SACRED HEART RESEARCH PUBLICATIONS

Journal of Computing and Intelligent Systems

Journal homepage: www.shcpub.edu.in

ISSN: 2456-9496

ANALYSIS ON MACHINE LEARNING BASED FEATURE SELECTION TECHNIQUES IN INTRUSION DETECTION

P.Vijayalakshmi #1, **P.M.Gomathi** #2

Received on 24th DEC 2021, Accepted on 03 FEB 2022

Abstract — An Intrusion Detection System (IDS) is a network technology it is built for detect the susceptibility of attacks in the network environment. In which intrusion detection system play the vital role in the attack. Intrusion Prevention System (IPS) extended the detection solutions by adding the facility to block threats in addition to detect them and it has become the deployment option for IDS/IPS technologies. An optimization algorithm is functions which are executed repeatedly by comparing the various outcomes till an optimum or reasonable solutions is identified with the help of the optimization result an effective feature selection can be done. In this paper, an analysis of feature selection in the intrusion detection using optimization algorithm such as support vector machine, random forest and whale optimizer techniques in order to detect the intrusion detection based on recognized attack patterns.

Keywords - Support Vector Machine, Random Forest, Whale Optimizer, Vulnerability

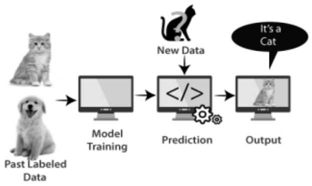
I. INTRODUCTION

Intrusion Detection System is the major aspect for the data transfer in the network in which IDS play a major role in preventing the attack which taken place in the network. Now a day's intruders will cause the hacking in the network environment as a result important data maybe corrupted. Here with the help of IDS it is possible to prevent the data. Feature selection algorithms can be view as the blend of a search technique for proposes new feature subsets. The simplest algorithm in which it is used to test the possible subset for feature selection and finding the best solution and minimum error conditions. Here we analysis the support vector machine, random forest and whale optimizer algorithm for the feature selection and intrusion detection

II.SUPPORT VECTOR MACHINE

Support Vector Machine is the supervised ML algorithm which mainly used for classification and regression problems. The main focus here is the classification problem. In SVM algorithm, each point of the data item was plot individually and n-dimensional space of each features are plot in the particular co-ordinates. There are two types of SVM is available Linear SVM and Non-Linear SVM.

In Linear SVM it is used to plot the dataset separately into two classes by using the single straight line, such dataset are said to be linearly separated dataset because of the data plot in the straight line direction. Non Linear SVM is termed as non linear plot because the data set is not plotted in the straight line direction and so the data here is called as non linear data and also called as Non-Linear SVM classifier. It is easy to understand the SVM algorithm with the best suitable example such as with the KNN classifier. If suppose a strange cat is seen it is needed to identify that it is supposed be cat or dog so in such case a perfect model of cat is created so that we can create a trained data for the strange cat to easy identification of cat such model can be created using SVM algorithm. Before creating the model we used to create lots of trained data in order to identify different features of cat and dog so that it can be easy to learn the cat and dogs, with this we can test the strange image capture. Based on decision taken with the support vector feature identification the solution such as appropriate cat is identified and it is found using the SVM algorithm in which by means of classification and regression the model created using trained datasets. Consider the below diagram.

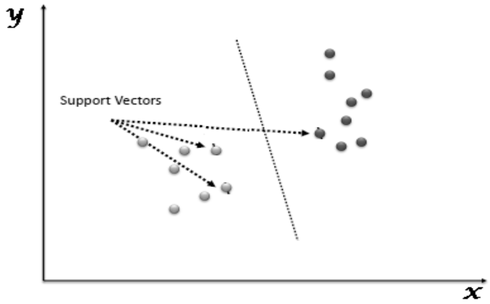


SVM algorithm has the high degree of suppleness in the classification and regression problem with different training dataset. Here is the best example of binary classification of SVM algorithm using the hyper plane to represent the boundary line of the two planes in its classes [1]. The best hyper plane is situated in the middle of the two sets of items of two classes. The best hyper plane is equal to the two different objects at different distance from the different classes.

* Corresponding author: E-mail: [1]vijiperumalsas@gmail.com
, [2] gomathipm@pkrarts.org

[1]Asst.Professor, Department of Computer Science, P.K.R. Arts College for Women, Gobichettipalayam, Erode(D)t, Tamil Nadu, India

[2]Head & Dean, Department of Computer Science, P.K.R. Arts College for Women, Gobichettipalayam, Erode(D)t, Tamil Nadu, India

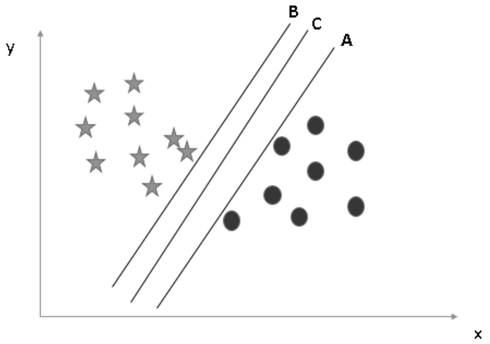


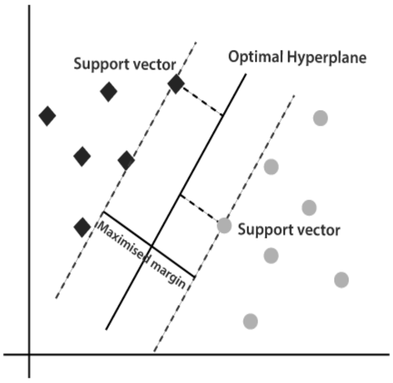
In Scenario 1- Identification of right plane:

Consider an example, We have three plane such as A,B and C. Here two planes is identified with the stars and circles and the best model of hyper plane is found with the thumb rule in which it segregates the two classes better. Considering this scenario the hyper plane B has excellent performance in their position.

In Scenario 2- Identification of right plane:

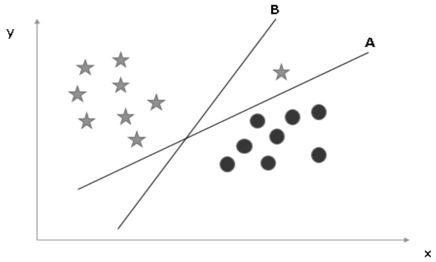
Consider an example, We have three plane such as A,B and C. With the segregating classes the right plane is identified with the help of maximum distance between the nearest data points of the hyper plane the best and right hyper plane is identified. The distance between the two hyper plane is called as Margin [2].





In Scenario 3- Identification of right plane:

Consider an example, We have three plane such as A,B and C. The right hyper plane is identified by the rule such as high margin compared with A. Initially it identify the higher margin hyper plane by accurately identify the classes prior to the maximum margin. Here the plane B has the error in the classification so plane A is classified correctly. Here the right hyper plane is A.



In Scenario 4- Classification of two classes:

In this scenario the separation of two classes using the straight line, where the starts lies outside the border line in this case where the circle lies in the outside the territory so SVM algorithm used to identify the point which is outside the boundary and generally ignore the hyper plane which is outsider and it has the maximum margin

In Scenario 5- Identify the hyper plane which is separated from each other

Here the hyper plane which lies in the straight line and it is said to be the linear in nature so classification scenario is needed to identify the best solution to this problem some additional features is needed to identify the feasible solutions x=z^2+y^2. Now, let's plot the data points on axis

In above plot, points to consider are:

- The values of x would be positive always because x is the sum with z and y and squared with x
- In the plot, the red circle appears close to each other in the origin such as z and y, Lead to the value z is lower and the star is relatively higher than the original results of the value such as x.

1) Linear SVM Example

- *Using the trained SVM model the weight value for the linear equation are found*
- *Get the m-offset for the linear equation*
- *Create the n-axis space for the datasets*
- *Get the m-values to plot the decision boundary*
- *Finally plot the decision boundary*
- *The plot are visually shown*

```
    u = clf.coef_[0]
    a = u[0] / u[1]
    nn = np.linspace(0, 13)
    mm = a * nn- clf.intercept_[0] /u[1]
  plt.plot(XX,mm, 'k-')
    plt.scatter(training_X[:, 0], training_X[:, 1], c=training_y)
    plt.legend()
    plt.show()
```

2) Non-Linear SVM Example

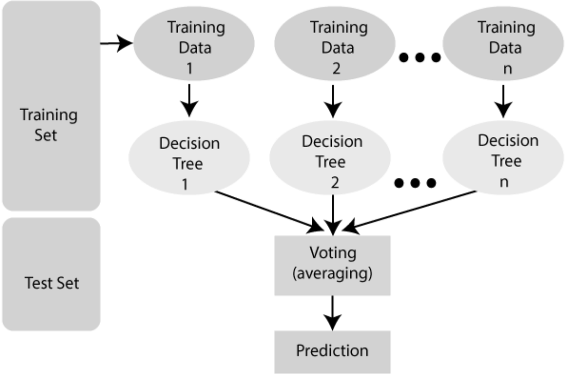
- *For nonlinear SVM problem plot the decision boundary [4]*
- *For evaluate model create the grid*
- *Shape the trained data*
- *Based on the model get the decision boundary and plot the boundary*

```
def plot_decision_boundary(model, bx=None):
if bx is None:
bx = plt.gca()
   xlim1 =bx.get_xlim()
   ylim1 =bx.get_ylim()
   x1 = np.linspace(xlim1[0], xlim1[1], 30)
   y1 = np.linspace(ylim1[0], ylim1[1], 30)
  Y, X = np.meshgrid(y1, x1)
  Xy1 = np.vstack([X.ravel(), Y.ravel()]).T
    P1=model.decision_function(xy).reshape(X.shape)
   ax.contour(X, Y, P1,
          levels=[0], alpha=0.5,
          linestyles=['-'])
```

III.RANDOM FOREST ALGORITHM

Random Forest is the most popular ML algorithm belongs to supervised learning mainly used for the classification and regression problems it is the efficient optimization technique method in order to find the best optimization solutions based the multi programming concept where it combine the multiple classifier and solve the complex problem and by means of it where the performance of the model where improved. Random forest work based on the decision tree. It contains the number of decision tree using various subsets for the given datasets and it also predicts the accuracy of the sample datasets. It will take the decision based on the majority votes of prediction tree using this one decision the centric solution will be found and the final output is accomplished through prediction method. The number of tree in the forest is used to identify the solution so with the help of tree appearance the accuracy will be found. Because of the greater number of tree the problem of over fitting is done.

The working of random forest algorithm is shown in the below diagram:



Random Forest works by two phases such as combining and prediction methods where by combining the decision tree the finite solution will be easily found and here N decision tree were used and by means of second phase through the prediction methods the tree is created and the working process of random forest algorithm are shown in below diagram:

Step-1: From training dataset select the random K value

Step-2: With the help of the selected dataset built the decision tree associated with it

Step-3: Built N number of decision tree

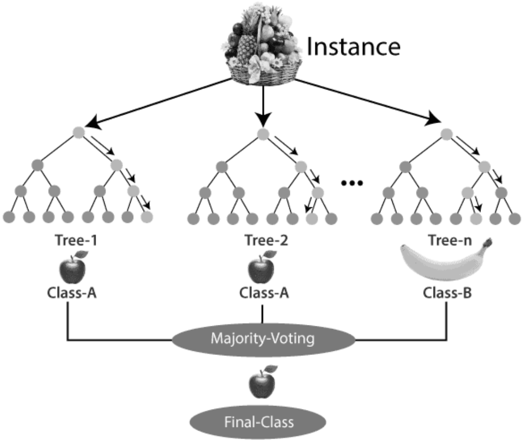
Step-4: Repeat Step 1 & 2.

Step-5: With the help of new data points predict the new decision tree and also assign the new point in which with the help of the new data point the solution can be found

The working of the algorithm will be found by the below examples:

Example: Suppose if we have a huge fruit images so the training datasets were given to the RF classifier and the datasets are easily identified with the help of the decision tree by means of the multiprogramming concept where the small decision tree are identified and then later it is combined with the large datasets In training phase the prediction result will be found by the decision tree and the dataset is divided into their subsets by the given tree by using this decision tree the majority of the result will be found

Application of Random Forest is shown below:



Applications of Random Forest

Four Applications were random forest used:

1. In banking sector the loan risk were identified using RF
2. In medicine disease risk were found through classification
3. In land we can identify the similar land using the RF algorithm
4. In Marketing field where recent trends where identified with the decision tree appearance

Implementation Steps

- In data pre-processing
- Setting the RF algorithm to the training set
- The test result prediction is done
- The accuracy of the result were found by the confusion matrix
- Final result are visualized

1.Data Pre-Processing Step

```
# importing library files
import numpy as nm
import matplotlib.pyplot as mtp
import pandas as pd
#importing training datasets
data_set1= pd.read_csv('user_data1.csv')
#Extracting Independent and dependent Variable      from
the file
w= data_set.iloc[:, [2,3]].values
u= data_set.iloc[:, 4].values
# separating the dataset into training and test set.
 import train_test_split
x_train, x_test, y_train, y_test= train_test_split(w,u, test_siz
e= 0.5, random_state=0)
#Scalling criteria
from sklearn.preprocessing
import StandardScaler
```

```
st_x1= StandardScaler()
x_train1= st_x.fit_transform(x_train1)
x_test1= st_x.transform(x_test1)
```

2. Appropriate Random Forest algorithm for the training set:

```
#Fitting
 import RandomForestClassifier
classifier= RandomForestClassifier(n_estimators1= 10, criteri
on1="entropyset")
classifier.fit(x_train1, y_train1)
```

n_estimators1= The default value is 10 and the required number of tree in RF is necessary to be consider the over fitting

criterion1= Analyze the accuracy of the entropy set by this the best accuracy result will be found

3. Test Set result for the Predicting Value

```
#Prediction the test result
y_pred1= classifier.predict(x_test1)
```

4. Confusion Matrix Creation

```
# Confusion matrix creation
from sklearn.metrics import confusion_matrix  cm1= confusio
n_matrix(y_test1, y_pred1)
```

5. Final result of Training Set are Visualized

```
x_set1, y_set1 = x_train1, y_train1
p1,px2 = nm.meshgrid(nm.arange(start1 = x_set[:, 0].min() -
 1, stop = x_set[:, 0].max() + 1, step  =0.01),
nm.arange(start = x_set[:, 1].min() -
 1, stop1 = x_set[:, 1].max() + 1, step = 0.01))
mtp.contourf(x1, x2, classifier.predict(nm.array([x1.ravel(), x
2.ravel()]).T).reshape(x1.shape),
alpha = 0.5, cmap = ListedColormap(('pink','blue' )))
mtp.xlim(x1.min(), x1.max())
mtp.ylim(x2.min(), x2.max())
for i, j in enumerate(nm.unique(y_set)):
   mtp.scatter(x_set[y_set == j, 0], x_set[y_set == j, 1],
      c = ListedColormap(('purple', 'green'))(i), label = j)
mtp.title('Random Forest Algorithm (Training set)')
mtp.xlabel('Age1')
mtp.ylabel('Salary1')
mtp.legend()
mtp.show()
```

6. Test Result are Visualized

```
#Visulaizing the test set result
from matplotlib.colors import ListedColormap
x_set1, y_set1 = x_test, y_test
x1, x2 = nm.meshgrid(nm.arange(start = x_set[:, 0].min() -
 1, stop1 = x_set[:, 0].max() + 1, step  =0.01),
nm.arange(start1 = x_set[:, 1].min() -
 1, stop1 = x_set[:, 1].max() + 1, step = 0.01))
mtp.contourf(x11, x12, classifier.predict(nm.array([x1.ravel()
, x2.ravel()]).T).reshape(x1.shape),
alpha = 0.5, cmap = ListedColormap(('purple','green' )))
mtp.xlim(x11.min(), x11.max())
mtp.ylim(x12.min(), x12.max())
for j,k in enumerate(nm.unique(y_set)):
   mtp.scatter(x_set[y_set == j, 0], x_set[y_set ==k, 1],
      c = ListedColormap(('pink', 'blue'))(i), label1 = j)
mtp.title('Random Forest Algorithm(Test set1)')
mtp.xlabel('Age1')
mtp.ylabel('Estimated Salary1')
mtp.legend()
mtp.show()
```

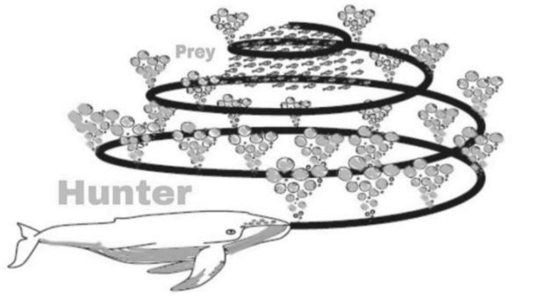
IV.WHALE OPTIMIZATION ALGORITHM (WOA)

The Whale Optimization Algorithm (WOA) it is the best optimization techniques for solving the upcoming networking attacks. This algorithm imitates the three phases such as searching for prey, encircling prey and bubble-net foraging behaviour of humpback whales. Whale Optimizer Algorithm is a meta-heuristics algorithm. The bubble net hunting usually create the bubble shaped structure through the structure the attack taken place this is the social behaviour of the humpback whales. There are seven different types of whale available in which humpback whale are the largest mammals in the oceans. This whale is usually a brilliant whale in which the it create a spiral shaped structure. Through the structure the whale travel unconditionally and the attack taken place this will be accomplished using the searching, encircling and bubble shot. This method is also called as bubble-net feeding method. The Knowledge is based on the spindle cells. Through the formation of special bubbles in the form of a spiral or path.[5]

Whale Leadership Hierarchy is investigated:

Basically the leader whale are the one will finds its prey when it finds the prey it will start diving inside the ocean and creating the spiral shaped bubble by 12 meter around the prey once the bubble shape is formed around the prey once the shape formed in swims up and attack taken place

- An old and well experienced whale will act as a leader whale will indicate the all the other whale regarding the prey and maintain synchronize
- On each lunge the leader will leads the other whale in the formation



Whale Optimization Algorithm and their Mathematical Model

The WOA algorithm imitates the social behaviour and hunting method of humpback whales in oceans

1. Searching for the prey
2. Encircle the prey
3. Examination phase: Through the bubble-net method attack the prey is done

Examination Phase: Searching Model

The search agent (humpback whale) looks for the best solution the prey it is based on the position of the prey. In each phase the position of the phase are updated and its randomly select the search agent for searching the prey instead of selecting the best search agent .Where $X\rightarrow_{randi}$s in the current population the random position vector were selected which is denoted as {B, C} and it is also called as coefficient values.

Besides the equations are {B and C} the best search agent are found with the similarities

Encircle prey

Once the target prey is found it is consider being the best solution and the prey is encircled by the spiral shape with the bubble net

Bubble –net attack model (examination phase)

Two approaches are designed based on the mathematical model of the bubble-net behavior of humpback whales

Through encircling mechanism

This behavior is achieved by decreasing the values such as humpback whales encircling values which is decreased 2 to 0 over the iteration once the spiral is formed the circle shape is updated based the prey position

Search for the prey:

Humpback whales search the prey in random order not according to the position of the prey

Algorithm

```
Step1: Declare the whales population
Step2: Calculate the fitness value such as  search agent=
best search agent
Step3:
while ( t 1< maximum number of iterations )
      for each search agent:
        Update
        if(p1<0.5):
          if(|A1|<1):
            Update current agent by eq. (1)          else:
            Select a random agent
            update current agent by eq (7)         else:
          update search agent by eq (5)
      end-for
      Check if any search agent were goes beyond the
search space
      Calculate fitness of each search agent  Update  if
there is a better solution
       t1 = t1+1
end-while
 Step4: return
```

V.CONCLUSION

In this paper a fact that is analysed in ML algorithm such as support vector machine, random forest and whale optimizer algorithm is made in order to find the high accuracy in feature selection for intrusion detection. In which intrusion detection is the major concern in the data transfer in network. In attack pattern support vector machine does not carry out well when we have huge datasets and also it requires higher time period for large dataset. Random forest techniques have lots of trees rather than having only one tree in decision tree and it combines their output which requires lots of computational power and resources and it also has longer training period for selecting features. In case of whale optimizer algorithm it supports larger dataset than SVM and RF. It also has an ability to avoid local and global optima in the network and get a global optimal solution which makes the feature selection globally.

REFERENCES

[1]. N. Kausar B.B.Samir .I.Ahamad and M.Hussain J.Theor Appl.Inf.Technol 60, 55(2014)

[2]. Noble W.S. Support Vector Machine Applications in computational biology in kernel methods in computational biology (eds.Schoelkopf, B, Tsuda.K & Vert, J.P) 71-92(MIT Press Cambridge, MA 2004)

[3]. Matthias Schonlau Rosie Yuyan ZouThe random forest algorithm for statistical learning, The stats journal (2020)20, Number 1, pp. 3-29

[4]. Guyon I, Weston J, Bamhill, S & Vapnik V.Gene selection for cancer classification using support vector machines. Mach Learn 46,389-422(2002)

[5]. Mirjalili S and Lewis A the Whale Optimizer Algorithm, Adv Eng Softw 2016; 95:51-67

[6]. Abdel-Basset M, Gunasekaran M,El-Shahat D,Mirjalili S (2018) A Hybrid Whale Optimization Algorithm based on Local Search Strategy for the Premutation flow shop scheduling problem 85:129-145

[7]. Aljarah I, Faris H Mirjalili s (2016) Optimizing connection weight in neural networks using the whale optimization algorithm

[8]. Edgar Osuna Robert Freund Federico GirosiAn Improved Training Algorithm for Support Vector Machine (to app ear in the Proc of IEEE NNSP'97, Amelia Island FL,24