KNN APPROACHES BY USING BALL TREE SEARCHING ALGORITHM WITH MINKOWSKI DISTANCE FUNCTION ON SMART GRID DATA

Dr.Belwin J Brearley[1], Dr.K. Regin Bose[2], Dr.K.Senthil[3], Dr.G.Ayyappan[4]

Assistant Professor (SG), Department of Electrical and Electronics Engineering, B.S Abdur Rahman Crescent Institute of Science and Technology, Chennai, India[1]

Professor, Department of CSE, Prince Dr. K. Vasudevan College of Engineering and Technology, Chennai[2]

Associate Professor, Department of CSE, Prince Shri Venkateshwara Padmavathy Engineering College, Chennai[3]

Associate Professor, Department of IT, Prince Shri Venkateshwara Padmavathy Engineering College, Chennai[4]

belwinbrearley@gmail.com[1], reginbosek@gmail.com[2], k.geethapriyan@gmail.com[3] , ayyappangmca@gmail.com[4]

Abstract:

The smart grid is a critical infrastructure area, so machine learning models involving it must be interpretable in order to increase user trust and improve system reliability. This research work concludes that the highest accuracy value is 88.02% of accuracy level which is acquired by n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The highest positive predictive value is 0.87 which is owned by n-neighbors=5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The maximum as well same Sensitivity value is 0.88 which yielded by n-neighbors=3 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, n-neighbors=5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. n-neighbors=5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The highest F-Measure value is 0.86 is produced by n-neighbors=3 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, n-neighbors=5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, n-neighbors=7 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The supreme value of Mean Square Contingency Coefficient shows that 0.45 which is given by n-neighbors=5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The maximum amount of kappa value is 0.42 which is picked by n-neighbors=3 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The highest ROC value is 0.80 which is given by n-neighbors =9 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. Highest PRC value is 0.87 which is given by there are two models. They are n-neighbors=7 and K-9 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function models. The Least Mean absolute deviation is 0.16 which is given by is shown by n-neighbors=1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The least root mean squared deviation is 0.31 which are demonstrated by n-neighbors=5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The smallest relative absolute error performance is 61.66 %which is hold by n-neighbors=1 with 1.0 of default radius,30 of leaf size,

minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The largest root relative squared error performance is 110.98 % which is hold by n-neighbors=1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. This research work recommended that the n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model due to its performance is best compare with other models with low deviations.

Keywords: Ball Tree, batch size , KNN, minkowski distance function, smart grid

I Introduction

The smart grid develops the traditional power grid with progressive extent and detecting, ICT, simulation analysis and control decision-making systems [3–5]. The smart grid has more advantages in self-healing, renewable energy consumption, situational awareness, information interaction and stability [5,6]. The smart grid is gradually becoming a power cyber-physical system that closely integrates measurement, communication and various external systems (such as weather, market, etc.) [7,8]. It continues to generate multi-source heterogeneous data with high-dimensional. The emergence of huge data can be provided data support for the study of smart grid issues, but also bring a new opportunity to smart grid management. [9]. Machine Learning plays a major role in promoting the development of AI technology. The application of Machine Learning technology in the smart grid is regarded as one of the leading technologies in the development of the power industry.

This paper organizes section 2 focuses on literature reviews of this work, section 3 focuses on the materials and methods, section 4 focuses on the results and discussions and finally section 5 focuses on the conclusions.

II Literature Survey

Artificial intelligence (AI) technology, which can improve the efficiency and accuracy of decision-making, is an important means to support the smart grid [10]. Machine Learning algorithms use few assumptions and a lot of computing power to mine complex relationships of history data [11]. The use of ML algorithms can form input–output relationship mapping for complex mechanisms in the smart grid, thereby breaking through the limitations of existing physical knowledge, so it is very suitable for dealing with the challenges of the smart grid. Commonly used ML algorithms include linear regression (LR) [12], SVM learning model [13], KNN learning models, [14], clustering learning models [15], decision tree learning model (DT) [16], meta learning [17], Artificial Neural Networks machine learning [18], etc. They are currently used to address many related issues in the smart grid [19], accurate prediction of distributed energy resources [20], and stability analysis of complex power grids [21]. In recent years, with the development of computing power, deep learning which is a special kind of ML is emerging.[23-24]

III Materials and Methods

This section presents that the materials and methods of this research work. The breast cancer dataset downloaded from UCI repository.[22] It contains 10000 instances and 13 attributes with binary class. The following table shows description of the dataset.

Table 1: Meta data of dataset

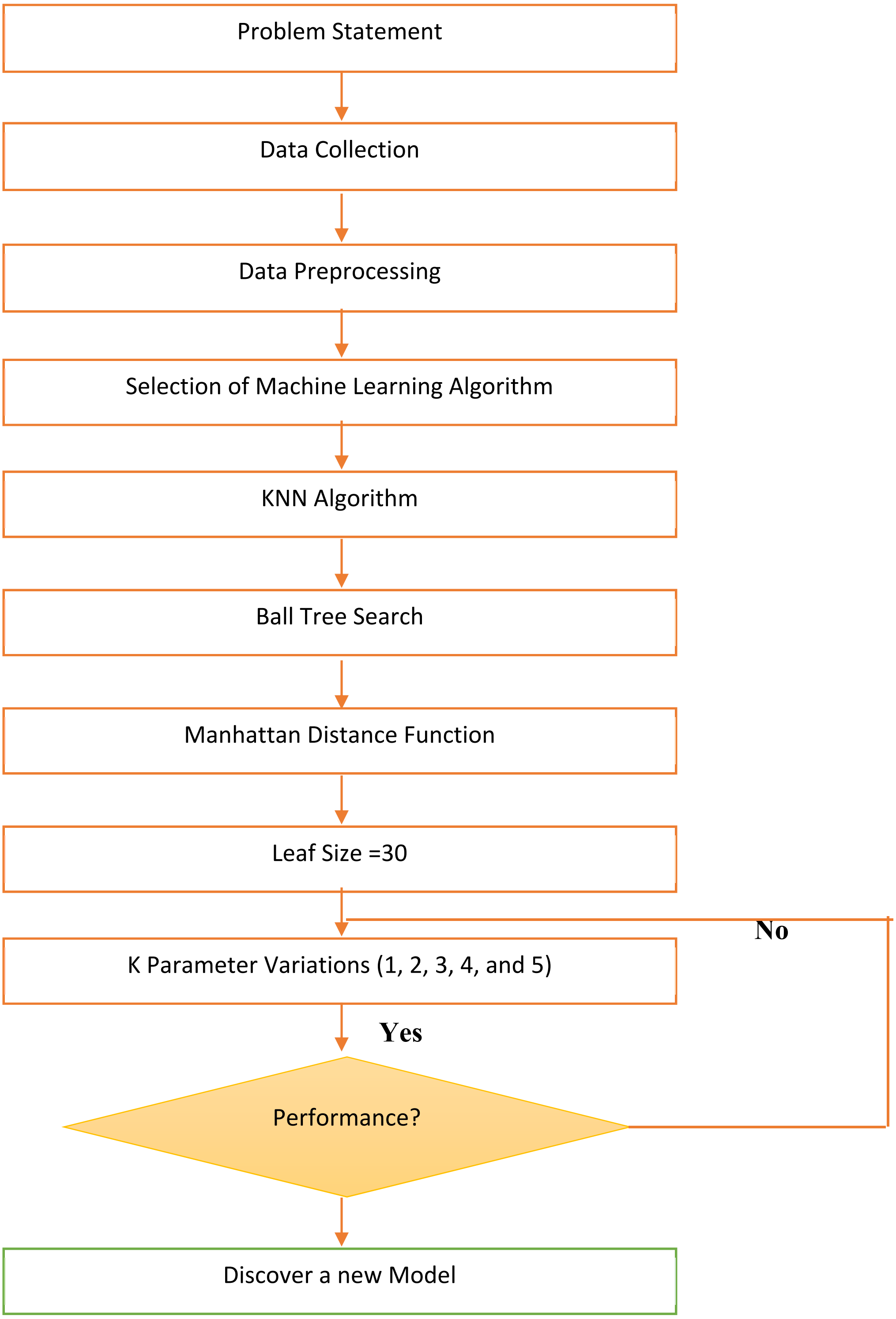
S.No	Attribute	Description	Data type
1	tau[x]	Reaction time of participant	Real
2	p[x]	nominal power consumed	Real
3	g[x]	coefficient (gamma) proportional to price elasticity	Real
4	stab	the maximal real part of the characteristic equation root	Real
5	stabf	the stability label of the system	Text

The borrowed dataset split up on 2:8 cross validation and it is processed by python library.

Proposed Method

1. Data Collection
2. Data Preprocessing

3. Apply KNN Algorithm with parameter are n-neighbor value, with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model.
4. Model Evaluation



This section shows that the results and discussions of this research work. This work focuses and selected K nearest neighborhood algorithm of lazy category by using with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. When varying the k parameter give the several outcomes.

Table 2: Performance measurements of Lazy Classifier

S.No	Classifier	n-neighbors	Accuracy	Positive predictive value	Sensitivity	Time (Sec)
1	K NN	1	84.02%	0.83	0.84	0.01
2		2	87.48%	0.86	0.88	0.00
3		3	88.02%	0.87	0.88	0.00
4		4	87.99%	0.87	0.88	0.01
5		5	87.72%	0.87	0.88	0.00

The table 2 shows that the performance levels of KNN models by applying various n-neighbors with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 84.02% of accuracy level, 0.83 of positive predictive value level, 0.84 of Sensitivity level and it takes to build only 0.01 seconds are shown by n-neighbors =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 87.48% of accuracy level, 0.86 of positive predictive value level, 0.88 of Sensitivity level and it takes to build only zero seconds are shown by n-neighbors=3 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 88.02% of accuracy level, 0.87 of positive predictive value level, 0.88 of Sensitivity level and it takes to build only zero seconds are shown by n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. shows 87.99% of accuracy level, 0.87 of positive predictive value level, 0.88 of Sensitivity level and it takes to build only 0.01 seconds are shown by n-neighbors =7 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 87.72% of accuracy level, 0.87 of positive predictive value level, 0.88 of Sensitivity level and it takes to build only zero seconds are shown by n-neighbors =9 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model.

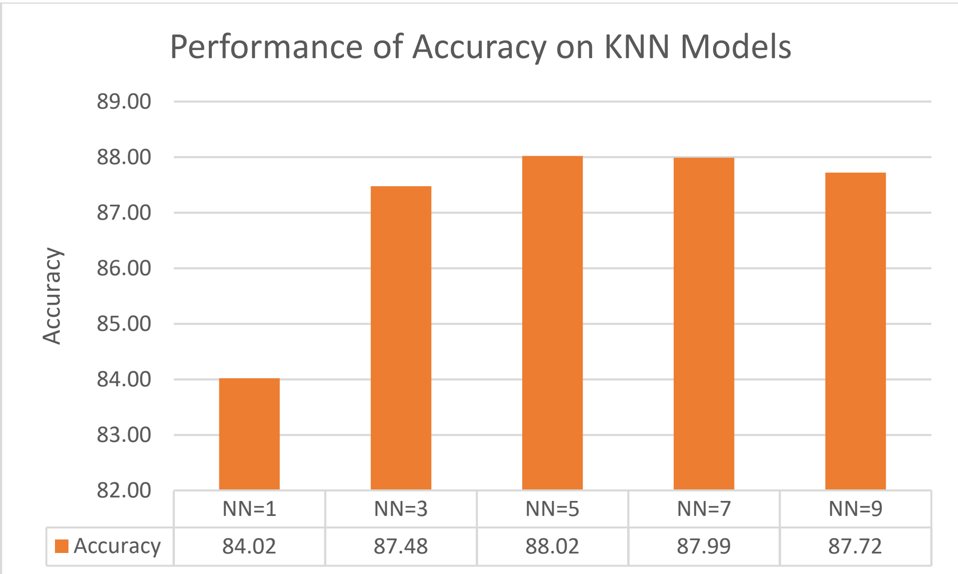


Figure2: Accuracy performance of KNN Models

The above diagram 2 shows that the measurements of accuracy performance of KNN Models by applying various n-neighbors with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 84.02% of accuracy level is lowest accuracy model compare with other models which is produced by n-neighbors =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The highest accuracy value is 88.02% of accuracy level which is acquired by n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. n-neighbors =3 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, n-neighbors =4 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model and n-neighbors =9 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model show that the 87.48% of accuracy, 87.99% of accuracy and 87.72% of accuracy level respectively.

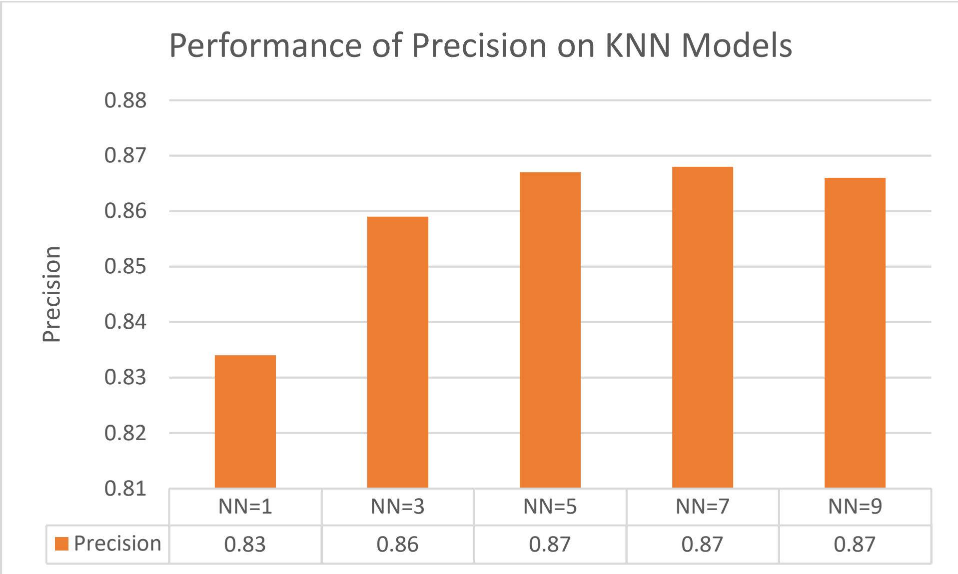


Figure 3: Positive predictive value performance of KNN Models

The above diagram 3 shows that the precision measurements of applying various n-neighbors with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 0.83 of positive predictive value level is the lowest accuracy model compare with other which is opted by n-neighbors =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The highest positive predictive value value is 0.87 which is owned by n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, n-neighbors =7 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model and n-neighbors =9 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. Positive predictive value value 0.86 is gained by n-neighbors =3 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model.

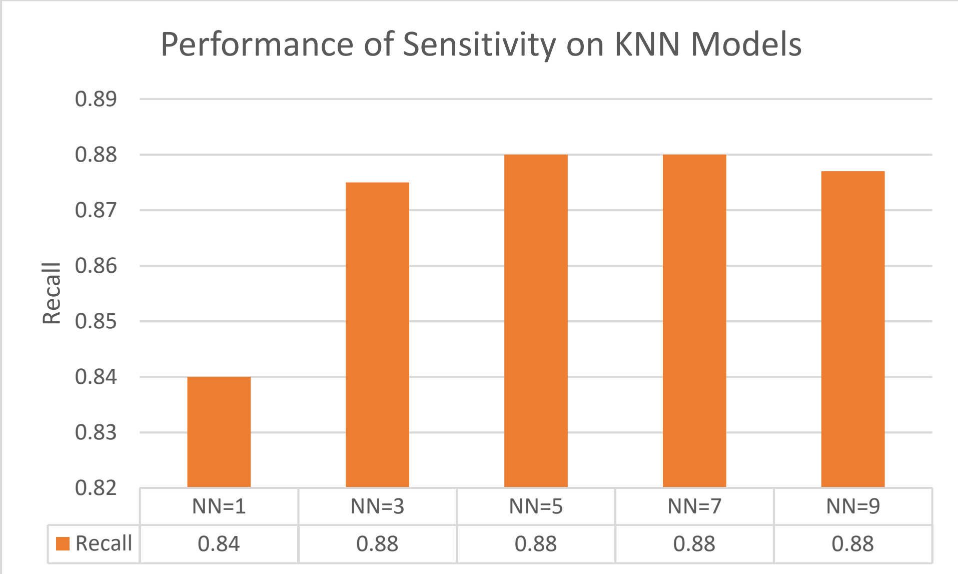


Figure 4: Sensitivity performance of KNN Models

The above picture 4 shows that the measurements of Sensitivity value on KNN models by applying various n-neighbors with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 0.84 of Sensitivity level is minimum Sensitivity value model compare with other models which is acquired by various n-neighbors =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The maximum as well same Sensitivity value is 0.88 which yielded by n-neighbors =3 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, N_NEIGHBORS=7 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model and n-neighbors =9 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model.

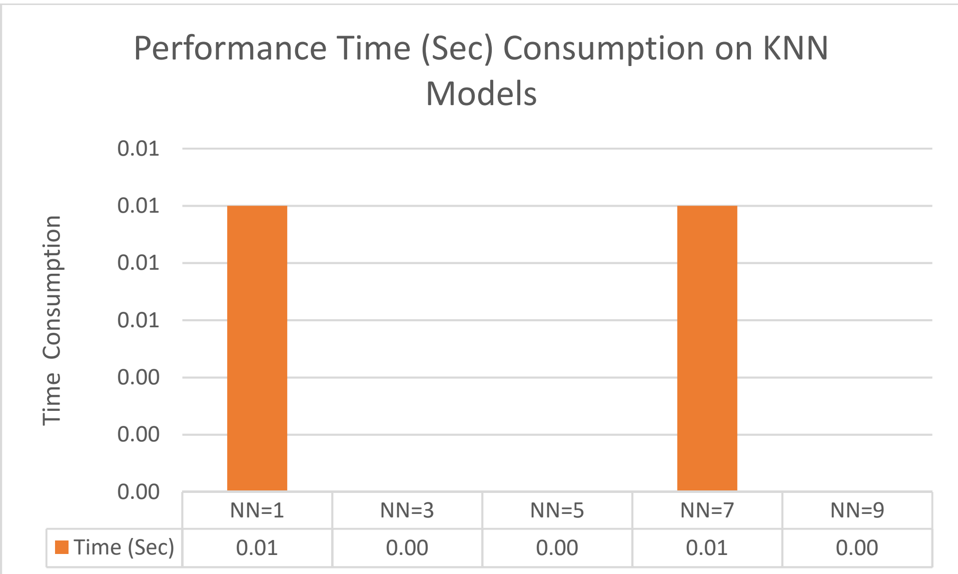


Figure 5: Time consumption performance of KNN Models

The above picture 5 shows that the measurements of time consumption for making KNN models by applying various n-neighbors with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The least time consumption is zero seconds which are shown by n-neighbors =3 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, and n-neighbors =9 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model for constructing their models. The maximum time consumption is 0.01 seconds which are shown by n-neighbors =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, and n-neighbors =7 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model for making their models.

Table 3: F- Measure, MCC and Kappa performance of Lazy Classifier

S.No	Classifier	n-neighbors	F-Measure	MCC	Kappa Statistic
1	K NN	1	0.84	0.36	0.36
		2	0.86	0.44	0.42
		3	0.86	0.45	0.41
		4	0.86	0.44	0.39
		5	0.85	0.41	0.36

The table 3 shows that the performance levels of KNN model by applying various n-neighbors with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 0.84 of F-Measure, 0.36 of Mean Square Contingency Coefficient value level, 0.36 of kappa statistic value is cropped by n-neighbors =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 0.86 of F-Measure, 0.44 of Mean Square Contingency Coefficient value level, 0.42 of kappa statistic value is cropped by n-neighbors =3 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 0.86 of F-Measure, 0.45 of Mean Square Contingency Coefficient value level, 0.41 of kappa statistic value is cropped by N n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 0.86 of F-Measure, 0.44 of Mean Square Contingency Coefficient value level, 0.39 of kappa statistic value is cropped by n-neighbors =7 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 0.85 of F-Measure, 0.41 of Mean Square Contingency Coefficient value level, 0.36 of kappa statistic value is cropped by n-neighbors =9 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model.

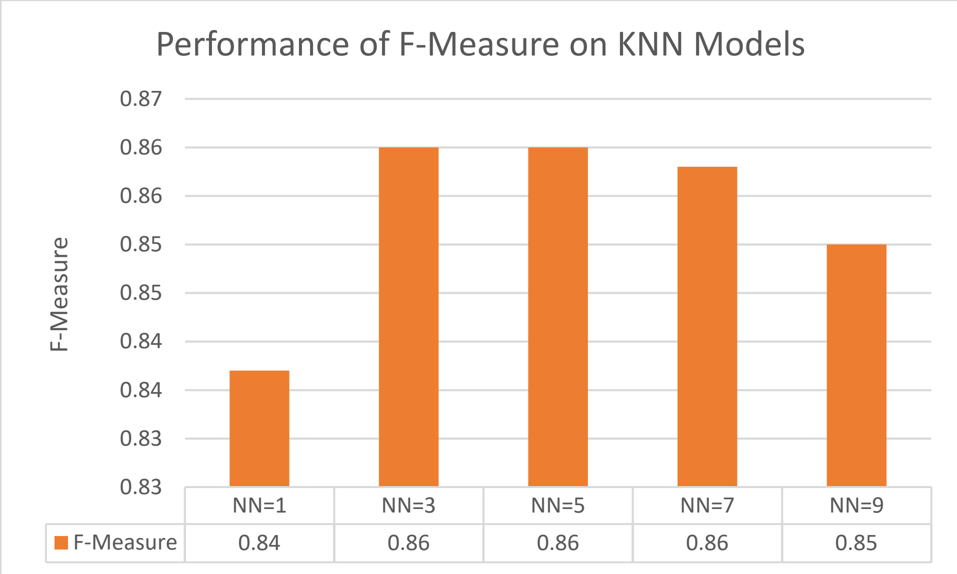


Figure 6: F-Measure performance of KNN Models

The above picture 4 shows that the measurements of F-Measure value on KNN models by applying various n-neighbors with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 0.84 of F-Measure which is a least F-measure value compare with other model which is given by n-neighbors =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The highest F-Measure value is 0.86 is produced by N n-neighbors 3 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, n-neighbors =7 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, and n-neighbors =9 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model.

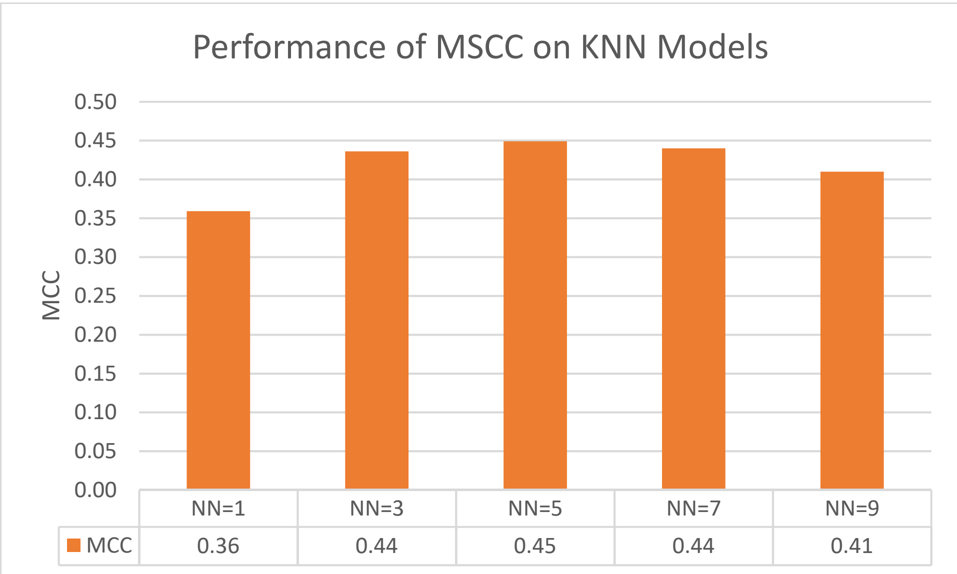


Figure 7: MSCC performance of KNN Models

The above picture 7 shows that the measurements of tiniest Mean Square Contingency Coefficient value on KNN models by applying various n-neighbors with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 0.36 of Mean Square Contingency Coefficient value is a tiniest Mean Square Contingency Coefficient value compare with other models which is picked by n-neighbors =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 0.44 of Mean Square Contingency Coefficient value are picked by there are two models. They are n-neighbors =3 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model and by n-neighbors =7 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model and supreme value of Mean Square Contingency Coefficientshows that 0.45 which is given by n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. n-neighbors =9 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model is showing 0.41 of Mean Square Contingency Coefficient value.

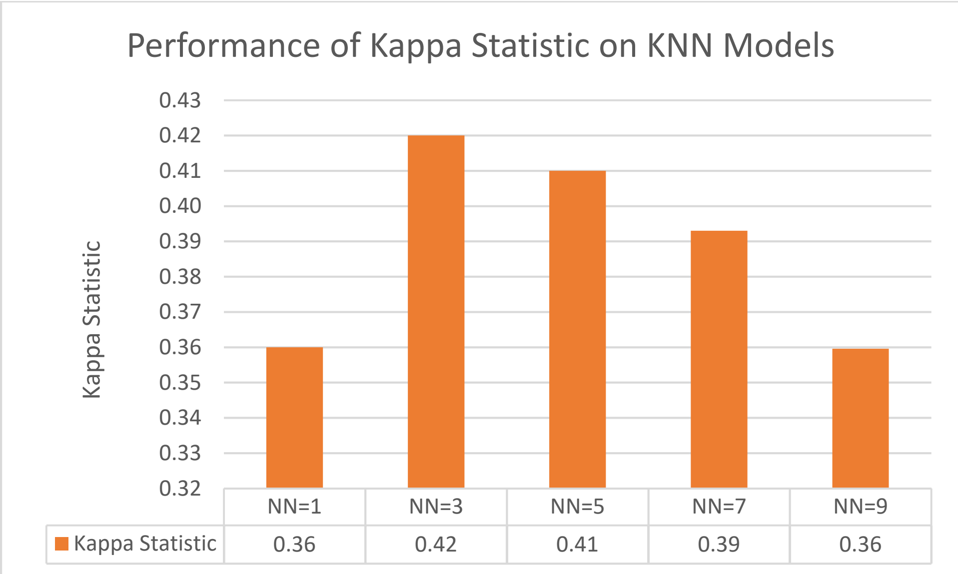


Figure 8: Kappa performance of KNN Models

The above picture 8 shows that the measurements of kappa value on KNN models when by applying various n-neighbors with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The smallest amount of kappa value is 0.36 which is picked by n-neighbors =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The maximum amount of kappa value is 0.42 which is picked by n-neighbors =3 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The rest of models n-neighbors =9 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, n-neighbors =7 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model and n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model are having 0.36 of kappa value, 0.39 of kappa value and 0.41 of kappa value respectively.

Table 4: ROC and PRC Performance of Lazy Classifier

S.No	Classifier	n-neighbors	ROC	PRC
1	K NN	1	0.67	0.81
		2	0.74	0.84
		3	0.78	0.86
		4	0.78	0.87
		5	0.80	0.87

The above table 4 shows that the measurements of Receiver Operating Characteristic value and Positive predictive value Sensitivity Curve value on KNN models by applying various n-neighbors with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 0.67 of Receiver Operating Characteristic level and 0.81 of Positive predictive value Sensitivity Curve level is owned by n-neighbors =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 0.74 of Receiver Operating Characteristic level and 0.84 of Positive predictive value Sensitivity Curve level is owned by n-neighbors =3 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 0.78 of Receiver Operating Characteristic level and 0.86 of Positive predictive value Sensitivity Curve level is owned by N_NEIGHBORS=5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 0.78 of Receiver Operating Characteristic level and 0.87 of Positive predictive value Sensitivity Curve level is owned by n-neighbors =7 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 0.80 of Receiver Operating Characteristic level and 0.87 of Positive predictive value Sensitivity Curve level is owned by n-neighbors =9 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model.

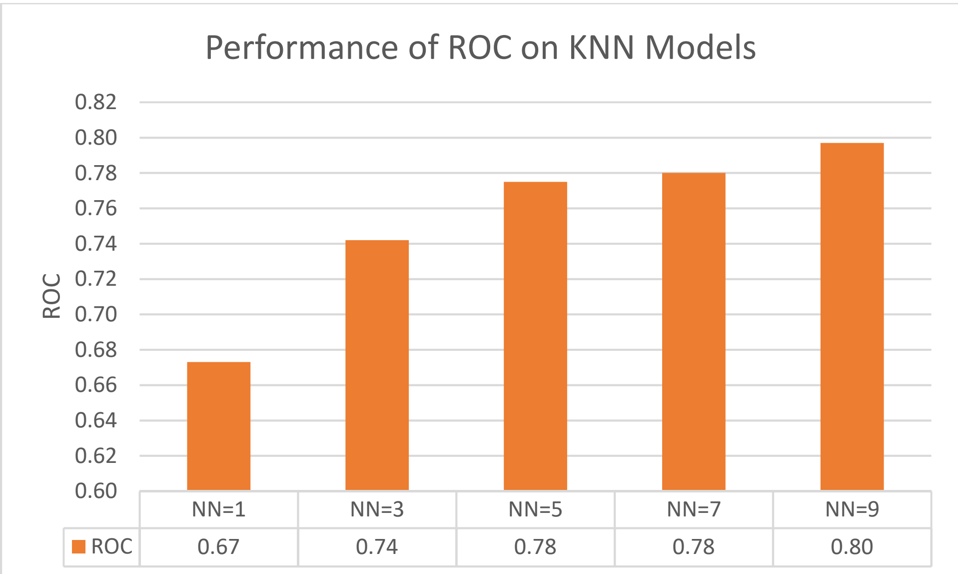


Figure 9: ROC performance of KNN Models

The above picture 9 shows that the measurements of ROC value on KNN models by applying various n-neighbors with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The smallest ROC value is 0.67 which is shown by n-neighbors =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model and highest ROC value is 0.80 which is given by n-neighbors =9 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The other models are showing 0.74 ROC value of n-neighbors =3 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, 0.78 ROC is given by n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model and n-neighbors =7 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model.

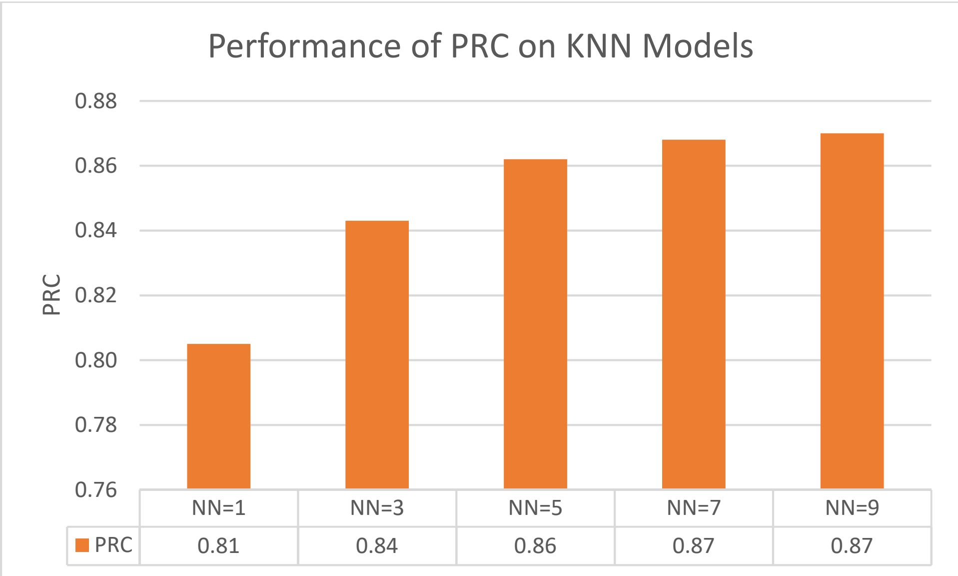


Figure 10: PRC performance of KNN Models

The above picture 10 shows that the measurements of Sensitivity value on KNN models by applying various n-neighbors with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. Least PRC value is 0.81 which is shown by n-neighbors =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. Highest PRC value is 0.87 which is given by there are two models. They are n-neighbors =7 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model and n-neighbors=9 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 0.84 of PRC value is given by n-neighbors =3 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model and 0.86 of PRC value is given by n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model.

Table 5: Mean Deviations of Lazy Classifier

S.No	Classifier	n-neighbors	Mean Absolute Error	Root Mean Squared Error
1	K NN	1	0.16	0.40
2		2	0.17	0.33
3		3	0.17	0.31
4		4	0.18	0.32
5		5	0.18	0.31

The above table 5 shows that the measurements of Sensitivity value on KNN models by applying various n-neighbors with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 0.16 of mean absolute deviation and 0.40 of root mean squared deviation are given by shown by n-neighbors =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 0.17 of mean absolute deviation and 0.33 of root mean squared deviation are given by shown by n-neighbors =3 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 0.17 of mean absolute deviation, 0.31 of root mean squared deviation are given by shown by n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 0.18 of mean absolute deviation, 0.32 of root mean squared deviation are given by shown by n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 0.18 of mean absolute deviation, 0.31 of root mean squared deviation are given by shown by n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model.

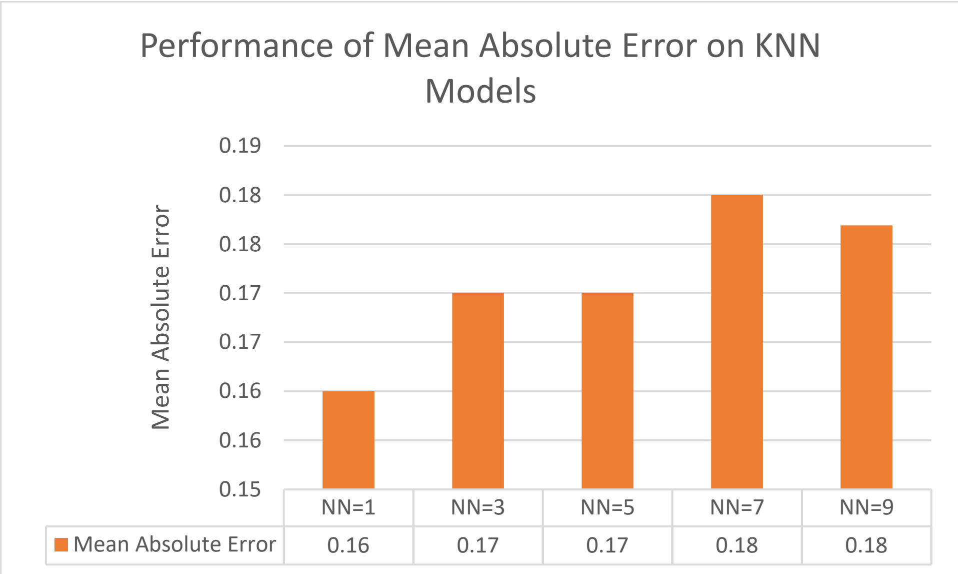


Figure 11: Mean Absolute Error performance of KNN Models

The above picture 11 shows that the measurements of Sensitivity value on KNN models by applying various n-neighbors with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The Least Mean absolute deviation is 0.16 which is given by is shown by n-neighbors=1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The Maximum Mean absolute deviation is 0.18 which are given by n-neighbors =7 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model and n-neighbors =9 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 0.17 of mean absolute error value is given by given by n-neighbors=3 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model and n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model.

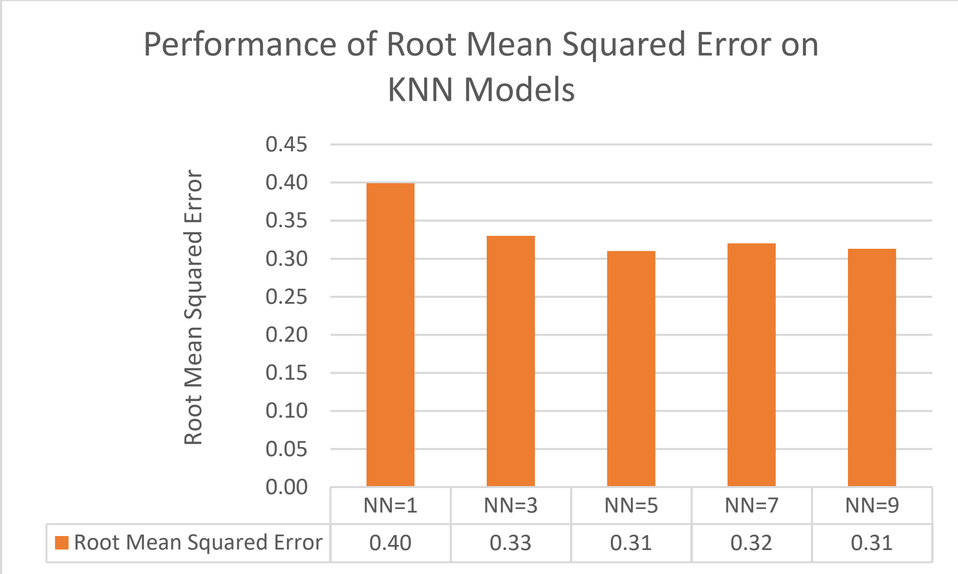


Figure 12: Root Mean Squared Error performance of KNN Models

The above picture 12 shows that the measurements of Sensitivity value on KNN models by applying various n-neighbors with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The least root mean squared deviation is 0.31 which are demonstrated by n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model and n-neighbors =9 by applying various n-neighbors with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The maximum root mean squared error value is 0.40 which is established by n-neighbors =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, n-neighbors =7 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, and N_NEIGHBORS=5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model are having 0.31 of root mean squared error, 0.32 of root mean squared error value and 0.33 of root mean squared error value respectively.

Table 6: Relative Absolute Deviation and Root Relative Squared Deviation of Lazy Classifier

S.No	Classifier	n-neighbors	Relative Absolute Error	Root Relative Squared Error
1	K NN	1	61.66%	110.98%
2		2	66.05%	92.84%
3		3	66.82%	88.39%
4		4	67.88%	87.63%
5		5	68.18%	86.89%

The above table 6 shows that the measurements of relative absolute error and root relative squared error value on KNN models by applying various n-neighbors with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. 61.66% of relative absolute error, 110.98% of root relative squared error is produced by n-neighbors =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. If n-neighbors =3, the model shows 66.05% of relative absolute error, 92.84% of root relative squared error is produced by K =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. If n-neighbors =5, the model shows 66.82% of relative absolute error, 88.39% of root relative squared error is produced by n-neighbors =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model.. If n-neighbors =7, the model shows 67.88% of relative absolute error, 87.63% of root relative squared error is produced by K =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model.. If n-neighbors =9, the model shows 68.18% of relative absolute error, 86.89% of root relative squared error is produced by n-neighbors =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model.

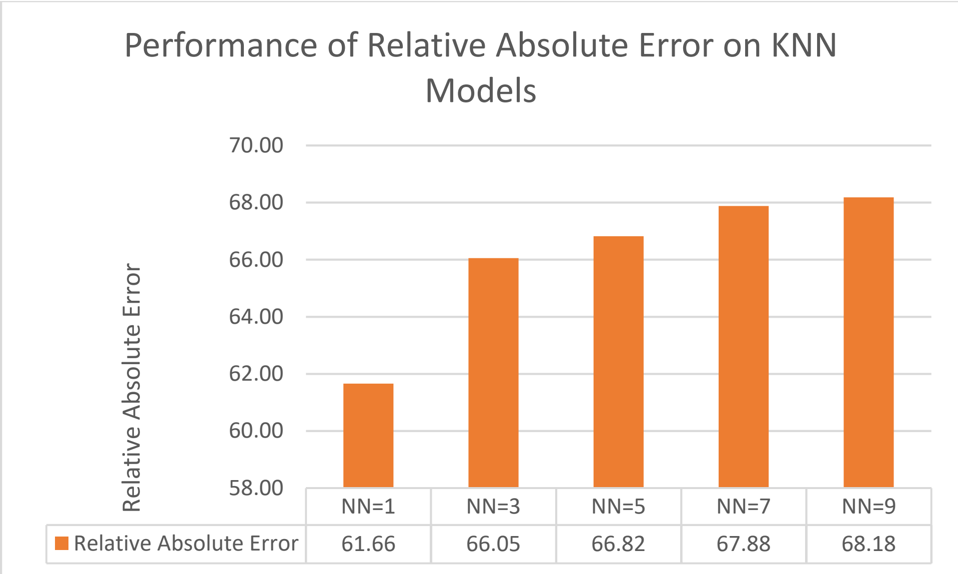


Figure 13: Relative Absolute Error performance of KNN Models

The above picture 13 shows that the measurements of relative absolute error value on KNN models by applying various n-neighbors with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The smallest relative absolute error performance is 61.66 %which is hold by n-neighbors =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The highest relative absolute error performance is 68.18% which is hold by n-neighbors =9 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. Rest of the n-neighbors =3 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, and n-neighbors =7 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model are holing 66.05% of relative absolute error value, 66.82% of relative absolute error value and 67.88% of relative absolute error value respectively.

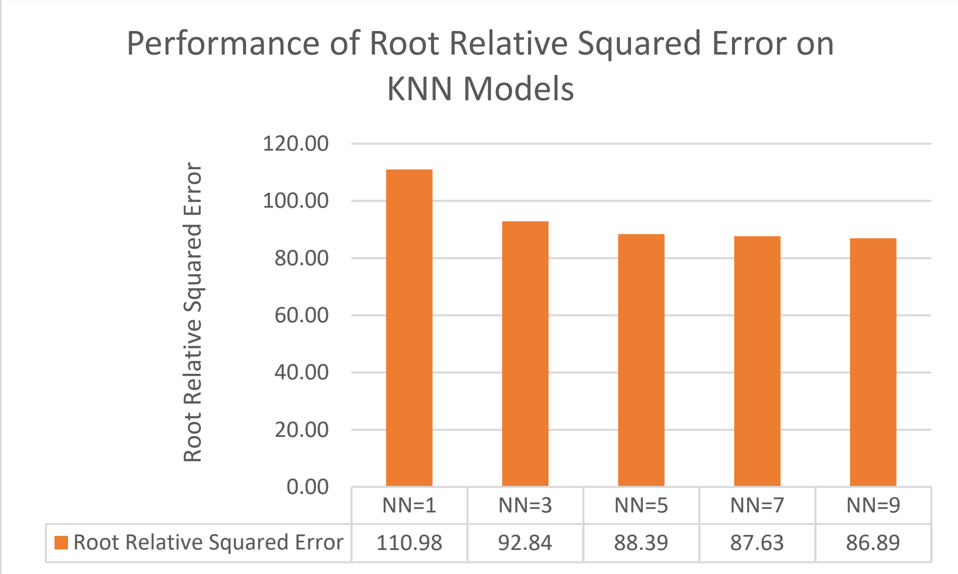


Figure 14: Root Relative Squared Error performance of KNN Models

The above picture 14 shows that the measurements of root relative squared error value on KNN models by applying various n-neighbors with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The largest root relative squared error performance is 110.98 %which is hold by n-neighbors =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The smallest root relative squared error performance is 86.89% which is hold by n-neighbors =9 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. Rest of the n-neighbors =3 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, and n-neighbors =7 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model are holing 92.84% of root relative squared error value, 88.39% of root relative squared error value and 86.89% of root relative squared error value respectively.

V Conclusions

This research work concludes that the highest accuracy value is 88.02% of accuracy level which is acquired by n-neighbors=5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The highest positive predictive value value is 0.87 which is owned by n-neighbors=5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The maximum as well same Sensitivity value is 0.88 which yielded by n-neighbors =3 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The highest F-Measure value is 0.86 is produced n-neighbors =3 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model, n-neighbors =7 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The supreme value of Mean Square Contingency Coefficient shows that 0.45 which is given by n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The maximum amount of kappa value is 0.42 which is picked by n-neighbors =3 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and

Manhattan distance function model. The highest ROC value is 0.80 which is given by n-neighbors =9 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. Highest PRC value is 0.87 which is given by there are two models. They are n-neighbors =7 and n-neighbors =9 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function models. The Least Mean absolute deviation is 0.16 which is given by is shown by n-neighbors =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The least root mean squared deviation is 0.31 which are demonstrated by n-neighbors =5 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The smallest relative absolute error performance is 61.66 %which is hold by n-neighbors =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. The largest root relative squared error performance is 110.98 %which is hold by n-neighbors =1 with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model. This research work recommended that the n-neighbors =5 by applying various n-neighbors with 1.0 of default radius,30 of leaf size, minkowski of distance metric, Ball Tree of Nearest Neighbor Searching algorithm, without parallel job, and Manhattan distance function model.

VI. Conflicts of Interest

The authors declare no conflict of interest.

References

[1] "Taming instabilities in power grid networks by decentralized control" (B. Schäfer, et al, The European Physical Journal, Special Topics, 2016, 225.3: 569-582), in which Dr. Schäfer (Network Dynamics, Max Planck Institute for Dynamics and Self-Organization - MPIDS, Göttingen, Germany) and his co-authors describe in detail the DSGC (Decentral Smart Grid Control) differential equation-based model to assess stability of smart grids;
[2] "Towards Concise Models of Grid Stability" (V. Arzamasov, K. Böhm and P. Jochem, 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), Aalborg, 2018, pp. 1-6), in which Dr. Arzamasov and his co-authors explore how data-mining techniques can address DSGC model simplifications.
[3] Dileep, G. A survey on smart grid technologies and applications. Renew. Energy 2020, 146, 2589– 625.
[4] Mollah, M.B.; Zhao, J.; Niyato, D.; Lam, K.Y.; Zhang, X.; Ghias, A.M.; Koh, L.H.; Yang, L. Blockchain for future smart grid: A comprehensive survey. IEEE Internet Things J. 2020, 8, 18–43.
[5] Syed, D.; Zainab, A.; Ghrayeb, A.; Refaat, S.S.; Abu-Rub, H.; Bouhali, O. Smart grid big data analytics: Survey of technologies, techniques, and applications. IEEE Access 2020, 9, 59564–59585.
[6] Hossain, E.; Khan, I.; Un-Noor, F.; Sikander, S.S.; Sunny, M.S.H. Application of big data and machine learning in smart grid, and associated security concerns: A review. IEEE Access 2019, 7, 13960–13988.
[7] Azad, S.; Sabrina, F.;Wasimi, S. Transformation of smart grid using machine learning. In Proceedings of the 2019 29th Australasian Universities Power Engineering Conference (AUPEC), Nadi, Fiji, 26–29 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
[8] Sun, C.C.; Liu, C.C.; Xie, J. Cyber-physical system security of a power grid: State-of-the-art. Electronics 2016, 5, 40.
[9] Yohanandhan, R.V.; Elavarasan, R.M.; Manoharan, P.; Mihet-Popa, L. Cyber-physical power system (CPPS): A review on modeling, simulation, and analysis with cyber security applications. IEEE Access 2020, 8, 151019–151064.
[10] Ibrahim, M.S.; Dong, W.; Yang, Q. Machine learning driven smart electric power systems: Current trends and new perspectives. Appl. Energy 2020, 272, 115237.
[11] Omitaomu, O.A.; Niu, H. Artificial intelligence techniques in smart grid: A survey. Smart Cities 2021, 4, 548–568.
[12] Jordan, M.I.; Mitchell, T.M. Machine learning: Trends, perspectives, and prospects. Science 2015, 349, 255–260.
[13] Dobson, A.J.; Barnett, A.G. An Introduction to Generalized Linear Models; Chapman and Hall/CRC: Boca Raton, FL, USA, 2018.
[14] Pisner, D.A.; Schnyer, D.M. Support vector machine. In Machine Learning; Elsevier: Amsterdam, The Netherlands, 2020; pp. 101–121.
[15] Deng, Z.; Zhu, X.; Cheng, D.; Zong, M.; Zhang, S. Efficient kNN classification algorithm for big data. Neurocomputing 2016, 195, 143–148.
[16] Xu, R.;Wunsch, D. Survey of clustering algorithms. IEEE Trans. Neural Netw. 2005, 16, 645–678.
[17] Myles, A.J.; Feudale, R.N.; Liu, Y.; Woody, N.A.; Brown, S.D. An introduction to decision tree modeling. J. Chemom. A J. Chemom.Soc. 2004, 18, 275–285.
[18] Sagi, O.; Rokach, L. Ensemble learning: A survey. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 2018, 8, 1249.
[19] Gurney, K. An Introduction to Neural Networks; CRC Press: Boca Raton, FL, USA, 2018.
[20] Doshi, D.; Khedkar, K.; Raut, N.; Kharde, S. Real Time Fault Failure Detection in Power Distribution Line using Power Line Communication. Int. J. Eng. Sci. 2016, 4834.
[21] Gu, C.; Li, H. Review on Deep Learning Research and Applications in Wind and Wave Energy. Energies 2022, 15, 1510.
[22] 22.https://archive.ics.uci.edu/ml/datasets/Electrical+Grid+Stability+Simulated+Data+
[23] Kumaravel, A., et.al, Trails with ensembles on sentimental sensitive data for agricultural twitter exchanges, Indian Journal of Computer Science and Engineering, 2021, 12(5), pp. 1372–1381.
[24] Venkatesh Babu, R et.al. Exploratory data analysis on macroscopic material behavior using micromechanical simulations by applying the gaussian processes with various kernels, Indian Journal of Computer Science and Engineering , 2021, 12(1), pp. 246–253.

Author Profile

Belwin J. Brearley received the B.E degree in Electrical and Electronics Engineering in 2003, M.E in Power Electronics and Drives in 2005 from Anna University, Chennai, India. She received the Ph.D degree in 2021 from B.S.A Crescent Institute of Science and Technology, India. She is currently working in B.S.A Crescent Institute of Science and Technology, India and has a teaching experience of 17 years. Her areas of interest include Microgrid protection, power electronics, and power electronic application to power system.

K.Regin Bose, received the ME degree in Computer Science and Engineering from Anna university in 2005 and Ph.D degree from B.S.A Crescent Institute of Science and Technology, India in 2015. He is currently working in Prince Dr. K. Vasudevan College of Engineering and Technology, India and has a teaching experience of 17 years. His areas of interest include cyber security and mobile communication.

K.Senthil, received Ph.D from Anna University in 2019. He is currently working in Prince Venkateshwara Padmavathy Engineering College, India. His Area interests are AI, ML, Networking and Image Processing

G.Ayyappan, received Ph.D from Bharath Institute of Higher Education and Research in 2018. . He is currently working in Prince Venkateshwara Padmavathy Engineering College, India. His Area interests are AI, ML, Networking and Image Processing.