Supervised Learning

Homework Assignment #2

Harmeet Singh-16233090

Assignment-: Implement Sequential Forward Floating Selection (SFFS) Scheme as a wrapper to obtain optimum features for the spoof detector.

Objective-: To implement spoof detector which is a two-class classifier that distinguishes live samples from the fake artifacts. Two type of features namely local binary patterns (LBP) and Binary Gabor Patterns (BGP) have been extracted from live and fake fingerprint images from LivDet 2011 fingerprint spoof dataset captured using Digital fingerprint sensor (data matrices extracted from https://umkc.box.com/s/daaj5wcx0y1qdx91cnjcgpx57ord422k). Here we used Support Vector Machine (SVM) classifier is used to train and validate the spoof detector.

Implementation-:

1. Here I have implemented the base SVM classifier for LBP and BGP features using various features of SVM.

2. Then I Implemented SFFS scheme algorithm to obtain the optimum features for the given data set.

3. Again by Using SVM (the classifier itself) as the objective function I had evaluated the subset of features selected by SFFS at each step.

4. Finally Reported the accuracy of SVM at each iteration of SFFS algorithm(78.45%).

- **Well Defined Matlab Code Explanation**-:

To comprehend the time complexity over several loops I used the section wise evaluation for the best accuracy of algorithm designed.

- Section1: This following section loads all the training data set files for the further evaluation.

```
%% Loading All Training Data Set
load('Train_All_Data_DigiBGP');
load('Train_All_Label_DigiBGP.mat');
load('Test_All_Data_DigiBGP.mat');
x=load('Test_All_Label_DigiBGP.mat');
```

- Section2: This following section initializes all the elements of the matrix for the SFFS algorithm evaluation.

```
%% Initialization of Matrix Elements
```

```
no_feat=size(Train_All_Data_DigiBGP,2);
Y=[];
argvalue=0;
```

- Section3: This following section elaborates the implementation of the SFS Algorithm.

```
%% Implementation of SFS Algorithm
perf_val=SFS(no_feat,Y);
[argvalue,argmax]=max(perf_val);
Y=[Y argmax];
```

- Section4: This following section elaborates the execution of SBS Algorithm.

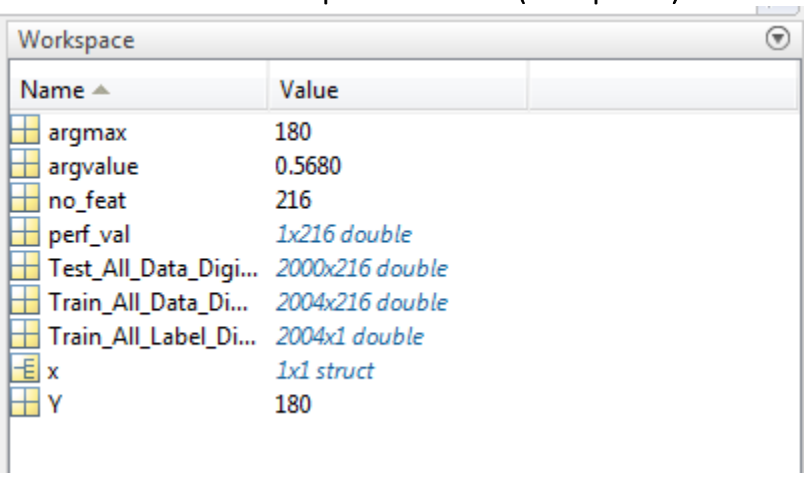
```
%% Implementation of SBS Algorithm (for backtracking)
J=zeros;
for i=1:length(Y)
    feat=Y(Y~=Y(i));
    SVMModel =
fitcsvm(Train_All_Data_DigiLBP(:,feat),Train_All_Label_DigiLBP);
    [label, score] = predict(SVMModel,Test_All_Data_DigiLBP(:,feat));
    test_lbp_label=x.Test_All_Label_DigiLBP;
    perf=classperf(test_lbp_label,label);
    J(i)=perf.CorrectRate ;
end

feat_drop=[]; %index for dropped features
feat_accu=[]; %values of the dropped features
for i=1:length(Y)
    if(J(i)>accuracy) %noting down the features whose values are higher than
the previous best value
        feat_accu=[feat_accu J(i)];
        feat_drop=[feat_drop i];
    end
end
if isempty(feat_drop)==0 %removing the feature which would increase the
performance
    [r,s]=max(feat_accu);
    accuracy=r;
    drop=feat_drop(s);
    Y=Y(Y~=Y(drop));
end
```

SBS and SFS algorithm which have been implemented are attached in the zip file attached which is simple as provided.

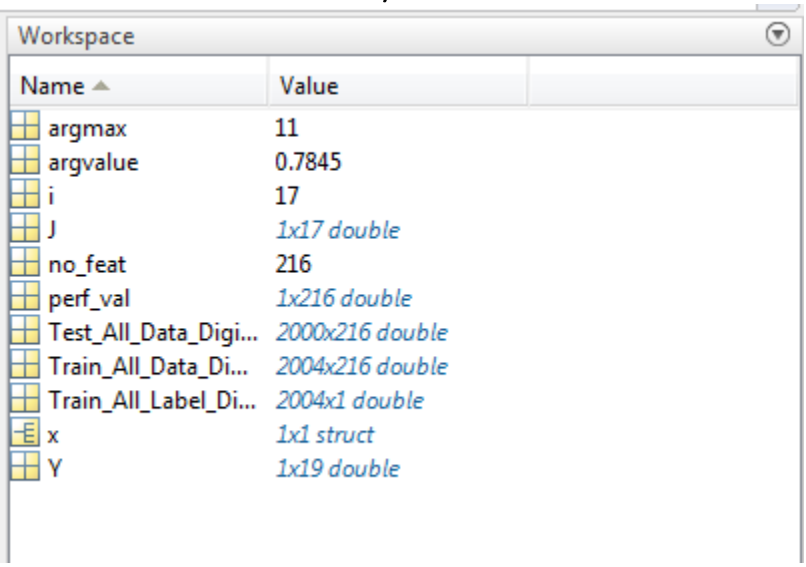
- Output Results with Screenshots- Following are the screen shots after deployment of the above matlab code which clearly depicts the no of features implemented and the no of iterations performed over the data.

➢ For BGP

- Initial Accuracy-56.80%(as depicted)
- No of Features Implemented-216(as depicted)

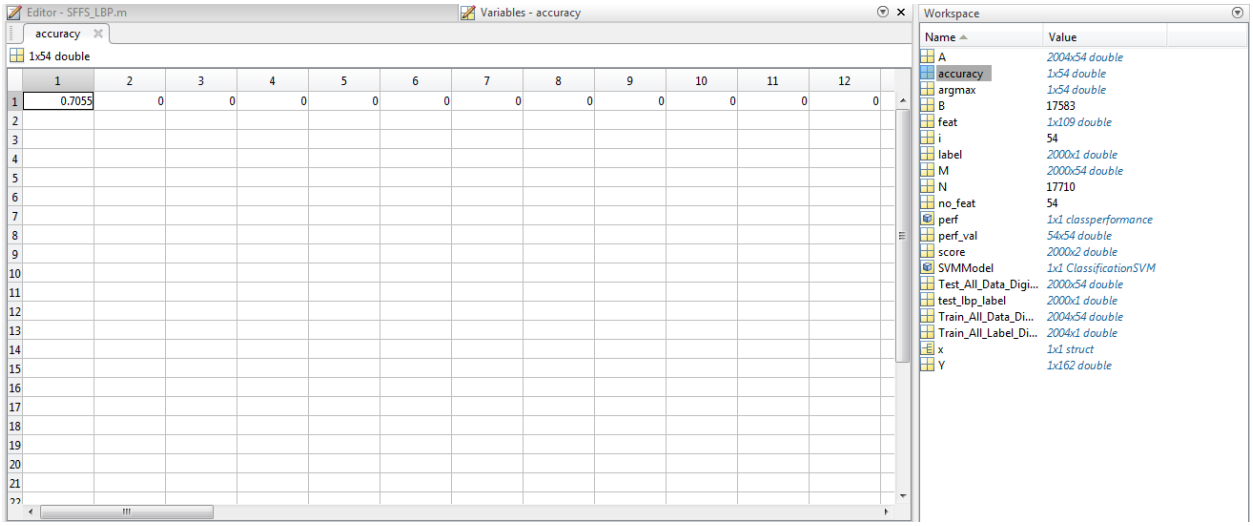


- Final constant Accuracy-78.45%

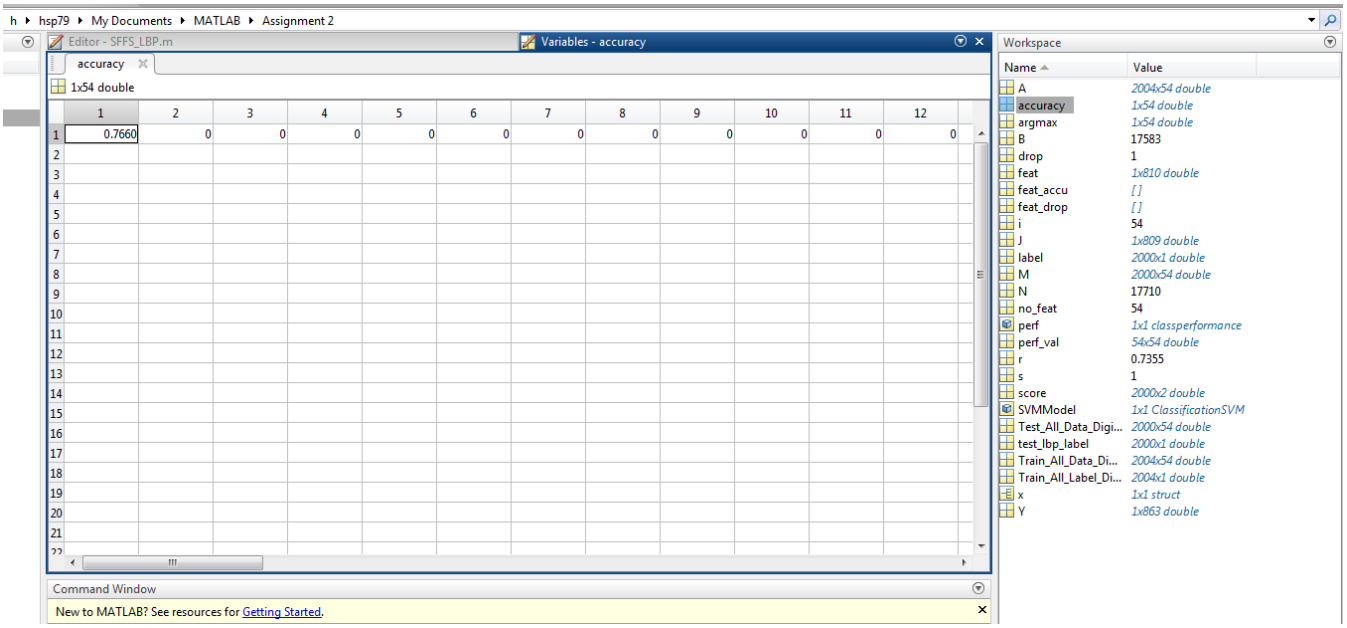


For LBP

- Initial Accuracy-70.55%
- No of Features Implemented-54(as depicted)

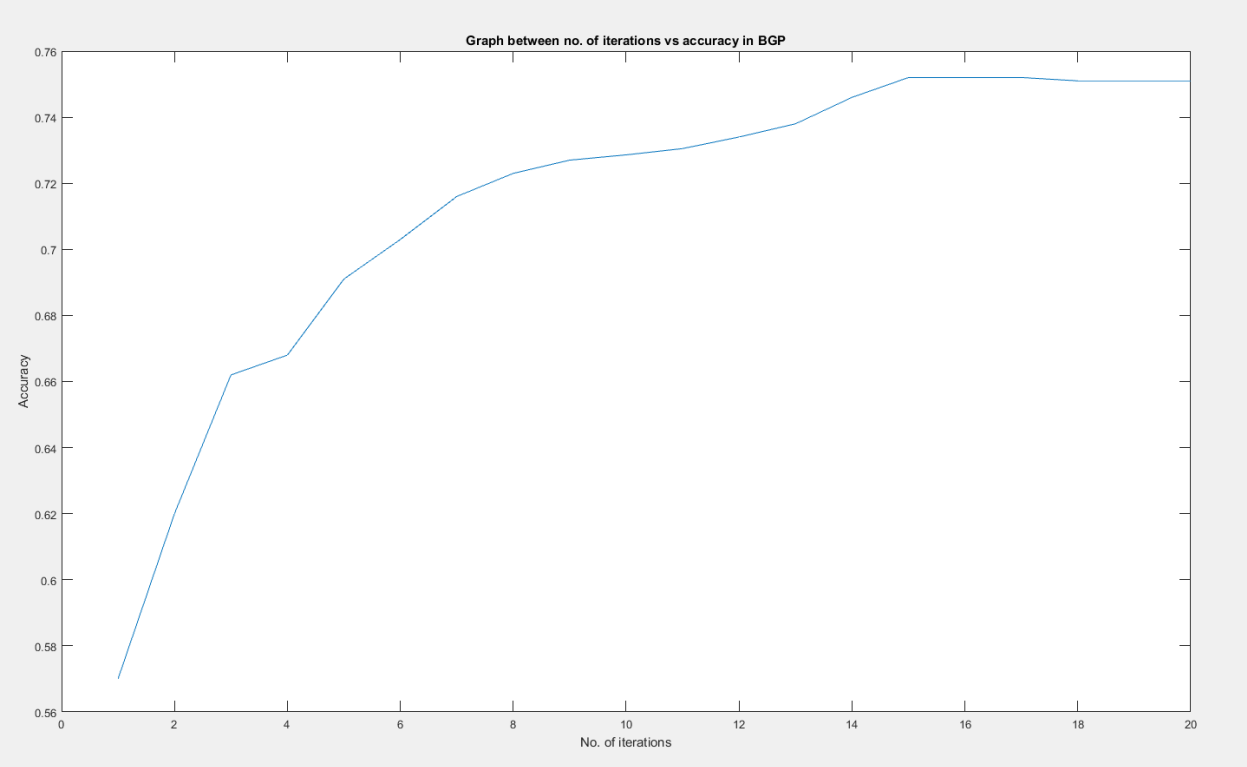


- Final constant Accuracy-76.60%

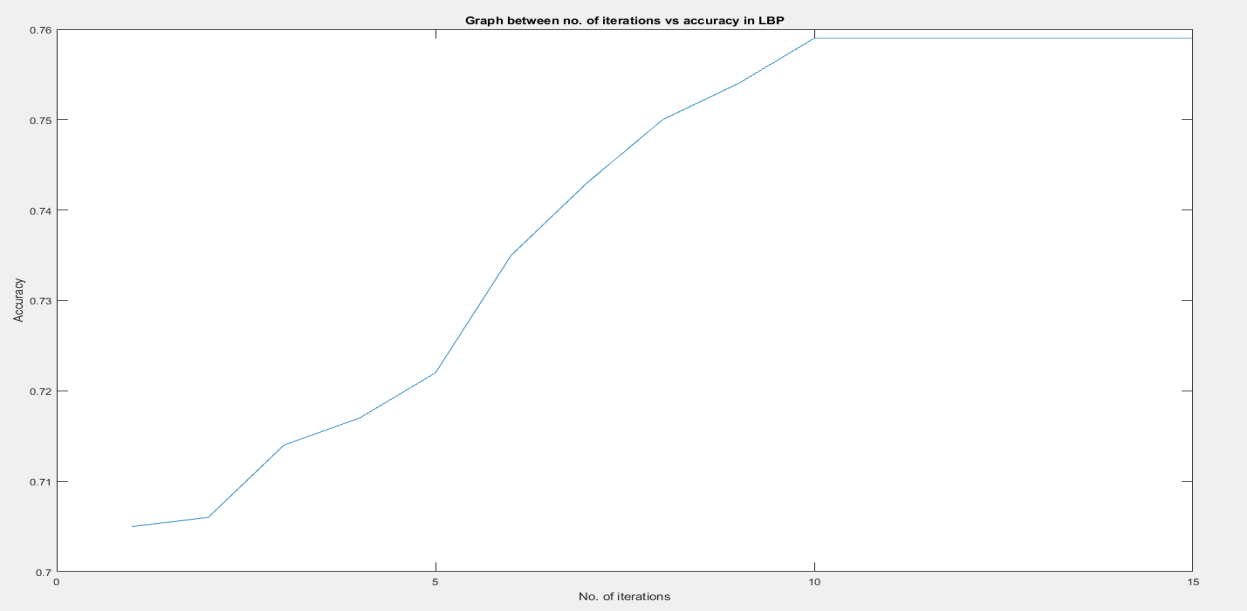


- Output Graphs- Following are the graph results

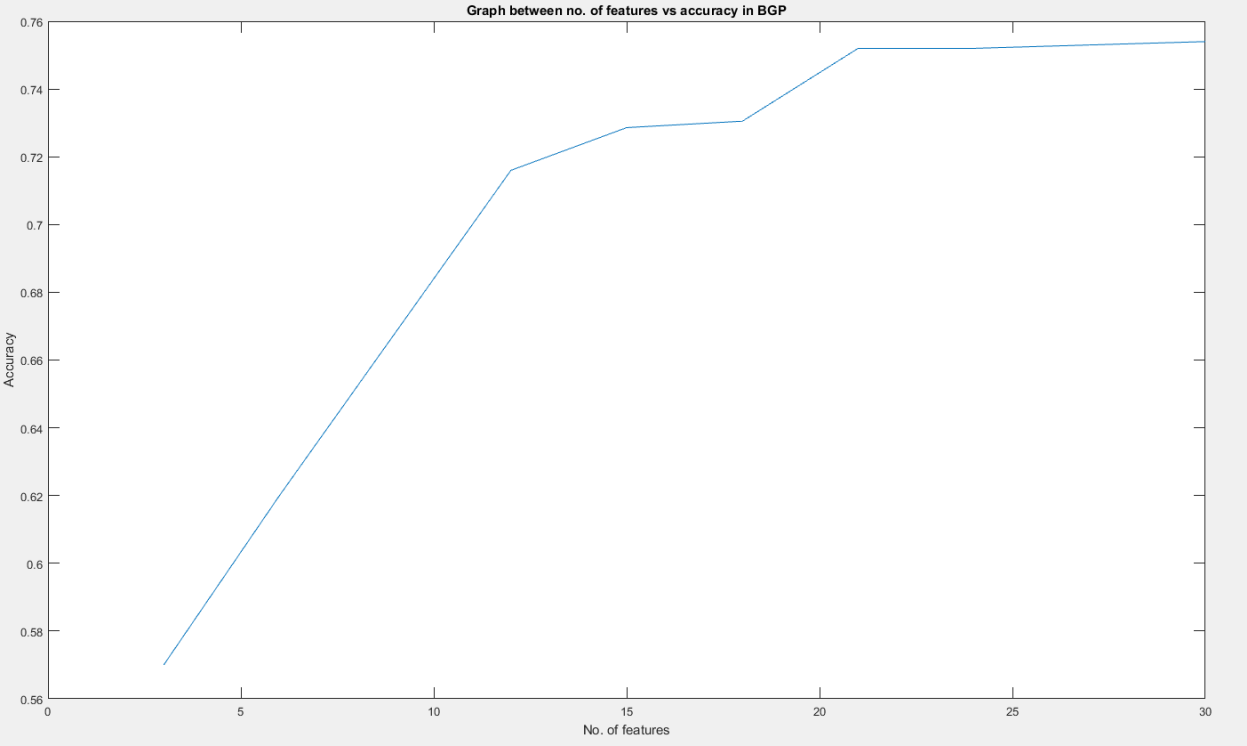
➢ **BGP : *Accuracy vs No. of SFFS iterations***



- **LBP :** ***Accuracy vs No. of SFFS iterations***



- **BGP :** ***No. of features vs Accuracy***



- **LBP :** ***No. of features vs Accuracy***