



Framework of Hybrid Classification Model as Smart Agent in Distributed Systems for Network Monitoring

¹Henry Alexander. I, ²Dr.R.Mallika

¹Research Scholar, Karpagam University, Coimbatore, India

²Assistant Professor, Department Of Computer Science, CBM College, Coimbatore

Abstract---The main objective of this study is to compare SVM (Support Vector Machine) classification algorithm and Neural Networks classification algorithms identify the pitfalls and propose a new hybrid classification algorithm which is reliable, fast, efficient and robust handling large data sets. A new version of Support Vector Machine algorithm is designed and developed in the study which is used for training the dataset followed by testing using Neural Networks classification algorithm. Generally most of the classification algorithms are working well for small and moderate data, while going for large datasets, the efficiency drops, this study analyses all these factors and proposing a new hybrid model, which solves all these drawbacks. The second main objective of this study is to analyze whether the number of back propagation steps are minimized in Neural Networks algorithm. This model can be implemented as smart agents to monitor and predict the networking issues that occur in distributed systems.

Keywords--- SVM, Neural Networks. Kernel, Gradient, Hessian. Hybrid, Distributed Systems, Smart Agents.

1. INTRODUCTION

A. Datamining

Datamining (sometimes called data or knowledge discovery) is a process of analyzing data from different perspective and summarizing the result into useful information.

B. Classification

Maps data into predefined groups or classes

- Supervised learning
- Pattern recognition
- Prediction

C. Clustering

Group's similar data together into clusters.

- Unsupervised learning
- Segmentation
- Partitioning

D. Regression

Is used to map a data item to a real valued prediction variable.

E. Frequently Used Classification Algorithms.

- Distance Vector Algorithm
- Rot Boost Ensemble Technique
- Simple Bayesian Classifier
- Support Vector Machine For classification
- Decision Tree Based Algorithm
- Back Propagation

F. Frequently Used Clustering Algorithms

- K-means
- Fuzzy C-means

- Hierarchical clustering
- Mixture of Gaussians

2. MODEL

Support Vector machines serves as a accurate model for learning large datasets which requires big memory capacity and long time.. Due to increase in the size of the databases in recent years, need to extract knowledge from large databases is increasing. KDD (Knowledge Discovery in Databases is a non-trivial process of identifying , valid potential and understandable patterns in data. Data mining is a particular pattern recognition task in the KDD process. We are interested in SVM algorithm proposed by Vapnik due to its relevance for classification and regression. SVM has been implemented in various fields like face identification, text categorization and bioinformatics. SVM are the most well-known algorithms of a class using the idea of kernel function.SVM solutions are obtained from quadratic programs (QP), so that the computational coast of an SVM is least square of the number of training data points and the memory requirement making SVM impractical. There is need to scale up the algorithm to handle massive datasets by dividing the original quadratic points into series of small problems incremental learning updating solution in growing training set.

A. Objectives

We have created a new algorithm that is very fast for building incremental, parallel and distributed SVM classifiers. A new kernel function is proposed and there by using this kernel function a new algorithm is proposed and designed to classify larger dataset for accuracy and efficiency. This algorithm can classify large data point's two million data points in 20-dimensional input space into two classes. This study uses the following notations in the proceedings. The inner dot product of two vectors x,y denoted by $x.y$. The 2-norm form of the vector x will be denoted by $\|x\|$. The matrix $A[m \times n]$ will be m data points in the n -dimensional real space R_n . The classes $+1, -1$ Of m data points are denoted by the diagonal matrix $D[m \times n]$ of $-1, +1, e$ will be the column vector of 1. W, b will be the coefficients.

f = function
w=coefficient of hyper plane
b = scalar variable
z = slag variable
D= diagonal matrix denoting class labels
A – m x n matrix
C = Constant

Figure 1 Derivatives used in Kernel Function and Algorithm.

B. Overview of the Proposed Mechanism

We propose a new hybrid model which is a combination of Support Vector Machine and Neural Networks. We classify the data using SVM and test the dataset with Neural Networks. We have created a new SVM algorithm which is fast and efficient and helps to classify larger datasets. The following section explains about the new modified Support Vector Machine Algorithm. Data points are classified with this algorithm and tested with Neural Networks Model.

C. Modified Support Vector System

Design of New Kernel Function

A comparative study of various kernel functions were made, whose drawbacks lead to the design of the new kernel function

Table 1 .Different Kernel Functions and their Drawbacks

S.No	Kernel Function	Drawbacks
1	Gaussian Function	Efficiency drops for larger datasets
2	Polynomial Function	Efficiency drops for larger datasets
3	Radial Basis Function	Efficiency drops for larger datasets
4	Fisher Function	Efficiency drops for larger datasets
5	Graph Function	Efficiency drops for larger datasets
6	String Function	Efficiency drops for larger datasets

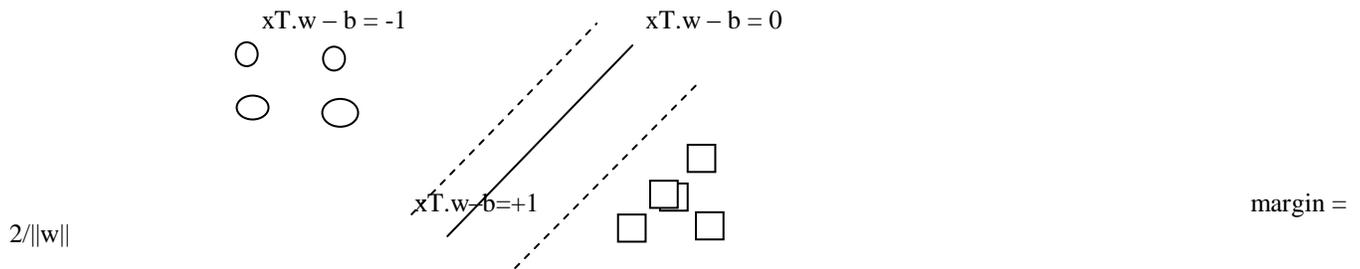


Figure 2 Linear separation of the data points into two classes

Considering a binary classification as shown in figure 2, with m data points in the n dimensional input space R^n with $m \times n$ matrix. The SVM tries to find the best separating plane for classes $+1$ and -1 . As per the standard algorithm it can maximize the distance between the supporting planes by the formulas $(x^T \cdot w \cdot b = +1)$ for class $+1$ and $(x^T \cdot w \cdot b = -1)$ for class -1 with the margins between the separating planes as $2/\|w\|$. Where $\|w\|$ is the 2 normal form of vector w . Any data point falling on wrong side of the plane is considered to be an error having corresponding slag variable $z > 0$. Therefore the SVM has to maximize the margin distance and minimize the error, so this study focuses on developing a new kernel function

The standard SVM formulae for a linear kernel function is given by the formulae

$$\begin{aligned} \text{Min } f(w, b, z) &= C \sum z + (1/2) \|w\|^2 \\ D(Aw - eb) + z &> -e \end{aligned} \quad (1)$$

Then, the classification function of a new data point x based on the plane is **predict(x) = sign(w.x-b)**. If the sign is positive, corresponds to right classification else wrong classification. Frequently used kernel functions and their pitfalls are listed below.

Now to minimize the complexity this study the kernel function is minimized to a function of one variable as per the sequence of steps shown below. Thus first order and second order derivatives of this function is correspondingly used in the new algorithm for faster execution and to support larger datasets. Now assuming the slag variable $z > 0$ and constant $C > 0$ to tune the errors and margin size equation can rewritten as

$$z = e - D(Aw - eb) \quad (2)$$

Now substituting the value of z in equation we get

$$\text{Min } f(w, b) = (C/2) \| (e - D(Aw - eb)) \|^2 + (1/2) \| w, b \|^2 \quad (3)$$

The equation can be again minimized as a function of one variable (i.e) $\text{Min } f(w)$, which still minimizes the complexity of the kernel function.

Thus taking $D(Aw - eb)$ into consideration and calculating for b we get as follows

$$\begin{aligned} D(Aw - eb) &= K \\ Aw - eb &= K/D \end{aligned} \quad (4)$$

$$b = 1/e (Aw - K/D) \quad (5)$$

By setting $1/e (Aw - K/D)$ to M equation 3 becomes

$$\text{Min } f(w) = (C/2) \| (e - D(Aw - eM)) \|^2 + (1/2) \| w \|^2 \quad (6)$$

By setting $[w_1 \ w_2 \ \dots \ w_n]^T$ to u and $[A \ -e]$ to H , then the SVM formulation (4) is rewritten by :

$$\text{Min } f(u) = (C/2) \| (e - DHu) \|^2 + (1/2) u^T u \quad (7)$$

Thus by using the above developed kernel function and correspondingly taking the first order and second order derivatives for the function for each data point, the efficiency and accuracy can be obtained for larger datasets. The algorithm can be described as the algorithm 1.

$\Delta f(\mathbf{u})$ **Hessian matrix** or **Hessian** is a square matrix of second-order partial derivatives of a function. It describes the local curvature of a function of many variables.

$\delta^2 f(\mathbf{u})$ **Gradient** of a scalar field is a vector field that points in the direction of the greatest rate of increase of the scalar field.

D. Preprocessing Steps for the Hybrid Model

- i. Selection of Databases (Diabetes, Cancer).
- ii. Use 60% of data for training, 40% for testing
- iii. Train the data using SVM algorithm
- iv. Test for accuracy.
- v. Input the trained data to Neural Networks.
- vi. Gradient and Hessian are calculated for all vector elements for minimize the error until the gradient is zero

- Input: training dataset represented by A and D matrices
 - Starting with $u_0 \in R^{n+1}$ and $i = 0$
 - Repeat
 - Move First Row to Buffer (For faster Execution)
 1) $u_{i+1} = u_i - \delta^2 f(u_i) - \Delta f(u_i)$
 2) $i = i + 1$
 Until $\Delta f(u_i) = 0$
 - Return u_i
Calculation of gradient f at u_i ,
 $\Delta f(u_i) = C(-DH)^T(e - DHu_i) + u_i$ (5)
Calculation of generalized Hessian of f at u_i ,
 $\delta^2 f(u_i) = C(-DH)^T \text{diag}([e - DHu_i]^*) (-DH) + I$ (6)
 with $\text{diag}([e - DHu_i]^*)$ denotes the $(n+1) \times (n+1)$ diagonal matrix whose j th diagonal entry is sub-gradient of the step function $(e - DHu_i)_+$.

Figure 3. Pseudo Code for Modified Support Vector Machine Algorithm [1]

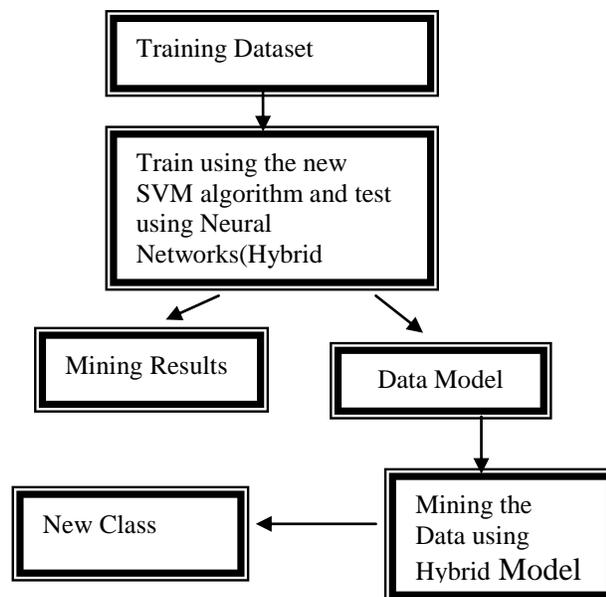


Figure 4. Proposed Model

As mentioned, the input data (training data) is acquired preprocessed and sent as input to the new Hybrid Model, a combination of SVM (Support Vector Machine) and Neural Networks. If the model exists, data is classified using the model

else a new hybrid model is designed. and accordingly data is trained using SVM and tested using Neural Networks .A new class is generated.

```
- Input: training dataset represented by  $A$  and  $D$  matrices
- Starting with  $u0 \in R^{n+1}$ 
Use SVM algorithm and get the required training dataset  $u_i$ .
Let  $u_i$  be the input to neural networks
1. Initialize the weights in the network (often randomly)
2. repeat
* for each example  $u_i$  in the training set do
1.  $O = \text{neural-net-output}(\text{network}, u_i)$  ;
Forward pass
3.  $T = \text{teacher output for } u_i$ 
4. Calculate error  $(T - O)$  at the output
Units (Normally this difference will be negligible when
comparing with ordinary Neural Networks Algorithm).
If needed follow the following steps for back propagation
5. Compute  $\delta_{wi}$  for all weights
from hidden layer to output layer ;
backward pass
6. Compute  $\delta_{wi}$  for all weights
from input layer to hidden layer ;
backward pass continued
7. Update the weights in the network
* end
8. until all examples classified correctly or
Stopping criterion satisfied
```

Figure 5 Pseudo Code for Hybrid Algorithm

1. Present a training sample to the neural network.
2. Compare the network's output to the desired output from that sample. Calculate the error in each output neuron.
3. For each neuron, calculate what the output should have been, and a *scaling factor*, how much lower or higher the output must be adjusted to match the desired output. This is the local error.
4. Adjust the weights of each neuron to lower the local error.
5. Assign "blame" for the local error to neurons at the previous level, giving greater responsibility to neurons connected by stronger weights.
6. Repeat the steps above on the neurons at the previous level, using each one's "blame" as its error.

E. Advantages of this algorithm.

- Data is well trained since we are using combination of two algorithms.
- Classification is fast and accurate.
- Able to handle large datasets.
- Weight matrix is minimized.
- No need for much weight adjustment during back propagation.
- We end in a accurate classification
- Error rate is negligible.

F. Implementation of Hybrid Model in Distributed Systems.

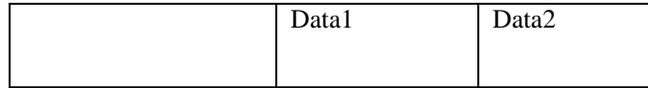
Distributed System

A distributed system is a system consisting of computers that do not share a common memory or a synchronized clock. The computers in a distributed system are connected via a communications network.

In the study classifying various networking issues and labeling the tuples, which helps to analyze and predict the problems which may happen in future. Various predefined networking issues are collected on the following issues

1. Server Crash
2. Network
3. Cable failure
4. Networking
5. Equipment's

Congestion



fault

Using these predefined data a model is designed. Smart Agents acting on various servers will collect the data

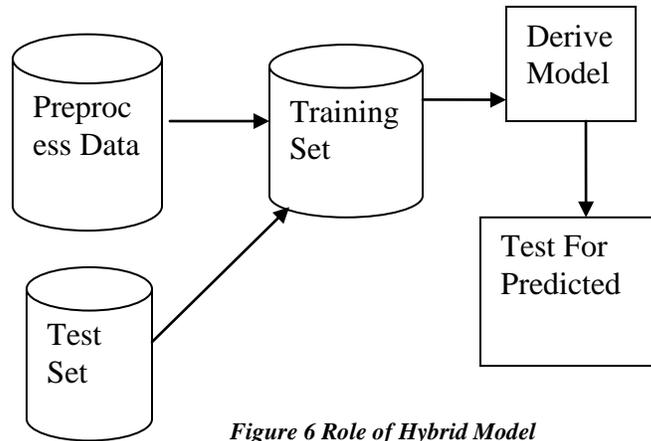


Figure 6 Role of Hybrid Model

G. Role of this Model.

This hybrid model will be installed in every server in distributed environment. Based on the predefined dataset, the model will tests whether any issues in the distributed systems matches the predefined dataset, if it matches the system takes appropriate action to resolve the problems. The model also predicts and forecast whether any networking issues might occur in future and helps to avoid the networking disasters that may occur in future. Since it acts intelligently according to the environment it acts as a smart agent.

H. Steps for Implementation of the Model

- Initialize all the servers and deploy the smart agents
- Develop appropriate routing algorithm to identify the problems in the distributed environment.
- Develop suitable GUI to visualize the problems by the user.
- Use the Hybrid Model to classify the problems.
- Based on the problems classified, adopt suitable visualization techniques to display the problems that may occur in future

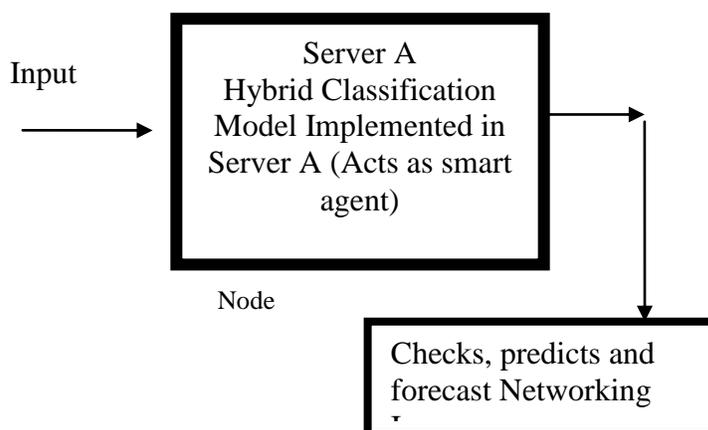


Figure 7. Implementation of Hybrid Algorithm in a single node in Distributed Environment

3. ANALYSIS AND DISCUSSION

Table 2 Testing Results for different data size

Training Set	10^4	10^6
Testing Set	10^3	10^5
Training time (s)	0.025	2.585
Accuracy (%)	97.25	93.36%

Table 3. Confusion Matrix

A
c
t
u
a
l
C
l
a
s
s

Predicted Class

	C1	C2
C1	True Positive	True Negative
C2	False Positive	True Negative

A. Performance Metrics

Sensitivity = t_pos / pos

Specificity = t_neg / neg

Precision = $t_pos / (t_pos + t_neg)$

Accuracy = $sensitivity * pos / (pos + neg) + specificity * neg / (pos + neg)$

Sensitivity (also called the *true positive rate*, or the recall rate in some fields) measures the proportion of actual positives which are correctly identified as such (e.g. the percentage of sick people who are correctly identified as having the condition).

Specificity measures the proportion of negatives which are correctly identified as such (e.g. the percentage of healthy people who are correctly identified as not having the condition, sometimes called the *true negative rate*).

Accuracy of a measurement system is the degree of closeness of measurements of a quantity to that quantity's actual (true) value.

Precision of a measurement system, also called reproducibility or repeatability, is the degree to which repeated measurements under unchanged conditions show the same results

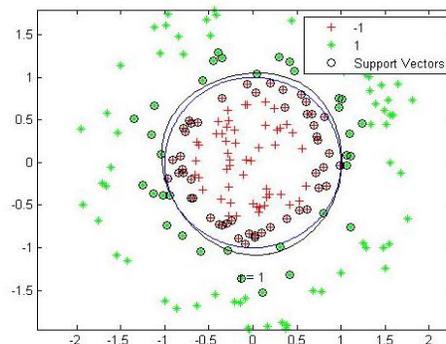


Figure 8. Plotted in Graph Using Hybrid Algorithm

4. CONCLUSION

Thus this model helps to solve the problems listed below

- One it removes the drawbacks of conventional Support Vector Machine algorithm using either of the Kernel functions listed in the study.
- The algorithm helps in accuracy and efficiency measures for larger datasets.
- Second advantage of this algorithm is, it minimizes the number of back propagation in Neural Networks algorithm which reduces the time and increases the efficiency in classifying data.
- This algorithm is fast, efficient and accurate for larger datasets.

REFERENCES

- [1] A Simple, Fast Support Vector Machine Algorithm For Data Mining Hiep-Thuan Do, Nguyen-Khang Pham, Thanh-Nghi Do College of Information Technology, Cantho University 1 Ly Tu Trong Street, Ninh Kieu District Cantho
- [2] Improvements to the SMO Algorithm for SVM Regression S. K. Shevade, S. S. Keerthi, C. Bhattacharyya, and K. R. K. Murthy .E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers.
- [3] Multi Agent Based Approach For Network Intrusion Detection Using Data Mining Concept Ankita Agarwal1, Sherish Johri2, Ankur Agarwal3, Vikas Tyagi4, Atul Kumar5 1,2,4,5 IMSEC, Ghaziabad DCMS, Muzaffarnagar. **Volume 3 No 3, March- 2012**
- [4] U. Fayyad, D. Haussler, and P. Stolorz, "Mining scientific data," Communications of the ACM, Vol. 39, 1996, pp. 51-57.
- [5] J. Zhang, W. Hsu, and M. L. Lee, "Image mining: Issues, frameworks and techniques," in Proceedings of the 2nd International Workshop Multimedia Data Mining, 2001, pp. 13-20.
- [6] B. Nagarajan and P. Balasubramanie, "Cluttered Background Removal in Static Images with Mild Occlusions" International Journal of Recent Trends in Engineering, Vol. 1, No. 2, May 2009
- [7] C. Ordonez and E. Omiecinski, "Image mining: A new approach for data mining," Technical Report GIT-CC-98-12, College of Computing, Georgia Institute of Technology, 1998.
- [8] B. Nagarajan and P. Balasubramanie, "Neural Classifier for Object Classification with Cluttered Background Using Spectral Texture Based Features" Journal of Artificial Intelligence, 2008, ISSN 1994-5450
- [9] A. Vailaya, A. T. Figueiredo, A. K. Jain, and H. J. Zhang, "Image classification for Content-based indexing," IEEE Transactions on Image Processing, Vol. 10, 2001, pp. 117-130.
- [10] R. F. Crompt and W. J. Cambell, "Data mining of multidimensional remotely sensed images," in Proceedings of the 2nd International Conference on Information and Knowledge Management