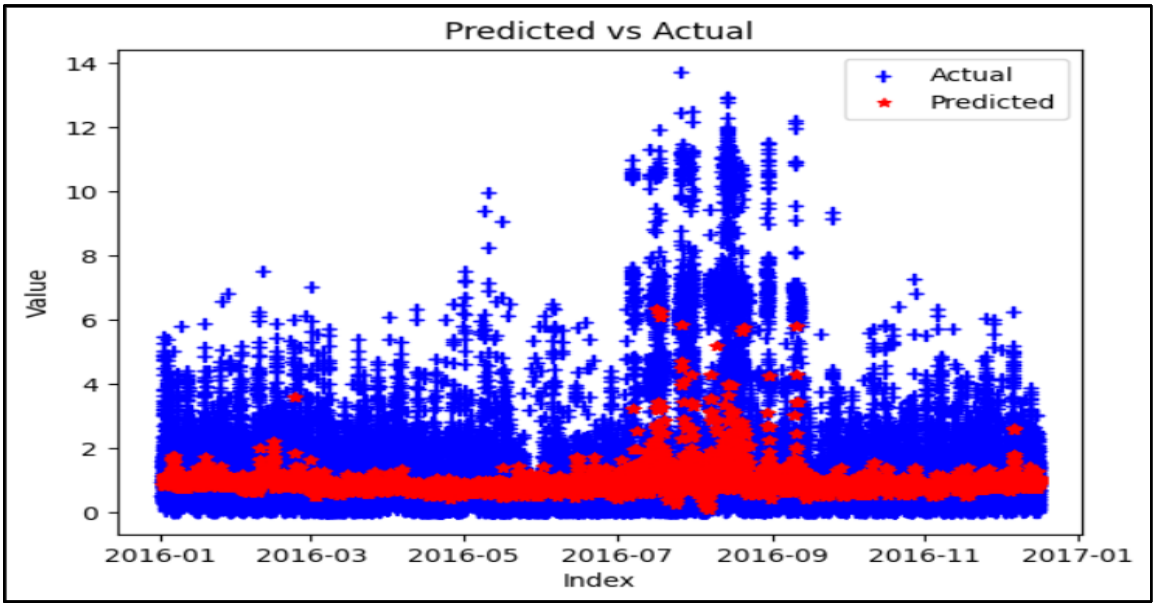

Fig 4.3.1 Scatter Plot for XGBoost

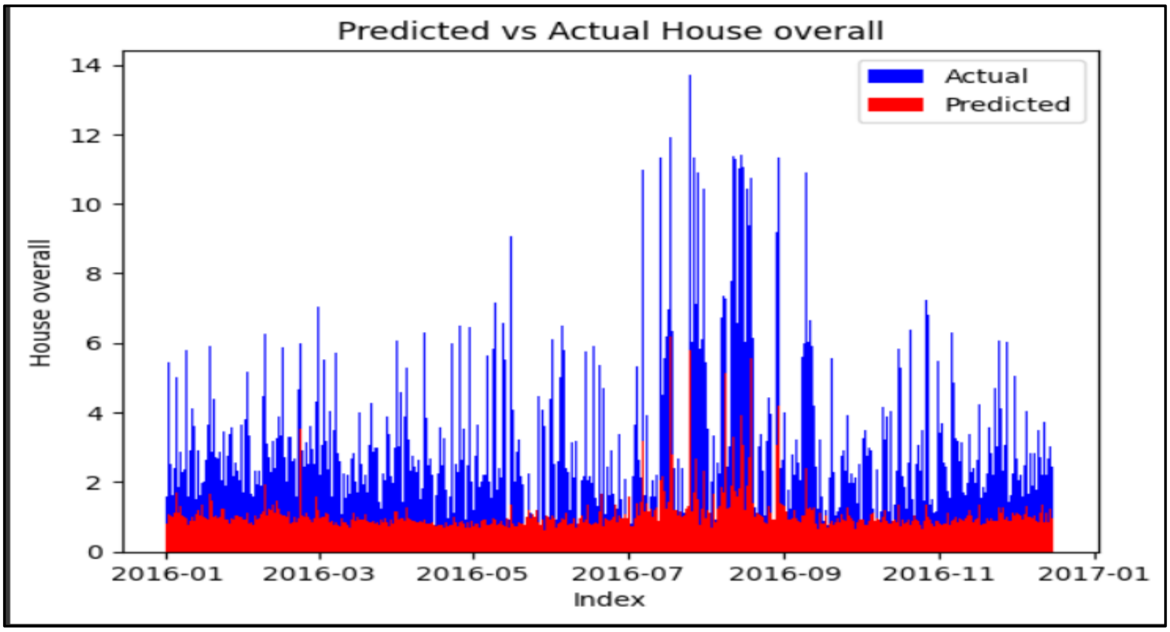

Fig 4.3.2 Bar Graph for XGBoost

4.4 Ensemble Model

Here the actual and predicted values of house overall energy consumption after using DT-RF-XGBoost an ensemble algorithm has been taken and plotted with time as index. The house overall energy consumption values are taken in Y-axis and time is taken in X-axis.The plotted results are shown in Fig 4.4.1 and 4.4.2

From the scatter plot we can infer that the predicted and actual values are less scattered, actual values are given in blue and predicted are given in red. From bar graph the predicted values are vastly matched with the actual values than compared with the other algorithms.

These values are evaluated with some evaluation metrics and the results are as follows mean square error(MSE) is ,mean absolute error(MAE) is ,root mean square error(RMSE) is ,R-squared error(RSE) is

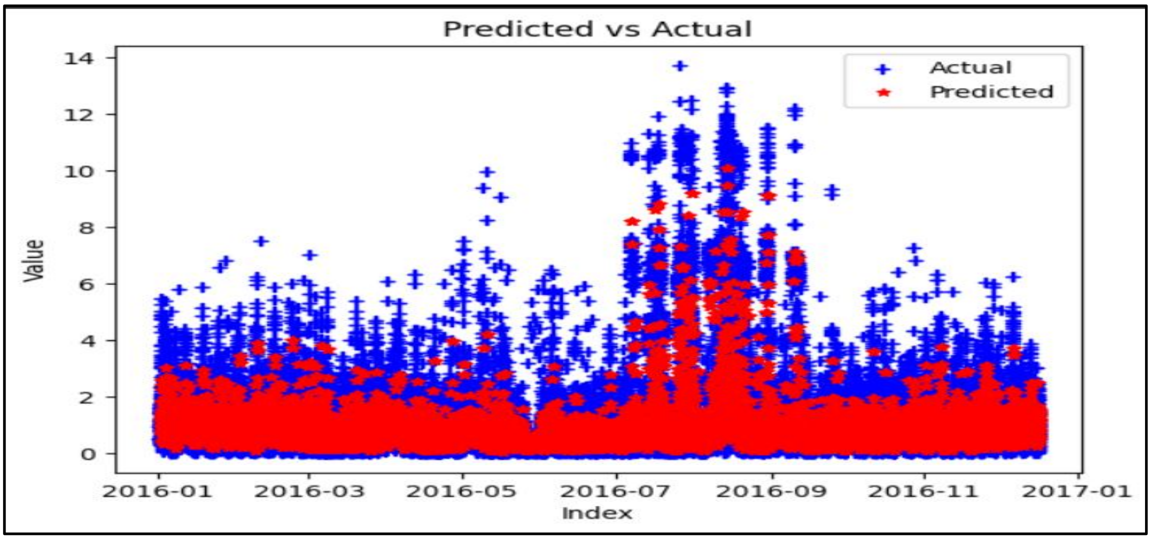

Fig 4.4.1 Scatter Plot for Ensemble Model

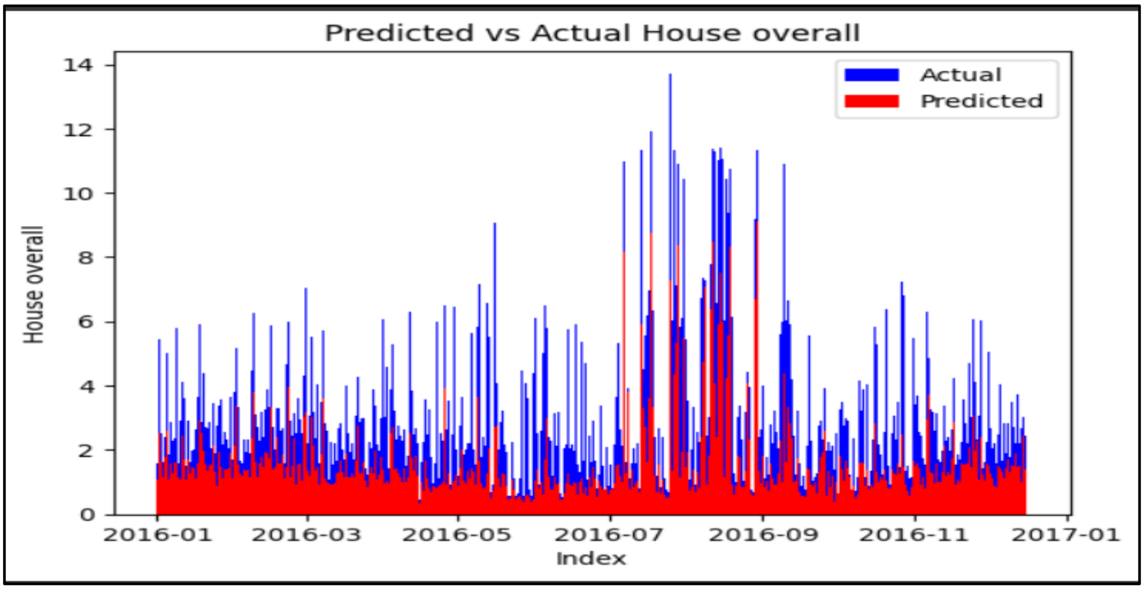

Fig 4.4.2 Bar Graph for Ensemble Model

ML Algorithm	MSE	RSE	MAE	RMSE
Random Forest	0.501789440386	0.5540277207376	0.345114989024	0.708370976527
Decision Tree	0.500842372177	0.5548694406576	0.3450486837756	0.707702177598

XGBoost	0.859823173947	0.2358203067058	0.521364103645	0.927266506435
Ensemble model	0.500836985076	0.5548742285180	0.345042054335	0.707698371537

Table 4.5 Comparison of four models using four metrics

Experimental Results:

The above scatter plots and bar plots from Fig 4.1.1 to 4.4.2 shows the plots between the time on X-axis and house overall consumption[Kw] on Y axis and the blue colour represents the actual values and the red colour represents the predicted value of the energy consumption for the HomeC.csv dataset.

From the scatter plots of Fig 4.1.1,Fig 4.2.1,Fig 4.3.1 the predicted values are scattered more from the original or actual values . We can see that Random Forest and Decision tree scatter plots from Fig 4.1.1 and Fig 4.2.1 are almost similar while the XGBoost scatter plot Fig 4.3.1 shows more deviation from the actual values .Comparing the three above models with the Ensemble model scatter plot from Fig 4.4.1 there is a less deviation from the actual values. So it performed well. So the Ensemble model is better than the above models.

CHAPTER 5

Conclusion and Future Plans

5.1 Conclusion:

Based on the performed study, it can be concluded that predicting energy consumption using more than one machine learning algorithms is an effective way to monitor energy usage in smart homes. The DT-RF-XGBoost-based ensemble model performed better than the baseline algorithms in terms of MSE, R2, RMSE, and MAE.

This indicates that the ensemble model has more potential in accurately predicting energy consumption and improving energy management in smarthomes. These models are widely used as this may reduce energy cost, increase energy efficiency and be predictable with climatic changes.These results provide home owners in making informed decisions about energy usage and conservation.

5.2 Future Works:

- Accuracy of the model can be improved by adding neural network models
- To make the process much convenient for more weather conditions in different areas

CHAPTER 6

REFERENCES

[1] Passive sensing and detection of boats along coastline using static sensorsProceedings of 2017 3rd IEEE International Conference on Sensing, Signal Processing and Security, ICSSS 2017 | Conference paper DOI: 10.1109/SSPS.2017.8071643 EID: 2-s2.0-85039923998 CONTRIBUTORS: Kavitha, R.; Thayaambika, M.; Malini, P.V.; Almelu, N.

[2] I. Machorro-Cano, G. Alor-Hernandez, ´ M.A. Paredes-Valverde, L. Rodríguez-Mazahua, J.L. Sanchez-Cervantes, ´ J.O. Olmedo-Aguirre, HEMS-IoT: A big data and machine learning-based smart home system for energy saving, Energies 13 (5) (2020) 1097.

[3] A. Akbari-Dibavar, S. Nojavan, B. Mohammadi-Ivatloo, K. Zare, Smart home energy management using hybrid robust-stochastic optimization, Comput. Ind. Eng. 143 (2020), 106425.

[4] I. Hussain, M. Ullah, I. Ullah, A. Bibi, M. Naeem, M. Singh, D. Singh, Optimizing energy consumption in the home energy management system via a bio-inspired dragonfly algorithm and the genetic algorithm, Electronics 9 (3) (2020) 406.

[5] K.H.N. Bui, I.E. Agbehadji, R. Millham, D. Camacho, J.J. Jung, Distributed artificial bee colony approach for connected appliances in smart home energy management system, Process. Expert Syst., Technol. Value Sugar Beet, Prog. Sugar Technol.: Proc. Gen. Assem. C.I.T.S. , 20th 37 (6) (2020) e12521.

[6] M. Elsisi, M.Q. Tran, K. Mahmoud, M. Lehtonen, M.M. Darwish, Deep learning-based industry 4.0 and Internet of Things towards effective energy management for smart buildings, Sensors 21 (4) (2021) 1038.