Prediction of Climate Change using SVM and Naïve Bayes Machine Learning Algorithms

C. Karthikeyan[a], Gurram Sunitha[b], J. Avanija[c] , K. Reddy Madhavi[d], and E.S. Madhan[e]

[a] Department of CSE, KoneruLakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India
[b]Professor, Department of CSE, SreeVidyanikethan Engineering College. Tirupati, India.
[c]Associate Professor, Department of CSE, SreeVidyanikethan Engineering College, Tirupati, India.
[d]Associate Professor, Department of CSE, SreeVidyanikethan Engineering College, Tirupati, India,
[e]School of Computing, SRM Institute of Science and Technology, SRM Nagar, Kattankulathur,Chengalpattu District, Tamil Nadu-603203, INDIA.

Article History: Received: 11 January 2021; Accepted: 27 February 2021; Published online: 5 April 2021

Abstract: Various reasons are there in failures of Intergovernmental Panel on Climate Change (IPCC) simulation model for prediction of climate change. For the better understanding of IPCC model's failures by researchers, an improvement is qualitative and quantitative analysis is required and to be implemented. We come across a continuous crashes in simulation of Parallel Ocean Program (POP2) component of the Community Climate System Model (CCSM4), while measuring the impact of ocean model parameter uncertainties on weather simulations, during the period of uncertainty quantification (UQ) ensemble. This manuscript analyse the different machine learning algorithms, such as, Random forest, Linear Regression, k-means and naïve-bayes algorithms. From machine learning, a quality classifier called support vector machine (SVM) classification is used to predict and quantify the failures probability as a function of the values of POP2 parameters. Apart from quantification and prediction, this method performs a better understanding in simulation crashes in other complex geo-scientific models.

Keywords: Support Vector Machine, Naïve Bayes' algorithm, Community Climate System Model.

1. Introduction

Recent global 3D climate models are working excellently in unfurl of complex scientific problems using latest software's. To solve the equations of state of machines, energy, atmospheric momentum, land, reservoirs of earth system, oceans, we need a thousand of details and billion lines of codes. With a quantity of interest, to prompt the cycles of sulphur, ozone gases [1] and other extent related to climates require a numerous algorithms on chemical, biological, geologic, and anthropogenic processes.

Machine learning deals with study of algorithms which improves consequently through experience. Several number of algorithms [2-4] has been plotted for several functions on Artificial Intelligence. This introduction section analyses the various machine learning algorithms. These algorithms operate on different dimensions of magnitude in time, space, solid, liquid, gas and different component phases, which contain fine particles having multiple complexity[5].Prediction of Weather using Data mining techniques yields well results better than the traditional metrological approaches [6].

In linear regression, which seeks to identify and forecast the low and high temperatures as a linear functional combinations of the features [12]. Since straight relapse can't be utilized with characterization information, this calculation didn't utilize the climate order of every day [15-16].Climatologically conditions need to be predicted to save the life of people which is a challenging problem. Machine learning techniques may be applied to forecast the extreme weather events. The author [16] addresses the use of ML algorithms to filter and visualize the extreme weather event. In the proposed system, AFM method is used to filter the events and based on the results the class labels are assigned.

The extreme and non- extreme weather events are visualized using DBSCAN and K-means clustering algorithm. The past extreme events like BOB (06), Thane and Vardahare validated and the results are verified by the parameters like homogeneity and completeness. This paper addresses the use of ML algorithms to filter and visualize the extreme weather event[8]. In the proposed system, AFM technique is used to filter the events and based on the results the class labels are assigned. The extreme and non- extreme weather events are visualized using DBSCAN and K-means clustering algorithm [17-18].In this paper, introduced probabilistic networks[7] and shows that applicability for nearby climate guaging and downscaling. The fundamental results appeared in this paper just outline how such models can be manufactured and how they can use for performing derivation.

2. Naive Bayes Algorithm

The below equation shows the Bayes theorem,

$$\mathbf{P(B \mid A)P(A)}$$

P(A |B)= P(B) ---- Eq.(1)

Probability of happening A calculated using Bayes Theorem. Here, B is the Evidence and A is the Hypothesis. where A and B are events

- To find probability of occasion A, given the occasion B is valid. Occasion B is named as proof.
- P(A) is the priori of A
- P(A|B) is deduced probability of B

3. Support Vector Machine (SVM)

SVM classification [9] classification problem can also be controlled using analogous methods, such as decision trees, and random forests. The learning algorithms associated with supervised learning model for regression and classification used for analysis are implemented using support vector networks or support vector machines (SVM) in machine learning.

A SVM classifier [14] builds a model to predict classes with new examples. A Binary SVM Classifier is obtained; if there are only 2 classes are available. An apparent gap separates these data points. A hyper plane dividing 2 classes will predict straight forward. The outcome of drawing of hyper-plane is called as a maximum-margin hyper plane.

4. Confusion Matrix

The below table shows the confusion matrix [10-11],

- Positive (P)
- Negative(N)
- True Positive (TP)
- False Negative (FN)
- True Negative (TN)
- False Positive (FP)

Table 1. Confusion Matrix

		Actual	
		Failure	**Success**
Predicted	**Failure**	**TP**	**FP**
	Success	**FN**	**TN**
		TPR = TP/(TP+FN)	**FPR = FP/(FP+TN)**

Classification Rate or Accuracy is given by the relation:

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{---- Eq.(2)}$$

5. Dataset Information

This dataset contains records of propagation crashes experienced during environment model vulnerability evaluation (UQ) troupes. Outfit individuals were built utilizing a Latin hypercube strategy in LLNL's UQ Pipeline programming structure to test the vulnerabilities of 18 model boundaries inside the (POP2) segment of the (CCSM4).The aim is to utilize classification to forecast simulation result using sensitivity analysis from input parameter values for regulating the seed of simulation crashes.

5.1. Attribute Information

Aim of it is to estimate environment model re-enactment results given scaled estimations of environment model information boundaries.

Section 1:Latin hypercube study ID (study 1 to contemplate 3)

Section 2: reproduction ID (run 1 to run 180)

Sections 3-20: values of 18 environment model boundaries

Section 21: simulation outcome (0 = disappointment, 1 = achievement)

5.2. Features description

Consider an environment dataset that portrays the climate conditions for if the environment crashes. Given climate conditions; each tuple characterizes the conditions as fit ("Yes") or unfit ("No") for environment crashes. The dataset grouped into Feature framework and the Response vector. In above dataset, the class variable name is "Outlook".

Table 2. Parameters Sampled in the CCSM4 Parallel Ocean Model

	Parameter[a]	[low, default, high]	Scale[b]	Module	Description
1	vconst_corr	$[0.3, 0.6, 1.2] \times 10^{7}$	lin	hmix_aniso	variable viscosity parameter (vconst_1, vconst_6)
2	vconst_2	[0.25, 0.5, 2.0]	log	hmix_aniso	variable viscosity parameter
3	vconst_3	[0.16, 0.16, 0.2]	lin	hmix_aniso	variable viscosity parameter
4	vconst_4	$[0.5, 2.0, 10.0] \times 10^{-8}$	log	hmix_aniso	variable viscosity parameter
5	vconst_5	[2, 3, 5]	lin	hmix_aniso	variable viscosity parameter
6	vconst_7	[30.0, 45.0, 60.0]	lin	hmix_aniso	variable viscosity parameter
7	ah_corr	$[2.0, 3.0, 4.0] \times 10^{7}$	lin	hmix_gm	diffusion coefficient for Redi mixing (ah) and background horizontal diffusivity within the surface boundary layer (ah_bkg_srfbl)
8	ah_bolus	$[2.0, 3.0, 4.0] \times 10^{7}$	lin	hmix_gm	diffusion coefficient for bolus mixing
9	slm_corr	[0.05, 0.3, 0.3]	log	hmix_gm	maximum slope for bolus (slm_b) and Redi terms (slm_r)
10	efficiency_factor	[0.05, 0.07, 0.1]	lin	mix_submeso	efficiency factor for submesoscale eddies
11	tidal_mix_max	[25.0, 100.0, 200.0]	log	tidal	tidal mixing threshold
12	vertical_decay_scale	$[2.5, 5.0, 20.0] \times 10^{4}$	log	tidal	vertical decay scale for tide induced turbulence
13	convect_corr	$[1.0, 10.0, 50.0] \times 10^{3}$	log	vertical_mix	tracer (convect_diff) and momentum (convect_visc) mixing coefficients in diffusion option
14	bckgrnd_vdc1	[0.032, 0.16, 0.8]	log	vmix_kpp	base background vertical diffusivity
15	bckgrnd_vdc_ban	[0.5, 1.0, 1.0]	lin	vmix_kpp	Banda Sea diffusivity
16	bckgrnd_vdc_eq	[0.01, 0.01, 0.5]	log	vmix_kpp	equatorial diffusivity
17	bckgrnd_vdc_psim	[0.1, 0.13, 0.5]	log	vmix_kpp	maximum PSI induced diffusivity
18	Prandtl	[4.0, 10.0, 20.0]	log	vmix_kpp	ratio of background vertical viscosity and diffusivity

6. Results and Discussions

```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
      yes        No
0.0952381 0.9047619

Conditional probabilities:
     Study
Y         [,1]      [,2]
  yes 1.833333 0.8451543
  No  2.000000 0.8140986

     Run
Y         [,1]     [,2]
  yes 94.72222 57.64998
  No  90.80994 53.45293

     vconst_corr
Y          [,1]      [,2]
  yes 0.8079654 0.1574152
  No  0.4830836 0.2831933

     vconst_2
Y          [,1]      [,2]
  yes 0.7880777 0.1436818

  yes 0.5180201 0.2755545
  No  0.5146385 0.2922793

     bckgrnd_vdc_eq
Y          [,1]      [,2]
  yes 0.4132399 0.2796761
  No  0.5209993 0.2909360

     bckgrnd_vdc_psim
Y          [,1]      [,2]
  yes 0.4140503 0.2597306
  No  0.5038055 0.2968494

     Prandtl
Y          [,1]      [,2]
  yes 0.5080638 0.3541220
  No  0.4994123 0.2874167

> predNB1=predict(NB,mtesting)
> length(predNB1)
[1] 162
> length(mtesting$outcome)
[1] 162
> table(mtesting$outcome,predNB1)
     predNB1
      yes  No
  yes   8  10
  No    0 144
> plot(predNB1)
> Accuracy(predNB1,mtesting$outcome)
[1] 0.9382716
```

Figure 1. Result Screenshot for Naive Bayes Classifier

The Figure 1.shows the predictive values of Naïve Bayes' Classifier for Discrete Predictors. The model makes the contingent Probability for each element independently. We likewise have the a priori probabilities which show the dissemination of our information. From the Figure 1. We understand that the accuracy value using naive Bayes classifier for discrete predictors is 0.9382

The underneath plot is showing that our classifier is giving best precision. The result classify 151 out of 156 "No" cases correctly and 1 out of 5 "Yes" cases correctly. This implies the capacity of Naïve Bayes calculation to anticipate "No" cases is about 96.5% yet it tumbles down to just 20% of the "Yes" cases bringing about a general precision of 94%.

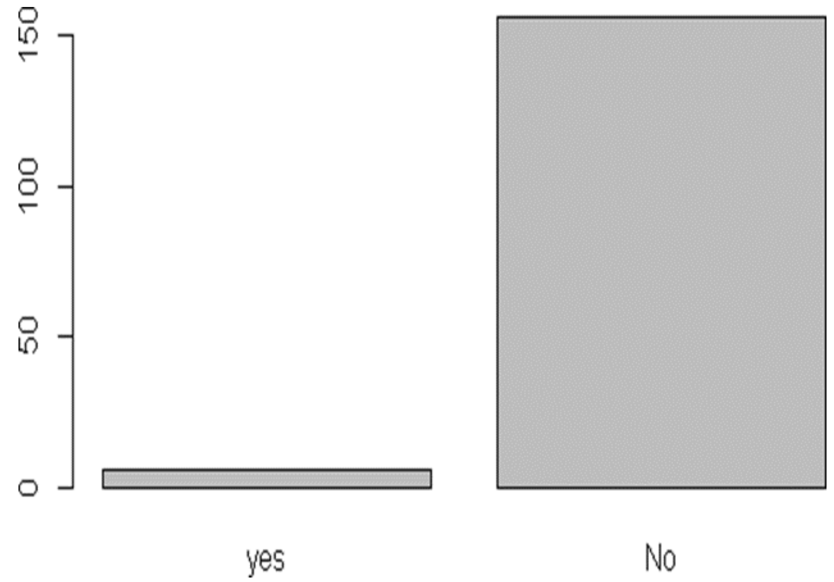

Figure 2. Prediction of 'Yes' and 'No' cases

We have to apply the same data set information in Support Vector Machine algorithm.

```
> confusionMatrix(
+     factor(test_pred),
+     factor(testing$outcome)
+   )
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0   9   1
         1   4 148

               Accuracy : 0.9691
                 95% CI : (0.9294, 0.9899)
    No Information Rate : 0.9198
    P-Value [Acc > NIR] : 0.00859

                  Kappa : 0.7663
 Mcnemar's Test P-Value : 0.37109

            Sensitivity : 0.69231
            Specificity : 0.99329
         Pos Pred Value : 0.90000
         Neg Pred Value : 0.97368
             Prevalence : 0.08025
         Detection Rate : 0.05556
   Detection Prevalence : 0.06173
      Balanced Accuracy : 0.84280

       'Positive' Class : 0
```

Figure 3. Result Screenshot for Confusion Matrix

Data slicing is to split the data into train and test set. Using confusion matrix, we can be able to know how accurately the model is working. We can get statistics of our results based on applying the confusion matrix. From the Figure 3., it clearly represent that the accuracy value is 0.9691 and balance accuracy is 0.84280. Our model using SVM shows the accuracy for test set is 97%.

From our results, we conclude that the SVM is the best algorithm for our dataset because the accuracy is more when compared to naïve bayes. Therefore, support vector machine algorithm is suitable for predicting climate crashes. The beneath plot is showing that our classifier is giving best precision on C =0.25

```
> svm_Linear_Grid
Support Vector Machines with Linear Kernel

378 samples
 20 predictor
  2 classes: '0', '1'

Pre-processing: centered (20), scaled (20)
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 339, 340, 341, 341, 340, 340, ...
Resampling results across tuning parameters:

  C     Accuracy   Kappa
  0.00        NaN        NaN
  0.01  0.9128254  0.0000000
  0.05  0.9302592  0.2733598
  0.10  0.9497733  0.5799444
  0.25  0.9516000  0.6574496
  0.50  0.9472353  0.6311070
  0.75  0.9455271  0.6289049
  1.00  0.9463818  0.6394730
  1.25  0.9446499  0.6288678
  1.50  0.9437952  0.6274922
  1.75  0.9402864  0.6032743
  2.00  0.9385545  0.5954182
  5.00  0.9368701  0.5995554

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was C = 0.25.
> plot(svm_Linear_Grid)
> test_pred_grid <- predict(svm_Linear_Grid, newdata = testing)
> test_pred_grid
  [1] 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
 1 1 1 1 1
 [47] 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 0 1 1 1
 [93] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1
[139] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1
Levels: 0 1
```

Figure 4. Result screenshot for SVM

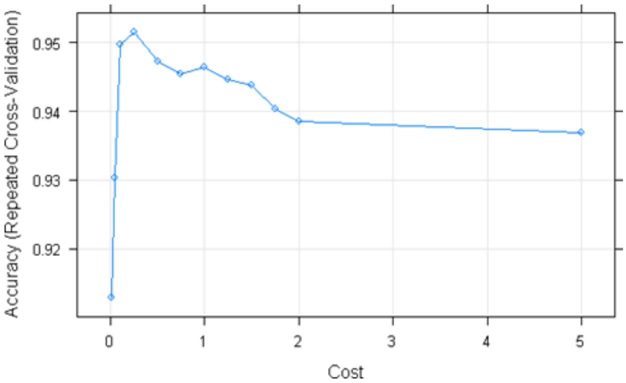

Figure 5. Accuracy

7. Conclusion

A sequence of code crashes occurred during execution of simulations with the mentioned parameters. Machine learning types are used to define in aggregating failure probabilities of listed parameters. Sensitivity analysis utilized to analyze various parameters and modules to explain the simulation failures. We have presented probabilistic networks and show their relevance for nearby climate forecasting and down scaling. In this paper, the fundamental results shows how such models can be fabricated and how they can use for performing derivation.

Further analysis is as yet required for deciding the pragmatic employable productivity of these models; first tests are being promising. We are right now chipping away at adjusting existing learning calculations for managing this unique issue, where the proof is constantly given in similar subset of factors.

References

1. ArunSai, G., BharatKumar, B., Likhitha, K. S., &Anitha, R. (2018). Detection of noxious gases by implementing internet of things technology. International Journal of Engineering and Technology(UAE), 7(2), 18-22. doi:10.14419/ijet.v7i2.32.13518
2. Srinivasa Rao, Y., Ravikumar, G., Kesava Rao, G., & Syed, M. S. (2017). Interconnected transmission line fault detection using wavelet transform and a novel machine learning algorithm. Journal of Advanced Research in Dynamical and Control Systems, 9(12), 142-150.
3. Bommadevara, H. S. A., Sowmya, Y., &Pradeepini, G. (2019). Heart disease prediction using machine learning algorithms. International Journal of Innovative Technology and Exploring Engineering, 8(5), 270-272.
4. Anila, M., &Pradeepini, G. (2017). Study of prediction algorithms for selecting appropriate classifier in machine learning. Journal of Advanced Research in Dynamical and Control Systems, 9(Special Issue 18), 257-268.
5. Pratuisha, K., Rao, D. R., & Murthy, J. V. R. (2017). A comprehensive analysis on different domain of machine learning. Journal of Advanced Research in Dynamical and Control Systems, 9(18), 349-356.
6. Angel Prathyusha, K., Mahitha, Y., Prasanna Kumar Reddy, N., & Raja Rajeswari, P. (2018). A survey on prediction of suitable crop selection for agriculture development using data mining classification techniques. International Journal of Engineering and Technology(UAE), 7(3.3 Special Issue 3), 107-109.
7. Narasinga Rao, M. R., Venkatesh Prasad, V., Sai Teja, P., Zindavali, M., &Phanindra Reddy, O. (2018). A survey on prevention of overfitting in convolution neural networks using machine learning techniques. International Journal of Engineering and Technology(UAE), 7(2.32 Special Issue 32), 177-180.
8. LalithendraNadh, V., &Syam Prasad, G. (2018). Support vector machine in the anticipation of currency markets. International Journal of Engineering and Technology(UAE), 7(2), 66-68. doi:10.14419/ijet.v7i2.7.10262
9. Lakshmi, C. R., Rao, D. T., & Rao, G. V. S. (2018). Fog detection and visibility enhancement under partial machine learning approach. Paper presented at the IEEE International Conference on Power, Control, Signals and Instrumentation Engineering, ICPCSI2017, 1192-1194. doi:10.1109/ICPCSI.2017.8391898
10. Rajesh Kumar, T., Suresh, G.R., Kanaga Subaraja, S. & Karthikeyan, C. (2020). "Taylor-AMS features and deep convolutional neural network for converting nonaudible murmur to normal speech". Computational Intelligence, 2020, pp. 1-12..
11. VijendraBabuD,N.R.Alamelu, Performance analysis of medical images applying novel Morpho codec, ARPN Journal of Engineering and Applied Sciences,2015, 10 (9), pp. 3966-3969.
12. Siva Kumar, P., Sarvani, V., Prudhvi Raj, P., Suma, K., &Nandu, D. (2017). Prediction of heart disease using multiple regression analysis and support vector machines. Journal of Advanced Research in Dynamical and Control Systems, 9(18 Special Issue), 675-682.
13. Banchhor, C., &Srinivasu, N. (2018). FCNB: Fuzzy correlative naive bayes classifier with MapReduce framework for big data classification. Journal of Intelligent Systems, doi:10.1515/jisys-2018-0020
14. Murthy, K. V. S. S. R., &Satyanarayana, K. V. V. (2018). Intrusion detection mechanism with machine learning process A case study with FMIFSSVM, FLCFSSVM, misuses SVM, anomaly SVM and bayesian methods. International Journal of Engineering and Technology (UAE), 7, 277-283.
15. Narayana, Mahaboob, B., Venkateswarlu, B., Sankar, J. R., &Balasiddamuni, P. (2018). A treatise on testing general linear hypothesis in stochastic linear regression model. International Journal of Engineering and Technology (UAE), 7(4.10 Special Issue 10), 539-542.
16. C.S.KanimozhiSelvi, G.Sowmiya, Prediction of Extreme Weather Events using Machine Learning Technique, International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 4 (2019) pp. 925-929, (2019).
17. Srinivas, K., & Kiran, K. V. D. (2018). Performance analysis of hybrid hierarchical K-means algorithm using correspondence analysis for thyroid drug data. Journal of Advanced Research in Dynamical and Control Systems, 10(12 Special Issue), 698-712.
18. Rachapudi, V., VenkataSuryanarayana, S., &SubhaMastan Rao, T. (2019). Auto-encoder based K-means clustering algorithm. International Journal of Innovative Technology and Exploring Engineering, 8(5), 1223-1226.