iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** VII **Month of publication:** July 2017

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue VII, July 2017- Available at www.ijraset.com

SVM Highest Probability Nearest Neighbour Algorithm -A Hybrid Approach for Prediction Analysis in Intelligent Systems

Jeet Thakur[1], Jay L.Borade[2]
[1]*Third year Information Technology student at Fr CRCE, Mumbai University,*
[2]*Assistant Professor Information Technology at Fr CRCE, Mumbai University*

Abstract: Prediction is basically based upon the historical time series data. The basic parameters of weather prediction as an example used throughout the paper are maximum temperature, minimum temperature, rainfall, humidity etc. The basic Data mining operations are employed to get a useful pattern from a huge volume of data set. Different testing and training scenarios are performed to obtain the accurate result. In this paper we are trying to predict the future conditions for examples like weather conditions based upon above parameters by Machine learning algorithmic Techniques. Experimental results of using SVM Highest probability algorithm prove to be a useful approach for forecasting. The Bayesian algorithm and Support vector machine are the most common methods for prediction. In this paper we are proposing Support vector method for nearest neighbour technique, optimum in accordance with the coding aspects in R and python programming languages.
Keywords : SVM, weather forecasting, Naïve Bayesian, k-means, Prediction Analysis.

I. INTRODUCTION

Forecasting is the application of science and technology to predict the state of the data set for a future time and a given location. Human beings have attempted to predict the data sets informally for millennia, and formally since at least the 19th century. Forecasts are made by collecting quantitative data about the current state of the data and using scientific understanding of algorithmic processes to predict how the data will evolve. Here to propose our algorithm we have taken Weather forecasting as a model for implementation. In this research a weather forecasting model using, The weather parameters like maximum temperature, minimum temperature, relative humidity and rainfall is going to be predicted using the features extracted over different periods as well as from the weather parameter time-series itself.

The approach applied here uses SVM Highest Probability Nearest Neighbour Algorithm a Hybrid approach for Prediction analysis for Intelligent Systems for supervised learning using the data recorded at a particular station. The data was used to allow the machine to learn itself from scratch and predict the future weather conditions as well. The model can be suitably adapted for making forecasts over larger geographical areas. The results indicate that, in some cases, the effect of weather is indirectly taken into account by other variables, and explicit use of weather variables may not be necessary. However, the decision to include or exclude weather variables should be analyzed for each individual situation.

II. WEATHER FORECASTING SYSTEMS

A. Weather Data Collection

Observations of atmospheric pressure, temperature, wind speed, wind direction, humidity, and precipitation are made near the earth's surface by trained observers, automatic weather stations. The World Meteorological Organization acts to standardize the instrumentation, observing practices and timing of these observations worldwide.

B. Weather Data Assimilation

During the data assimilation process, information gained from the observations is used in conjunction with a numerical model's most recent forecast for the time that observations were made to produce the meteorological analysis. This is the best estimate of the current state of the atmosphere. It is a three dimensional representation of the distribution of temperature, moisture and wind.

C. Numerical Weather Prediction

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue VII, July 2017- Available at www.ijraset.com

Numerical Weather Prediction (NWP) uses the power of computers to make a forecast. Complex computer pro- grams, also known as forecast models, run on supercomputers and provide predictions on many atmospheric variables such as temperature, pressure, wind, and rainfall. A forecaster examines how the features predicted by the computer will interact to produce the day's weather

III. COMMON DATA MINING ALGORITHMS USED FOR PREDICTION MODELS

The most common techniques mentioned above like Numerical weather prediction, Weather Data Assimilation, etc. are used to collect data and project analyzed data. However analyzing the data is mostly done using Data mining concepts.

There are 2 very widely used Algorithms used for clustering data and analyzing the patterns in them.

A. Naïve Bayesian classifier
B. Support vector machine nearest neighbor Apart from these there are many algorithms which are useful in classifying data into frequent data sets like
C. Apriori algorithm
D. K means
E. K mediod
F. K nearest neighbor

These algorithms have their own advantages and dis-advantages.

Each Data mining algorithms have their common language to code in, prediction and analysis are better done in Machine learning practice where the common language to code in are the Python and R programming language

The R programming language has 2 in-built packages namely for the most widely used data mining algorithms.

The inbuilt packages of R allow users to dynamically allow the machine to learn the data sets and then later analyze the change in trends of the data sets. Let us understand both these algorithms and their coding practices separately and then come on to a conclusion about the better practice amongst the two.

We would look at all the algorithms and also check out the ways they are used to determine the weather for the day or over the week. Python also being the second common coding language used in Machine learning we will check out their codes in python too, at the end we will summarize why SVM is suited to be the best used method projected by us for Machine learning.

IV. EXISTING PROMINENT ALGORITHM USED IN MACHINE LEARNING PREDICTION MODELS – NAÏVE BAYESIAN CLASSIFICATION AND CLUSTERING ALGORITHM

A. Definition

It is a classification technique based on Bayes' theorem with an assumption of independence between predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, here a day may be considered to be sunny if it is dry, not overcast and bright. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier would consider all of these properties to independently contribute to the probability that this day is sunny.

Naive Bayesian model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:

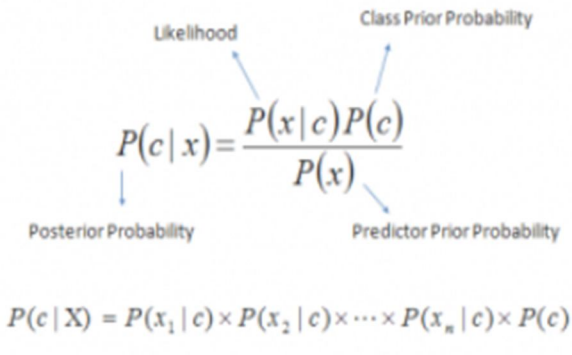


Fig1: Formula for Naïve Bayesian [3]

Here,

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue VII, July 2017- Available at www.ijraset.com

1) P(c|x) is the posterior probability of class (target) given predictor (attribute).
2) P(c) is the prior probability of class.
3) P(x|c) is the likelihood which is the probability of predictor given class.
4) P(x) is the prior probability of predictor.

B. Algorithm for Prediction Via Classifier

for $i = 1 : N$ **do**
  for $c = 1 : C$ **do**
    $L_{ic} = \log \hat{\pi}_c$;
    for $j = 1 : D$ **do**
      if $x_{ij} = 1$ **then** $L_{ic} := L_{ic} + \log \hat{\theta}_{jc}$ **else** $L_{ic} := L_{ic} + \log(1 - \hat{\theta}_{jc})$
  $p_{ic} = \exp(L_{ic} - \text{logsumexp}(L_{i,:}))$;
  $\hat{y}_i = \text{argmax}_c \, p_{ic}$;

Fig2: Algorithm for Naïve Bayesian Classification [8].

Where,

$$\pi_c = \frac{N_c}{N}$$

This is the frequency that class C appears in the training examples.

$$\hat{\pi}_c = \frac{N_c}{N}$$

Adding a hat indicates that this frequency is to be used as an estimate of the probability of class C appearing in the population as a whole.
In terms of Naïve Bayes, we can see these probabilities as priors.

$$\hat{\theta}_{jc} = \frac{N_{jc}}{N_c}$$

This is an estimate of the probability of the j^{th} feature appearing when you restrict your attention to class C. These are the conditional probabilities: P(j | c) = probability of seeing feature j given class c -- and the *Naive* in Naive Bayes means that we assume they are independent.

C. Code Logic for Python

```
#Import Library from sklearn.naive_bayes import GaussianNB
#Assumed you have, X (predictor) and Y (target) for training data set and x_test(predictor) of test_dataset
# Create SVM classification object model = GaussianNB() # there is other distribution for multinomial classes like Bernoulli Naive Bayes, #Train the model using the training sets
check_score
 model.fit(X, Y)
#Predict Output
predicted= model.predict(x_test)
```

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue VII, July 2017- Available at www.ijraset.com

D. Code Logic for R Programming

```
library(e1071)
x <- cbind(x_train,y_train)
# Fitting model
fit <-naiveBayes(y_train ~ ., data = x)
summary(fit)
#Predict Output
predicted= predict(fit,x_test)
```

E. Calculation Factors for Prediction

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Fig3: Example of a data set [8].

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Fig4: Frequency of training data [8].

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

Fig5: Example of a problem statement [8].

This frequency table has been achieved by viewing and tabulating values from Fig 4
This table calculates the necessary ratios for Bayesian classification used for prediction on testing data base.
The above table is a sample weather Meta data as an example.

F. Prediction Performance

From the above example, we can calculate a few factors for performance with which one can understand the wide acceptance of this very algorithm where one can find a few factors with values while conducting this experiment.
This Algorithm is widely used in the industry so far and hence has been tested on various fronts for which we have tabulated these Vectors of measurement [7].

Matrix used	Confusion matrix
Accuracy	57.14%
Precision	66.67%
Recall	66.67%
AUC overall	0.481

Table.1: Performance tabulation for Naive Bayesian classification

The algorithm is pretty well suited for most applications wherein Data mining and Business intelligence uses Machine learning to predict the tuples.

However this in not the most indigenous solution we have as far as technology has been transforming we too have come up here with our own experimental solution to this regard where we can use an alternate Algorithm in place of the widely used Naïve Bayesian algorithm which has better performance and simple to use features along with better prediction statistics.

The way in which we can separate out high cost prediction and low cost prediction is by using a complete new algorithm which has a constant rate at learning and Predicting values and that being the "C- constant" For the SVM or the SVR is very useful.

Let us learn more about our proposed new Algorithm, SVM nearest neighbour Algorithm for Prediction.

V. PROPOSED ALGORITHM FOR PREDICTION ANALYSIS SVM HIGHEST PROBABILITY NEAREST NEIGHBOUR ALGORITHM

A. Definition

It is a classification method. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate.

For example, if we only had two features like Height and Hair length of an individual, we'd first plot these two variables in two dimensional space where each point has two co-ordinates (these co-ordinates are known as Support Vectors)

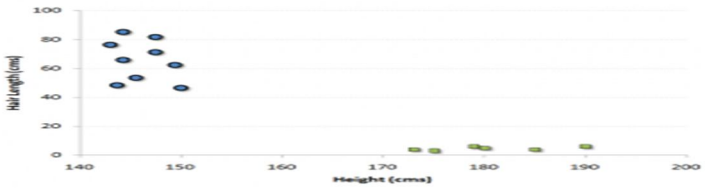


Fig6: SVM Plotting Example for a random data set [1]

In this sample data let us find a line that splits the data between the two differently classified groups of data. This will be the line such that the distances from the closest point in each of the two groups will be farthest away.

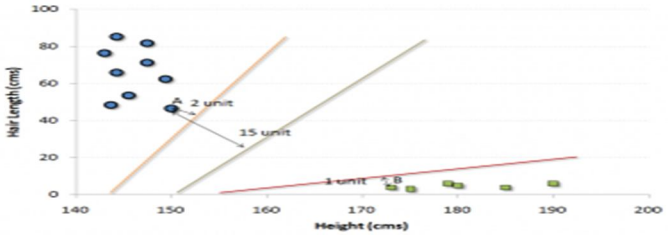


Fig7: SVM plotting for multiple regression line for a random data set [1]

In the example shown above, the line which splits the data into two differently classified groups is the black line, since the two closest points are the farthest apart from the line. This line is our classifier. Then, depending on where the testing data lands on either side of the line, that's what class we can classify the new data as.

To understand the definition better, let us follow why we are proposing this new Algorithm better than the ones currently being used.

VI. PROPOSED NEW ALGORITHM

A. Algorithm

The Highest Probability nearest Neighbour Classifier

1) Required: Sample X; Training set T= {(X1, Y1), (X2, Y2),.....(Xn,Yn)}

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue VII, July 2017- Available at www.ijraset.com

Set of possible values as neighbour K {K1, K2, …, Kn}
Parameter C for Error balancing

2) Ensure: Decision Y belongs to (-1, 1)

Step1: Start Algorithm
Step2: Order the training tuples by value of K(Xi,Xi) – 2*K(Xi,X) in ascending order.
Step3: MinError=1000; Value=0;
Step4: if first K1 values are all from the same class C then
Return C
end if
Step5: for all K do
Train SVM model on the first K training samples in the ordered list.
Classify x using SVM model with equal error costs and get result Yp
Classify the same training samples using this model.
Fit the parameters A and B for the estimation of P(Y=1| Y=p)
ErrorNeg= P(Y=1| Y=p); ErrorPos=1- P(Y=1| Y=p)
If ErrorPos<MinError then
MinError=ErrorPos; Value=1;
End If
Step6: If ErrorNeg * C<MinError then
MinError=ErrorNeg *C; Value=-1;
End If
End for
Step7: Return Value
Step8: End

VII. CODE LOGIC FOR PYTHON

```
#Import Library
from sklearn import svm
#Assumed you have, X (predictor) and Y (target) for training data set and x_test(predictor) of test_dataset
# Create SVM classification object
model = svm.svc() # there is various option associated with it, this is simple for classification.
# Train the model using the training sets and check score
model.fit(X, y)
model.score(X, y)
#Predict Output
predicted= model.predict(x_test)
```

VIII. CODE LOGIC FOR R PROGRAMMING

```
library(e1071)
x <- cbind(x_train,y_train)
# Fitting model
fit <-svm(y_train ~ ., data = x)
summary(fit)
#Predict Output
predicted= predict(fit,x_test)
```

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue VII, July 2017- Available at www.ijraset.com

IX. PREDICTION PERFORMANCE AND VECTORS

This new Algorithm has been seen very capable of outnumbering the common Algorithms namely, Naïve Bayesian, Decision Tree and K-means algorithm. This new Algorithm makes decisions not only based on patterns found repetition in the training data set but also makes use of C-constant. The SVM highest probability nearest neighbour algorithm uses 3 sectional cuts, the soft boundary and the hard boundary as well.

The Algorithm was used for various prediction models and for weather forecasting we found out it was way better than the rest of the algorithms used earlier.

We have already seen how useful the Naïve Bayesian algorithm is and also have studied its performance vectors lets view the performance parameters regarding this unique algorithm.

C constant	Forecasting performance (%)
0.1	71.5
0.2	69.1
0.3	69.1
0.4	68.9
0.5	68.7
0.6	68.4

Table2: The performance mapping of the algorithm alongside its learning via C constant

From the above statistical table one can understand the basic difference of this algorithm working better in regards to other parametric based learning algorithms is that the C-constant is something which is mapped and learned when a data set has enough samples to plot many cases to easily identify a test data entry as to which sample of the learned result does it fall [2]. The data is usually demarked via 3 sample lines along the training data which easily demark types of cases, all of which is explained up at the definition of the algorithm.

Further let's understand the analysis of why we are asserting this algorithm efficient enough to be used for industry and research applications.

X. ANALYSIS OF EXISTING AND PRESCRIBED ALGORITHMS

The final analysis of the above mentioned algorithm is only possible and valid when we understand that the algorithm is tested and validated on a testing database and we find the results better than the normal or frequently used algorithms.

The predicted output of this very hybrid algorithm is to be better at predicting the amount of rainfall in a particular region over the seasons and months, also the prediction of what is the average rainfall for the next year's monsoon season. The algorithm also being proposed here as a hybrid algorithm works on the advantages of both the SVM and the K- nearest neighbour algorithm. The advantage of C constant is that the holistically learning of the machine not only allows it to learn the patterns followed in the structured learning databases and allows it to self-learn the new trends followed in such a unpredicted set of data the machine needs to learn.

Having said that it's difficult we still have analysed many cases where we can with authority assert that the algorithm we are proposing

XI. ANALYSIS OF OUTPUT PROPOSED ALGORITHM VS EXISTING ALGORITHM

In inference to the analysis both the algorithm works on the databases fed into the system by various users, here we are presenting a general overview of the system by predicting various data records over the past and then following the unsupervised learning [5]. Unsupervised learning principle follows the self-learning process where the system accepts pattern of user information and the query fired on the system to produce desirable outputs. Now moving further, we would first provide findings as to why the said algorithm (Proposed) is superior for prediction and unsupervised learning.

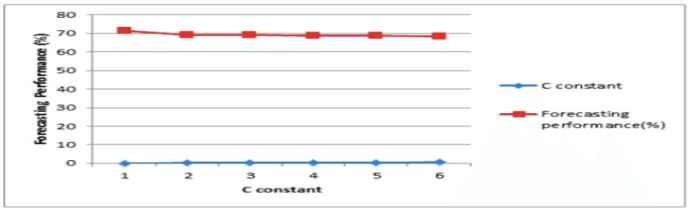


Fig8: C-constant mapping as parameter of success

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue VII, July 2017- Available at www.ijraset.com

The above learning Vs Prediction chart analysis for the C – Constant based learning which is a unique approach followed by the Algorithm.
The machine has a very high learning rate for unsupervised data which indeed is the need of the hour.

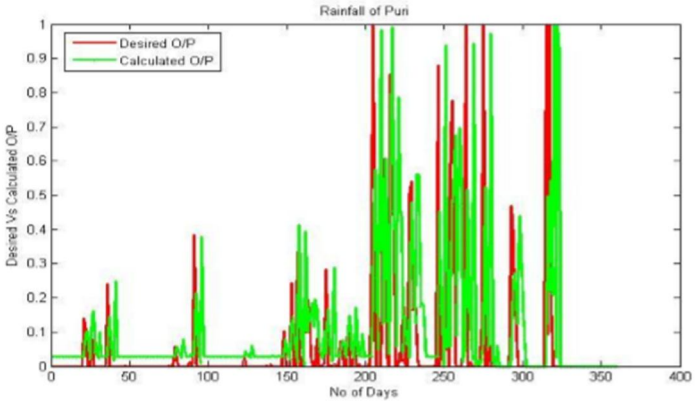


Fig9: Prediction of algorithm with real data set

The graph here is the actual data set mapping of the rainfall at Puri (Maharashtra) using this very algorithm where one can easily notice that the prediction analysis is as good as the supervised learning methods.

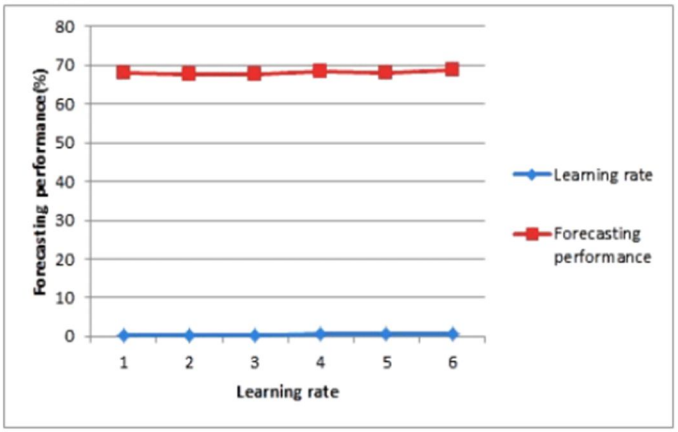


Fig10: Learning rate vs Prediction rate mapping

The above graph is the learning to prediction rate measure of the famous Naïve Bayesian classifier, as being the supervised learning method it grows only after understanding a certain sets of sample to perfection.

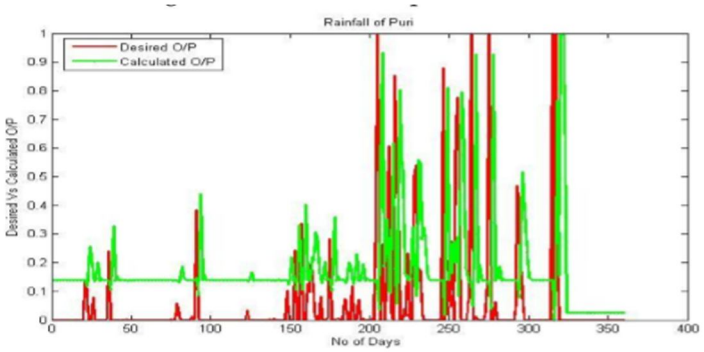


Fig11: Prediction vs Actual plotting of data

The above graph is plotted to the actual data of the rainfall in Puri (Maharashtra) using the Naïve Bayesian method, here one can see prediction analysis of the system only started after learning a certain amount of data as per the learning graph.

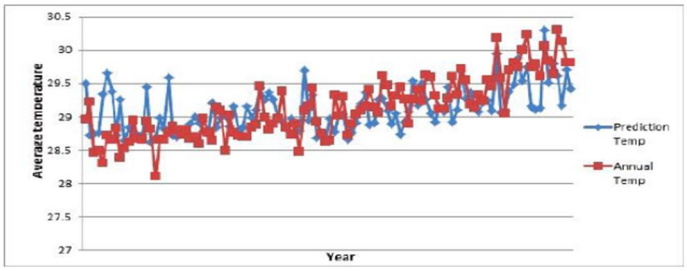


Fig12: Annual average prediction plotting for SVM-HPNN

The above graph is the average temperature analysis of predicted vs actual temperature. The above being the proposed method the high success ratio is clinically achieved.

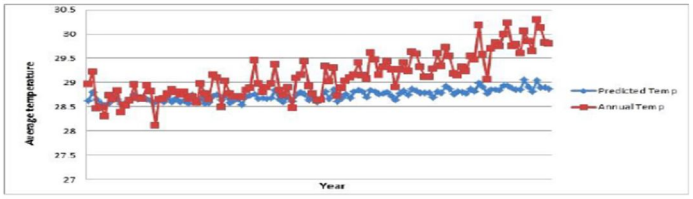


Fig13: Annual average prediction plotting for Naïve Bayesian

The above is the prediction vs actual annual temperature chart of the naïve Bayesian algorithm used for probability methods as numerical prediction method. This has its own minute drawbacks solved in our proposed algorithm.

XII. CONCLUSION

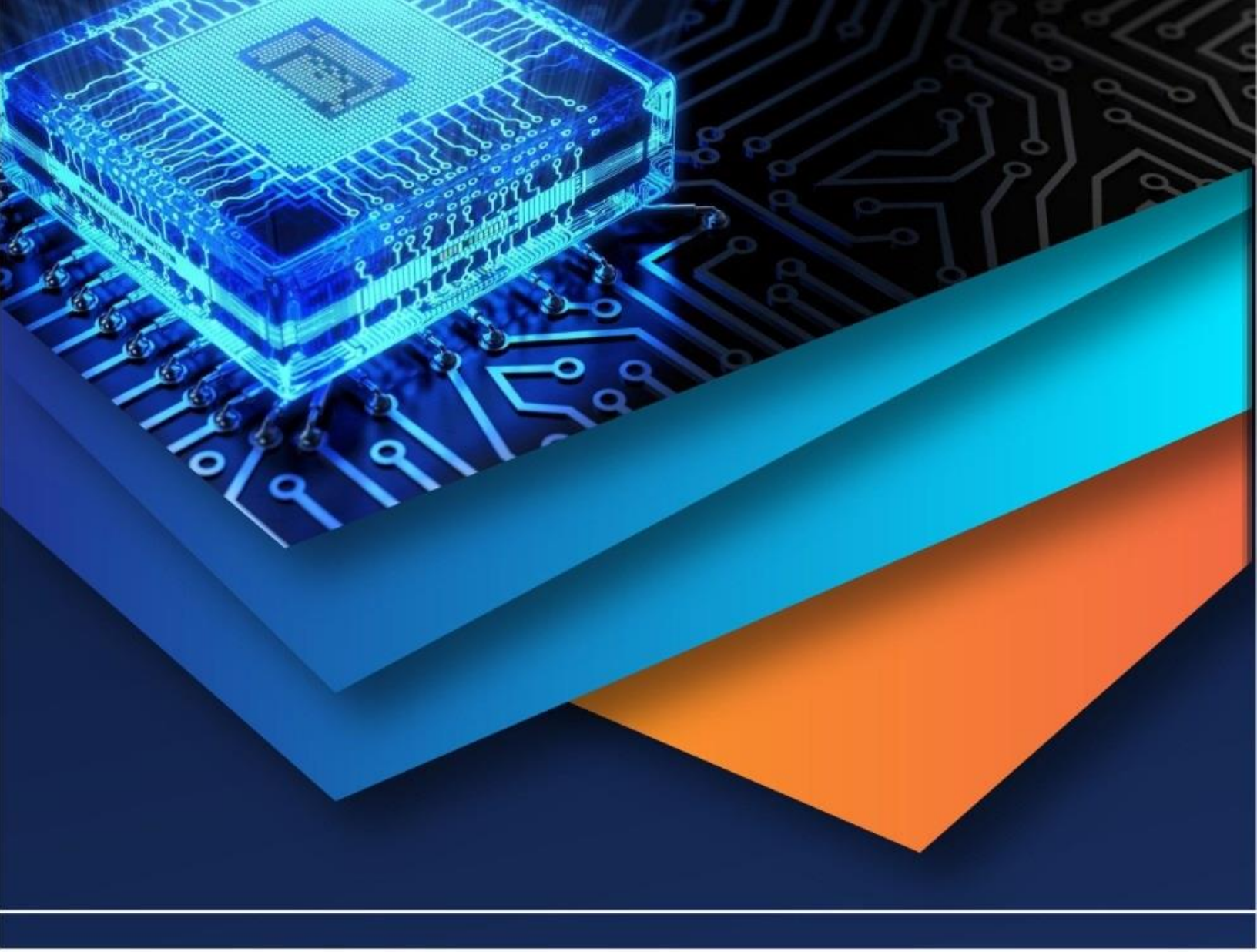
The graphs and tables pertaining to the information presented via this paper is a contribution towards improving the prediction analysis of an intelligent system. The proposed algorithm SVM Highest Probability Nearest Neighbour Algorithm (SVM - HPNN) is efficient and useful in allowing the system learn its data point's right from the beginning and allow the system to function efficiently.

The machine is not expected to work exactly on the basis as per how humans predict on the basis of reading charts but is expected to quote appropriate results as and when new charts prepared are fed to the system the system currently used requires the engineers to even bifurcate the data into labelled sets which is not expected here while using the SVM- HPNN Algorithm.

The further improvements to this method are possible while curating modules using this very algorithm and having certain methods embedded in R Programming language, which will allow further simplification of this algorithm and industry read for mass utilization

REFERENCES

[1] Hearst, Marti A, "Support vector machines." Intelligent systems and their applications, IEEE 13.4 (1998): 18-28
[2] Radhika Y, M Shashi, "Atmospheric Temperature prediction using support vector machines", International Journal of Computer theory and Engineering 1.1 (2009): 1793-8201
[3] Rish , Irina. "An empirical study of the naive Bayes classifier." IJCAI 2001 workshop on empirical methods in artificial intelligence. Vol. 3. No. 22. 2001. FLEX Chip Signal Processor (MC68175/D), Motorola, 1996.
[4] T. Euler. Publishing Operational Models of Data Mining Case Studies. In Proceedings of the Workshop on Data Mining Case Studies at the 5th IEEE International Conference on Data Mining (ICDM), pages 99–106, Houston, Texas, USA, 2005.
[5] Witten, Ian H., et al. "Weka: Practical machine learning tools and techniques with Java implementations." (1999).
[6] A. E. Raftery, T. Gneiting, F. Balabdaoui, and M. Polakowski. Using Bayesian model averaging to calibrate forecast ensembles. Monthly Weather Review, 133:1155–1174, 2005.
[7] J.M.Sloughter, A.E.Raftery, T.Gneiting, and C.Fraley. Probabilistic quantitative precipitation forecasting using Bayesian model averaging. Monthly Weather Review, 135:3209–3220, 2007.
[8] J. M. Sloughter, T. Gneiting, and A. E. Raftery. Probabilistic wind speed forecasting using ensembles and Bayesian model averaging. Journal of the American Statistical Association, 105:25–35, 2010.
[9] C. Fraley, A. E. Raftery, and T. Gneiting. Calibrating multi-model forecasting ensembles with exchangeable and missing members using Bayesian model averaging. Monthly Weather Review, 138: 190–202, 2010.
[10] F.A.EckelandC.F.Mass. Effective mesoscale, short range ensemble forecasting. Weather and Forecasting, 20:328–350, 2005.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY