**Methodology**

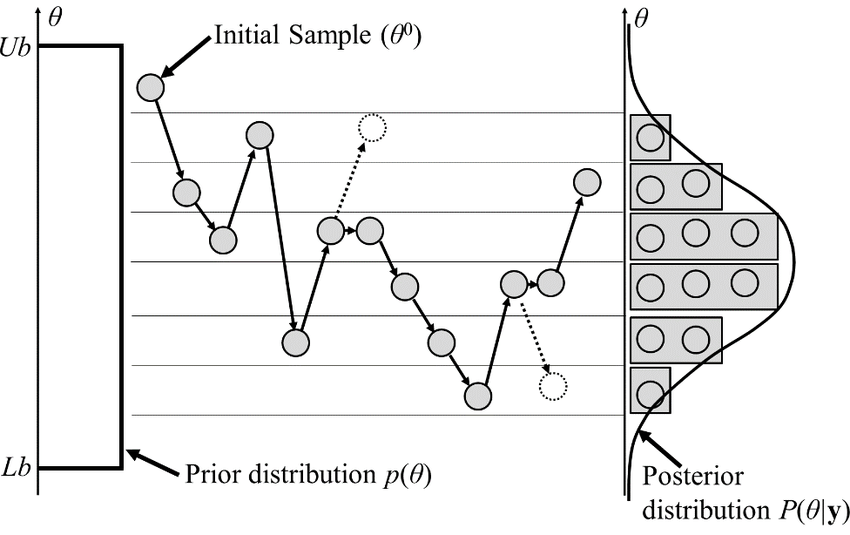
This work comprises of prediction of asphalt temperature (Y) from given air temperature (x1), time of the day (x2), and the depth of measurement(x3). In the perspective machine learning problem the data can be divided into target data and input data, where Y is target and X = [x1 x2 x3] is input. Linear regression formulation for the data can be given by the equation [1]

(1)

Where Yi is i th data point of Y, x1i , x2i , x3i are the ith data points of Air temperature, Time of the day and Depth respectively. β0 , β1 , β2 , β3 are the unknowns in equation, called as regression coefficients. The regression coefficients can be calculated by using Markov Chain Monte Carlo (MCMC) method, which is explained in next subsection.

**Markov Chain Mote Carlo(MCMC)**

MCMC is a class of algorithms used for sampling from an unknown distribution when set of observations is given [5]. From several of the MCMC algorithms we have chosen Metropolis-Hastings algorithm [1] because it is most suitable for multidimensional sampling [6] (this application needs 4-dimensional sampling). While using MCMC for solving regeression problem, the regression coefficients are set to some guess values (or random values). Then the algorithm is iterated by taking random walk along a given distribution, also called as transition function. Here we have taken gaussian normal distribution given by (2) [1].

Figure 1: Graphical illustration of MCMC algorithm

(2)

In every iteration, new regression coefficients are generated by the transition function, with an optimal variance σ. Large variance can cause miss of maxima and too small variance can cause increase in iterations and algorithm may stuck at local maxima. Here σ = 0.0003 is selected by observing the performance of the chains, and μ = βold . This transition is done by using normal distribution random number generator. Hence the transition can be shown as βnew = N(βold , σ). Next step of Metropolis Hastings algorithm is to decide whether to accept or reject the new values of regeression coefficients generated by transition function. For this purpose, ratio of posterior probabilities is calculated and compared with a random number. If the ration of posteriors is greater than random number then the values of β are accepted otherwise the values are rejected. This process can be shown by following equations [1],

(3)

where, R is acceptance ratio and function π is the posterior. Calculation of posterior is discussed later in this section. Then probability of acceptance is defined as P(acc) = min {1, R}, this constrains value of acceptance probability between 0 to 1. A random number from uniform distribution is then generated and called as U. If the random number U < P(acc), then new values of β are accepted and iterations are continued after storing the accepted values in an array and setting βcurrent = βnew. If U ≥ P(acc) then new values of β are rejected and iterations are continued by setting βcurrent = βcurrent. Generalized working of MCMC is shown in figure – 1 and the Metropolis Hastings algorithm flowchart implemented is depicted in figure – 2.

**Posterior Analysis** In order to make decision regarding acceptance or rejection of new proposed regression coefficients there is need of some sample distribution which we can roughly consider as a posterior. If one follows bayes formula then posterior can be defined as [1]

(4)

Denominator term in (4) i.e. normalizing constant cannot be evaluated analytically as the posterior is unknown, thus the term can ignored while calculating approximate posterior for acceptance ratio. Thus (4) can be written as Posterior ∝ Likelihood × Prior, taking ratio of two posteiors will provide an approximate posterior ratio. Prior and likelihood can be taken as some simple distribution from which samples can be obtained. In this work normal distribution is taken as likelihood, representing φ as precision parameter for simplicity, this can be written as

(5)

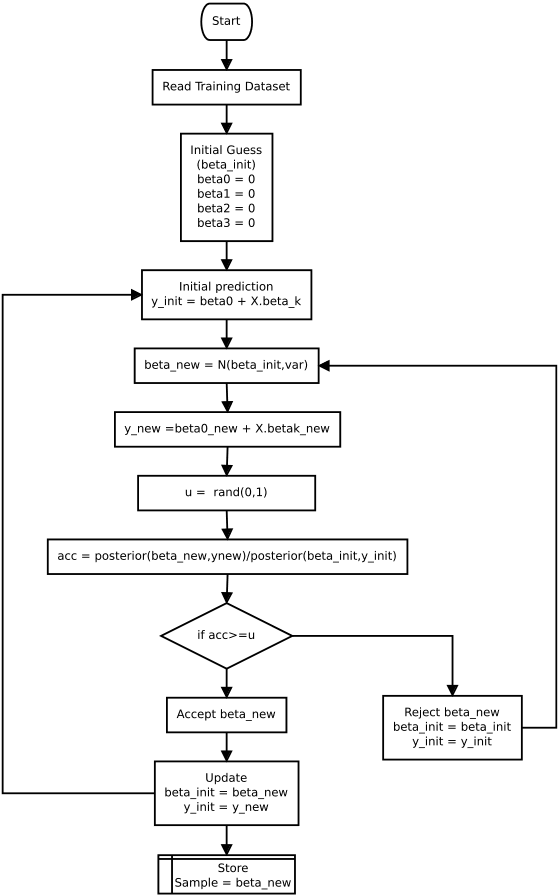
where, same as (1). prior is given by gamma distribution as

(6)

where a = 1 and b = 1 are parameters of gamma distribution, and s k are standard deviations of each value of β, which updates with update in β in every iteration. Hence by using substitutes for likelihood and prior from (5) and (6), posterior can be written as

(7)

where βk = β0 , β1 , β2 , β3 . The calculations have multiple products and multiplications and hence can cause introduction very small non-negative numbers in further calculations, which due to computer datatype limitations may cause erroneous results. To avoid this in actual implementation of posterior calculations, all equations are converted into logarithms.

Figure 2: Metropolis - Hastings algorithm

**Random Forests**

Random forests is a decision tree based algorithm which can be used for classification as well as regression tasks. The target data in this work is Asphalt temperature is continuous data, hence the task here is of regression. The term “Random Forest (RF)” in this paper will indicate “Regression Forest” henceforth. RF can be defined as a “Collection of tree structures classifiers/regressors {h(x,Θt), t = 1, ....} where the Θt are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x in classification task and mean of output of all trees is taken in regression task” [3] as shown in figure - 3.

RF implementation in this work is adpted from chapter 4 of [4]. As given in definition above, output of the forest is average of all the trees given by,

(8)

where π(y|x) is posterior average of the all forests and T is the number of trees. The posterior of each individual is obtained by training the tree for an objective function which is to maximise information gain, given by

(9)

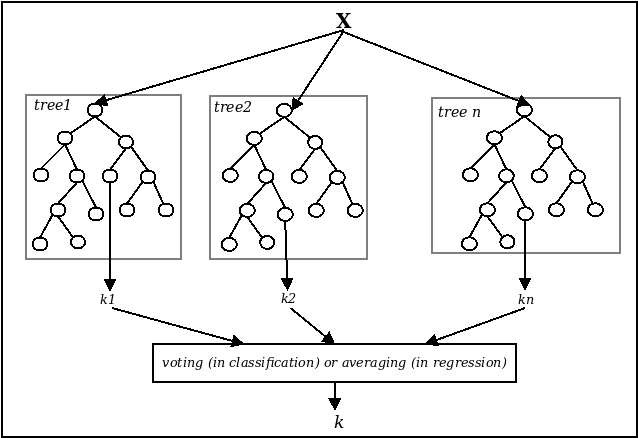
where, Λy is conditional covariance matrix, Sj is training data available at node j, SjL and SjR are left and right node splits of the data. But output in out dataset is single variable array. Hence the information gain can be rewritten as

(10)

where y is the estimated output and yj is mean of the training sample at j th node. In this process of maximum infaormation gain, the data arrived at node j is then split into its right or left child. The data split is governed by the decision making component of tree called as “weak learner” given by

(11)

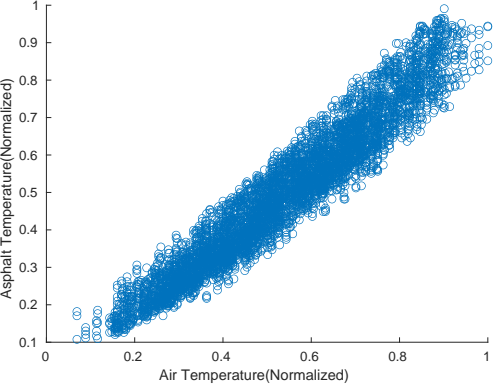
where, φ(x) represent feature space of the data, ψ is geometric shape repre- senting data seperation entity, τ1 and τ2 are the constraining variables that get trained in each iteration. In this work, the data seperators used are vertically aligned to axis (y-axis). Such axis aligned seperators are called as “Stumps” [8]. In this work, decision stumps seperate the data showing relation between each input parameter and the asphalt temperature. Decision stumps for the parameters of day of the time and the depth are simple and are directly defined due to their discrete nature, however for relation between air temperature and the asphalt temperature decision stumps need to be calculated seperately at each training iteration depending upon difference between two consecutive values to be seperated.

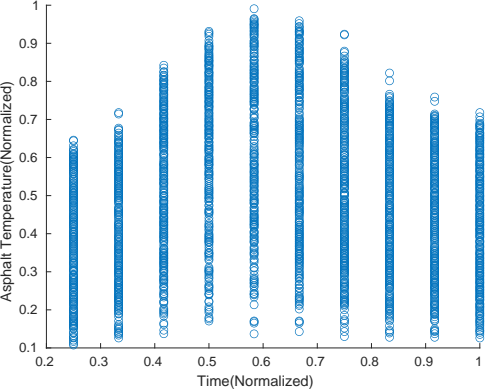
Figure 3: Graphical representation of Random Forest algorithm

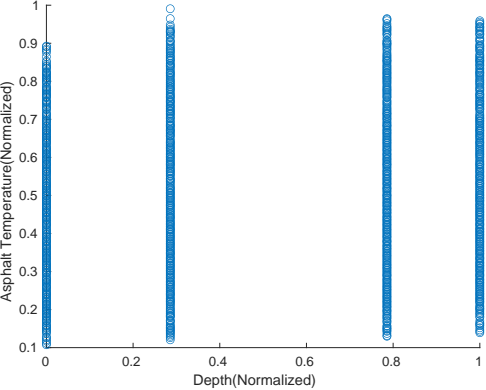
**Hybrid Algorithm (RF based multiple MCMC regression)**

Figures 4, 5, and 6 show the realation between asphalt temperature and air temperature, time of the day and depth respectively. It is clear from the image that the data cannot have a best fit with prediction using a single linear regression model comprising of four regression coefficients as given in equation 1.

As seen in figure 5 and figure 6, time of the day and depth are divided into finite discrete values. As given here there are 10 finite sets of time of the day and 4 finite sets of the depth. Hence this makes clear that the data can be divided into 40 discrete sets, and hence increase of total correlation coefficients from 4 to 160 (4 x 40). The Hybrid algorithm proposed in this work splits data into best possible number (i.e 40) of discrete sets and the sets are labeled. RF is used to learn the labels and MCMC regression is done for each set of the data is combined with the labels. The hybridisation approach is presented in this section.

Figure 4: Scatter plot between Asphalt temperature and air temperature

Figure 5: Scatter plot between Asphalt temperature and time of the day

F

The training part of the algorithm can be represented in following steps

1. **Make clusters of the data using k-means clustering**

The k-means clustering algorithm can be explained in 4 steps according to [2] as

1. Randomly choose initial j clusters C = {c1 , ......., cj }
2. Iterate over j clusters and select closest set of points to each cluster
3. Set a centroid ci for each cluster
4. Repeat 2 and 3 until no cluster changes

after applyinng this to our training dataset the dataset is divided into j = 40 clusters i.e. X = {x1 , x2 ....., x40 }. This assigns each data point a class, that is named after its cluster.

2. **Train random forest to learn the cluster of training data**

Target data in this RF part is training labels that are genrated by the k-means clustering. As described earlier, there is seperate MCMC regression for each data set, but to know which regression coefficients belongs to which data set, there is need of labeling regression coefficients with data set. This work is done by the RF. As shown in figure 6 RF learns the label of data set and puts corresponsing label to each set of regression coefficients produced by MCMC.

3. **Train MCMC for each cluster**

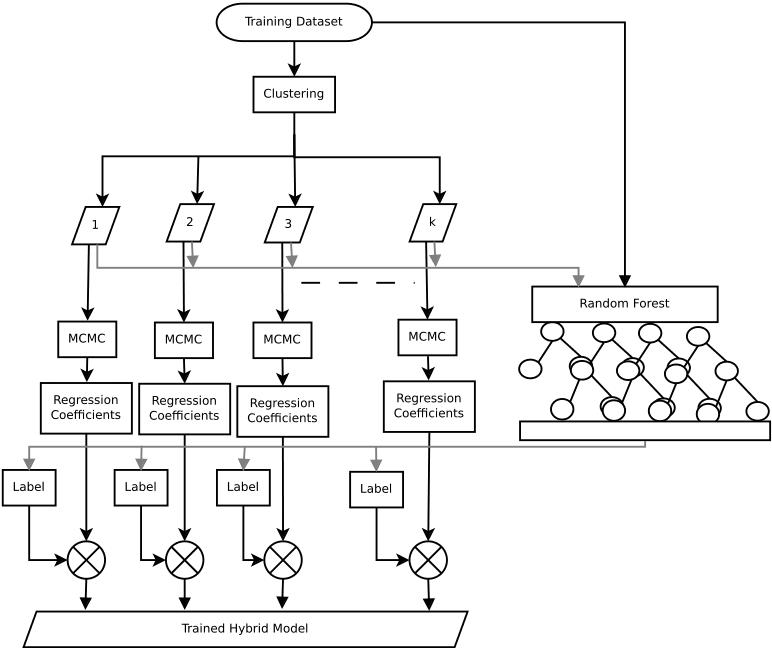
As given in equation 5, 6 and 7, each MCMC posterior distribution is dependent of 4 values of βk. But here in the proposed hybrid algorithm, as the data is split into j = 40 sets, making 160 values of βkj , the posterior distribution that will be obtained for all 160 values can be rewritten as

(12)

which is the product of all individual posteriors of jth dataset given by πj .

(13)

Where,

Figure 6: Hybrid algorithm

(14)

and

(15)

4. **Label corresponding regression coefficients with label generated by the RF**

As seen in equation 12 and 13 the regression coefficient output of this algorithm will for a j x k matrix, where j = 40 and k = 4. As shown in equation 1, it is fixed that β0 is offset value, β1 is air temperature regression coefficient, β2 is time of the day regression coefficint and, β3 is depth regression coefficient, but selection of βk from its j number of sets, is totally dependent on the label of data decided by RF. Hence the data labels generated by RF are added to the sets of regression coefficients generated by MCMC.

The training procedure of hybrid algorithm is depicted in figure 7.

The testing/prediction procedure for the hybrid algorithm can be shown stepwise as

1. read test data

2. classify test data into one of the 40 clusters using RF classifier

3. read regression coefficients corresponding to the class

4. multiply regression coefficients and the result is predicted asphalt temperature

**Results and Disscussion**

Test of the developed algorithms in this work is done for five random combinations of the test dataset, each length of 1500 points. Each algorithm, i.e. Radom Forest, MCMC, and the proposed algorithm, are tested for same combination of the data each time and the results are noted.

Performance measure of the prediction is evaluated using following parameters

1. The R2 parameter is square of correlation, showing 1 for the best prediction, shows similarity between patterns of predicted data and observed data. The parameter is prone to show good fit despite of constant offset available in the data. where R is correlation. R2 is given by

where yo and yp are observed and predicted asphalt temperature values. N is number of test/observation samples.

2. RMSE (Root Mean Square Error): RMSE is the square root of the average of squared errors. The effect of each error on RMSE is proportional to the size of the squared error; thus larger errors have a disproportionately large effect on RMSE. It is given by

3. MAE (Mean Absolute Error) : MAE is the average of the absolute values of the errors. MAE is fundamentally easier to understand than the square root of the average of squared errors(RMSE). MAE is given by

4. NSE (Nash-Sutcliffe Efficiency): NSE is used in the evaluation shows “goodness of the fit” [7] between observed and predicted data. NSE is given by

From observation of the results it is clear that the proposed algorithm gives better results for all evaluation parameters used here. From table 1 it can be seen that best performance is given by Hybrid algorithm in combination M3 with R2 = 0.9625, RM SE = 0.0378, MAE = 0.0308 and NSE = 0.9605.

To understand intutively about comparison of the performance of the models graphical representation of correlation coefficient (R) and RMSE with standard deviation is shown as taylor diagram. Five taylor diagrams for five sets of data are shown with each having four comparison points in it. First point is of original data (observed data) with R = 1, RMSE = 0 ans some standard deviation. Then next there are points of the predicted data by different models. The data points expect to show maximum R, minimum RMSE and standard deviation as similar as possible to the observed data. From taylor diagrams it can be observed that the Hybrid algorithm data point is closest to the actual data and then MCMC and then RF. And also there is significant difference between performance of RF, MCMC and the hybrid algorithm.

Boxplot analysis is also used for performance analysis of the three algorithms. The boxplots shown on follwing figures compare the absolute prediction error. On each box, the central mark indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme data points not considered outliers, and the outliers are plotted individually using the '+' symbol. As seen form the figures it is clear that in each case, best performance is shown by the hybrid algorithm with none or minimum outlier points. And the RF and MCMC contain highest and second highest number of outliers in prediction error respectively.

**Perfomance Evaluation**

*R*

2

**RMSE**

**MAE**

**NASH**

**M1**

**RF**

0.9249

0.0534

0.0387

0.9228

**MCMC**

0.9384

0.0481

0.0389

0.9340

**HYBRID** 0.9606

0.0385

0.0316

0.9578

**M2**

**RF**

0.9251

0.0526

0.0383

0.9237

**MCMC**

0.9362

0.0481

0.0388

0.9329

**HYBRID** 0.9604

0.0379

0.0312

0.9577

**M3**

**RF**

0.9224

0.0540

0.0394

0.9197

**MCMC**

0.9382

0.0480

0.0388

0.9335

**HYBRID** 0.9616

0.0379

0.0311

0.9584

**M4**

**RF**

0.9268

0.0536

0.0395

0.9259

**MCMC**

0.9399

0.0479

0.0387

0.9369

**HYBRID** 0.9628

0.0377

0.0310

0.9608

**M5**

**RF**

0.9272

0.0529

0.0385

0.9253

**MCMC**

0.9381

0.0485

0.0388

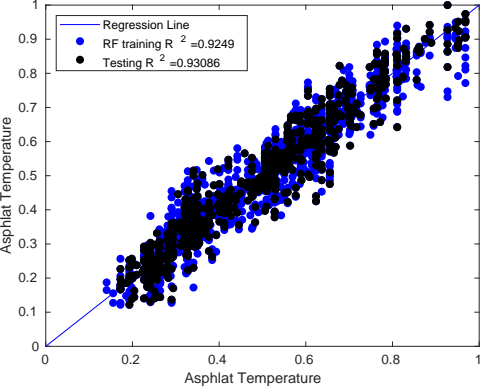
0.9346

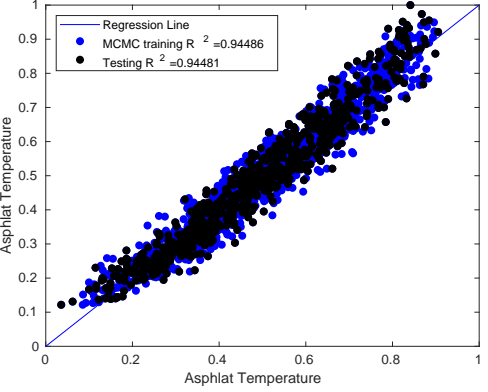
**HYBRID** 0.9619

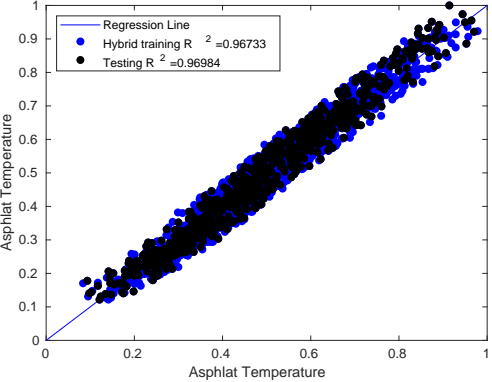
0.0381

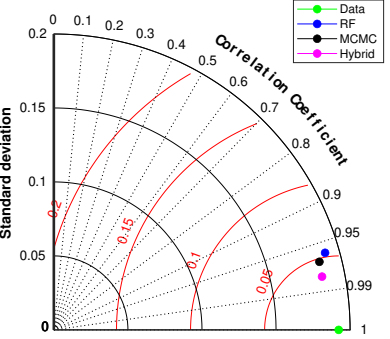
0.0312

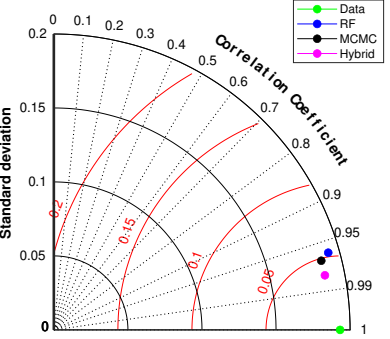
0.9590

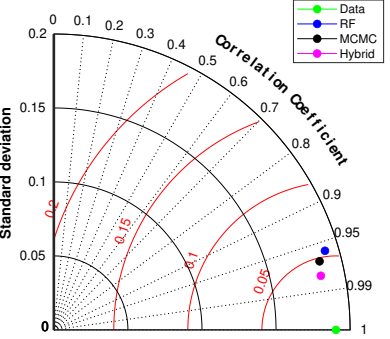


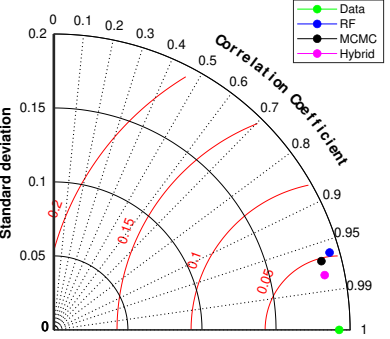


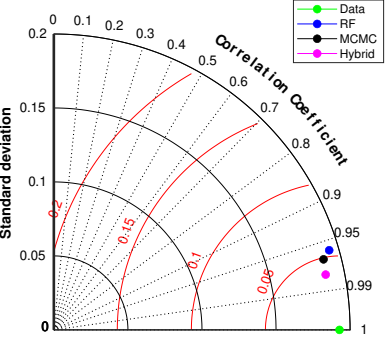


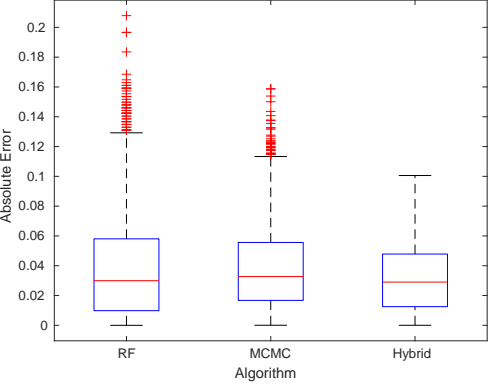


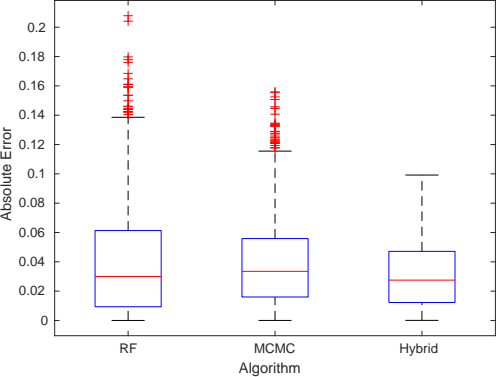


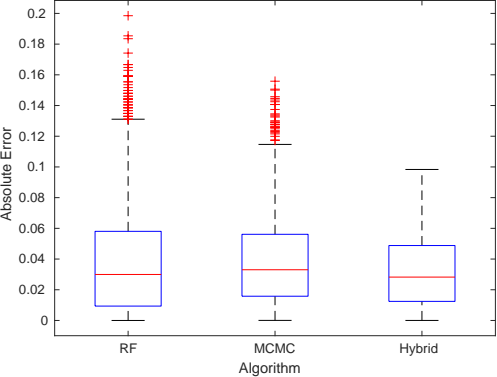


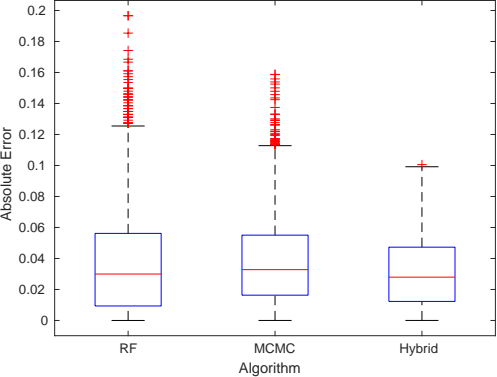


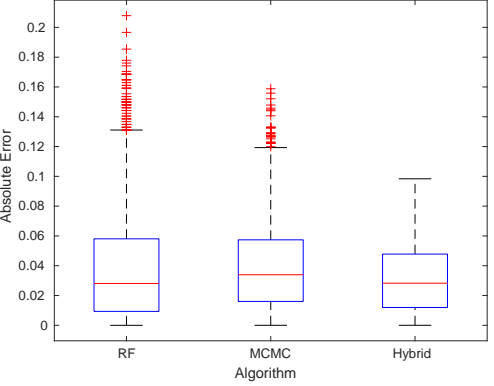
****

****

****

****

****

****

**References**

[1] J. Albert and J. Hu. Probability and Bayesian Modeling. Chapman & Hall/CRC Texts in Statistical Science. CRC Press, 2019.

[2] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, pages 1027–1035, 2007.

[3] Leo Breiman. Using iterated bagging to debias regressions. Machine Learning, 45(3):261–277, 2001.

[4] A Criminisi, J Shotton, and E Konukoglu. Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. Technical report, 2011.

[5] D.J.C. MacKay. Information theory, inference, and learning algorithms. Cambridge Univ Pr, 2003.

[6] Daniel Mackey, David W. Hogg, Dustin Lang, and Jonathan Goodman. emcee the mcmc hammer. Publications of the Astronomical Society of the Pacific, 125(925).

[7] Richard H. McCuen, Zachary Knight, and A. Gillian Cutter. Evaluation of the nash-sutcliffe efficiency index. Journal of Hydrologic Engineering, 11(6):597–602, 2006.

[8] Paul Viola and MJ Jones. Robust real-time face detection. IJCV, 57(2):137–154, 2004.