APPENDIX

Sample Coding

```
clc;
clear all;
close all;
warning off

% Training dataset for ANFIS model
anfisTrainData = [5 10 13.3 31.2 11.2
5 15 19.8 48.2 15.3
5 20 30.1 61.2 22.3
10 10 25.4 46.4 13.2
10 15 32.3 53.2 21.6
10 20 38.2 68.6 26.8
15 10 26.7 48.5 18.2
15 15 34.6 61.7 25.8
15 20 38.4 78.2 30.3];
fprintf('ANFIS Training Dataset\n')
fprintf(' Lf Lw Tensile Flexural Impact\n')
disp(anfisTrainData)

in_fis_ten = genfis1([anfisTrainData(:,1:2) anfisTrainData(:,3)]);
out_fis_ten = anfis([anfisTrainData(:,1:2) anfisTrainData(:,3)],in_fis_ten,30);
figure,plot(anfisTrainData(:,1),anfisTrainData(:,3),anfisTrainData(:,1),evalfis(an
fisTrainData(:,1:2),out_fis_ten));
legend('Training Data','ANFIS Output');
```

```
in_fis_flex = genfis1([anfisTrainData(:,1:2) anfisTrainData(:,4)]);
out_fis_flex = anfis([anfisTrainData(:,1:2) anfisTrainData(:,4)],in_fis_flex,30);
figure,plot(anfisTrainData(:,1),anfisTrainData(:,4),anfisTrainData(:,1),evalfis(an
fisTrainData(:,1:2),out_fis_flex));
legend('Training Data','ANFIS Output');
in_fis_imp = genfis1([anfisTrainData(:,1:2) anfisTrainData(:,5)]);
out_fis_imp = anfis([anfisTrainData(:,1:2) anfisTrainData(:,5)],in_fis_imp,30);
figure,plot(anfisTrainData(:,1),anfisTrainData(:,5),anfisTrainData(:,1),evalfis(an
fisTrainData(:,1:2),out_fis_imp));
legend('Training Data','ANFIS Output');
load tm

anfisTestData=[5 11
 7 20
 12 9
 9 11
 11 12
 15 19];

r1 = evalfis(anfisTestData,out_fis_ten);
r2 = evalfis(anfisTestData,out_fis_flex);
r3 = evalfis(anfisTestData,out_fis_imp);
fprintf('ANFIS Test Dataset\n')
fprintf(' Lf Lw Tensile Flexural Impact\n')
disp(tm)
load rm
fprintf('RM Test Dataset\n')
fprintf(' Lf Lw Tensile Flexural Impact\n')
disp(rm)
fprintf('Applying Genetic Algorithm for Tensile\n');
```

```
%Genetic algorithm
h = waitbar(0,'Applying Genetic Algorithm for Tensile');
X1 = double(tm(:,3));
X1 = X1(:);
numCluster = 5;
X2 = double(rm(:,3));
X2 = X2(:);

lenChrom = numCluster;
numMutations = 5;
numCrossOver = 6;
fprintf('Number of Chromosomes:%f\n',lenChrom)
fprintf('Number of Mutation:%f\n',numMutations)
fprintf('Number of Cross Over:%f\n',numCrossOver)

Y1 = X1;
Y2 = X2;
C1 = zeros(1, length(lenChrom));
C2 = zeros(1, length(lenChrom));
while(length(Y1)> (2 * numCluster))
 tempY = Y1;
 indC1 = ceil((length(Y1)-1e-6)*rand(1,5));
 indC2 = ceil((length(Y2)-1e-6)*rand(1,5));
 if((~isempty(indC1)) && (~isempty(indC2)))
 if(~isempty(C1))
 if(max(indC1)<=length(Y1))
 C1 = Y1(indC1);
 if(length(unique(C1)) ~= numCluster)
 continue;
 else
```

```
Y1(indC1) = [];
end
else
continue;
end
end
if(~isempty(C2))
if(max(indC2)<=length(Y2))
C2 = Y2(indC2);
if(length(unique(C2)) ~= numCluster)
Y2 = tempY;
continue;
else
Y2(indC2) = [];
end
else
continue;
end
end

C1(1:3) = C2(1:3);
C2(1:3) = C1(1:3);
T = C1(4);
C1(4) = C1(1);
C1(1) = T;

T = C2(4);
C2(4) = C2(1);
C2(1) = T;
```

```
normClustersC1 = zeros(1, lenChrom);
DC1 = zeros(numCluster, length(X1));
for c = 1: numCluster
DC1(c,:) = abs(X1 - C1(c));
end

currentClustersC1 = zeros(numCluster, length(X1));
for t = 1:length(X1)
indx = find(DC1(:,t) == min(DC1(:,t)));
currentClustersC1(indx(1), t) = X1(t);
end

for c = 1: numCluster
tempL = currentClustersC1(c,:);
tempL(find(tempL==0)) = [];
normClustersC1(c) = norm(tempL);
end

normClustersC2 = zeros(1, lenChrom);
DC2 = zeros(numCluster, length(X1));
for c = 1: numCluster
DC2(c,:) = abs(X1 - C2(c));
end

currentClustersC2 = zeros(numCluster, length(X1));
for t = 1:length(X1)
indx = find(DC2(:,t) == min(DC2(:,t)));
currentClustersC2(indx(1), t) = X1(t);
end
```

```
for c = 1: numCluster
tempL = currentClustersC2(c,:);
tempL(find(tempL==0)) = [];
normClustersC2(c) = norm(tempL);
end

maxC1 = max(normClustersC1);
maxC2 = max(normClustersC2);
maxC = [maxC1 maxC2];
indmaxC = find(maxC == max(maxC));
if(indmaxC == 1)
C2 = zeros(1,(numCluster));

else
C1 = zeros(1,(numCluster));

end
if(sum(C1)~=0)
initialKmeanCent = C1;
else
initialKmeanCent = C2;
end

end
waitbar(length(Y)/(2 * numCluster),h);
end
close(h);
if(sum(C1)==0)
fprintf('ANFIS Data is best data:\n')
fprintf('Fl\t Fw\t Impact\n')
for m = 1:6
```

```
 fprintf('%f\t%f\t%f\n', tm(m,1),tm(m,2),tm(m,3))
 end

else
 fprintf('RM Data is best data:\n')
 fprintf('Fl\t Fw\t Impact\n')
 for m = 1:6
 fprintf('%f\t%f\t%f\n', rm(m,1),rm(m,2),rm(m,3))
 end
end
fprintf('Applying Genetic Algorithm for Flexural\n');
h = waitbar(0,'Applying Genetic Algorithm for Flexural');
X1 = double(tm(:,4));
X1 = X1(:);
numCluster = 5;
X2 = double(rm(:,4));
X2 = X2(:);

lenChrom = numCluster;

Y1 = X1;
Y2 = X2;
C1 = zeros(1, length(lenChrom));
C2 = zeros(1, length(lenChrom));
while(length(Y1)> (2 * numCluster))
 tempY = Y1;
 indC1 = ceil((length(Y1)-1e-6)*rand(1,5));
 indC2 = ceil((length(Y2)-1e-6)*rand(1,5));
 if((~isempty(indC1)) && (~isempty(indC2)))
 if(~isempty(C1))
```

```
if(max(indC1)<=length(Y1))
C1 = Y1(indC1);
if(length(unique(C1)) ~= numCluster)
continue;
else
Y1(indC1) = [];
end
else
continue;
end
end
if(~isempty(C2))
if(max(indC2)<=length(Y2))
C2 = Y2(indC2);
if(length(unique(C2)) ~= numCluster)
Y2 = tempY;
continue;
else
Y2(indC2) = [];
end
else
continue;
end
end

C1(1:3) = C2(1:3);
C2(1:3) = C1(1:3);

T = C1(4);
C1(4) = C1(1);
```

```
C1(1) = T;

T = C2(4);
C2(4) = C2(1);
C2(1) = T;

normClustersC1 = zeros(1, lenChrom);
DC1 = zeros(numCluster, length(X1));
for c = 1: numCluster
DC1(c,:) = abs(X1 - C1(c));
end

currentClustersC1 = zeros(numCluster, length(X1));
for t = 1:length(X1)
indx = find(DC1(:,t) == min(DC1(:,t)));
currentClustersC1(indx(1), t) = X1(t);
end

for c = 1: numCluster
tempL = currentClustersC1(c,:);
tempL(find(tempL==0)) = [];
normClustersC1(c) = norm(tempL);
end

normClustersC2 = zeros(1, lenChrom);
DC2 = zeros(numCluster, length(X1));
for c = 1: numCluster
DC2(c,:) = abs(X1 - C2(c));
end
```

```
currentClustersC2 = zeros(numCluster, length(X1));
for t = 1:length(X1)
indx = find(DC2(:,t) == min(DC2(:,t)));
currentClustersC2(indx(1), t) = X1(t);
end

for c = 1: numCluster
tempL = currentClustersC2(c,:);
tempL(find(tempL==0)) = [];
normClustersC2(c) = norm(tempL);
end

maxC1 = max(normClustersC1);
maxC2 = max(normClustersC2);
maxC = [maxC1 maxC2];
indmaxC = find(maxC == max(maxC));
if(indmaxC == 1)
C2 = zeros(1,(numCluster));

else
C1 = zeros(1,(numCluster));

end
if(sum(C1)~=0)
initialKmeanCent = C1;
else
initialKmeanCent = C2;
end

end
waitbar(length(Y)/(2 * numCluster),h);
```

```
end
close(h);
if(sum(C1)==0)
 fprintf('ANFIS Data is best data:\n')
 fprintf('Fl\t Fw\t Impact\n')
 for m = 1:6
 fprintf('%f\t%f\t%f\n', tm(m,1),tm(m,2),tm(m,4))
 end

else
 fprintf('RM Data is best data:\n')
 fprintf('Fl\t Fw\t Impact\n')
 for m = 1:6
 fprintf('%f\t%f\t%f\n', rm(m,1),rm(m,2),rm(m,4))
 end
end
fprintf('Applying Genetic Algorithm for Impact\n');
h = waitbar(0,'Applying Genetic Algorithm for Impact');
X1 = double(tm(:,5));
X1 = X1(:);
numCluster = 5;
X2 = double(rm(:,5));
X2 = X2(:);

lenChrom = numCluster;

Y1 = X1;
Y2 = X2;
C1 = zeros(1, length(lenChrom));
C2 = zeros(1, length(lenChrom));
```

```
while(length(Y1)> (2 * numCluster))
 tempY = Y1;
 indC1 = ceil((length(Y1)-1e-6)*rand(1,5));
 indC2 = ceil((length(Y2)-1e-6)*rand(1,5));
 if((~isempty(indC1)) && (~isempty(indC2)))
 if(~isempty(C1))
 if(max(indC1)<=length(Y1))
 C1 = Y1(indC1);
 if(length(unique(C1)) ~= numCluster)
 continue;
 else
 Y1(indC1) = [];
 end
 else
 continue;
 end
 end
 if(~isempty(C2))
 if(max(indC2)<=length(Y2))
 C2 = Y2(indC2);
 if(length(unique(C2)) ~= numCluster)
 Y2 = tempY;
 continue;
 else
 Y2(indC2) = [];
 end
 else
 continue;
 end
```

```
end

C1(1:3) = C2(1:3);
C2(1:3) = C1(1:3);

T = C1(4);
C1(4) = C1(1);
C1(1) = T;

T = C2(4);
C2(4) = C2(1);
C2(1) = T;

normClustersC1 = zeros(1, lenChrom);
DC1 = zeros(numCluster, length(X1));
for c = 1: numCluster
DC1(c,:) = abs(X1 - C1(c));
end

currentClustersC1 = zeros(numCluster, length(X1));
for t = 1:length(X1)
indx = find(DC1(:,t) == min(DC1(:,t)));
currentClustersC1(indx(1), t) = X1(t);
end

for c = 1: numCluster
tempL = currentClustersC1(c,:);
tempL(find(tempL==0)) = [];
normClustersC1(c) = norm(tempL);
end
```

```
normClustersC2 = zeros(1, lenChrom);
DC2 = zeros(numCluster, length(X1));
for c = 1: numCluster
DC2(c,:) = abs(X1 - C2(c));
end

currentClustersC2 = zeros(numCluster, length(X1));
for t = 1:length(X1)
indx = find(DC2(:,t) == min(DC2(:,t)));
currentClustersC2(indx(1), t) = X1(t);
end

for c = 1: numCluster
tempL = currentClustersC2(c,:);
tempL(find(tempL==0)) = [];
normClustersC2(c) = norm(tempL);
end

maxC1 = max(normClustersC1);
maxC2 = max(normClustersC2);
maxC = [maxC1 maxC2];
indmaxC = find(maxC == max(maxC));
if(indmaxC == 1)
C2 = zeros(1,(numCluster));

else
C1 = zeros(1,(numCluster));

end
if(sum(C1)~=0)
initialKmeanCent = C1;
```

```
 else
 initialKmeanCent = C2;
 end

 end
 waitbar(length(Y)/(2 * numCluster),h);
end
close(h);
if(sum(C1)==0)
 fprintf('ANFIS Data is best data:\n')
 fprintf('Fl\t Fw\t Impact\n')
 for m = 1:6
 fprintf('%f\t%f\t%f\n', tm(m,1),tm(m,2),tm(m,5))
 end

else
 fprintf('RM Data is best data:\n')
 fprintf('Fl\t Fw\t Impact\n')
 for m = 1:6
 fprintf('%f\t%f\t%f\n', rm(m,1),rm(m,2),rm(m,5))
 end
end
```

Sample screen shots of simulation in MATLAB

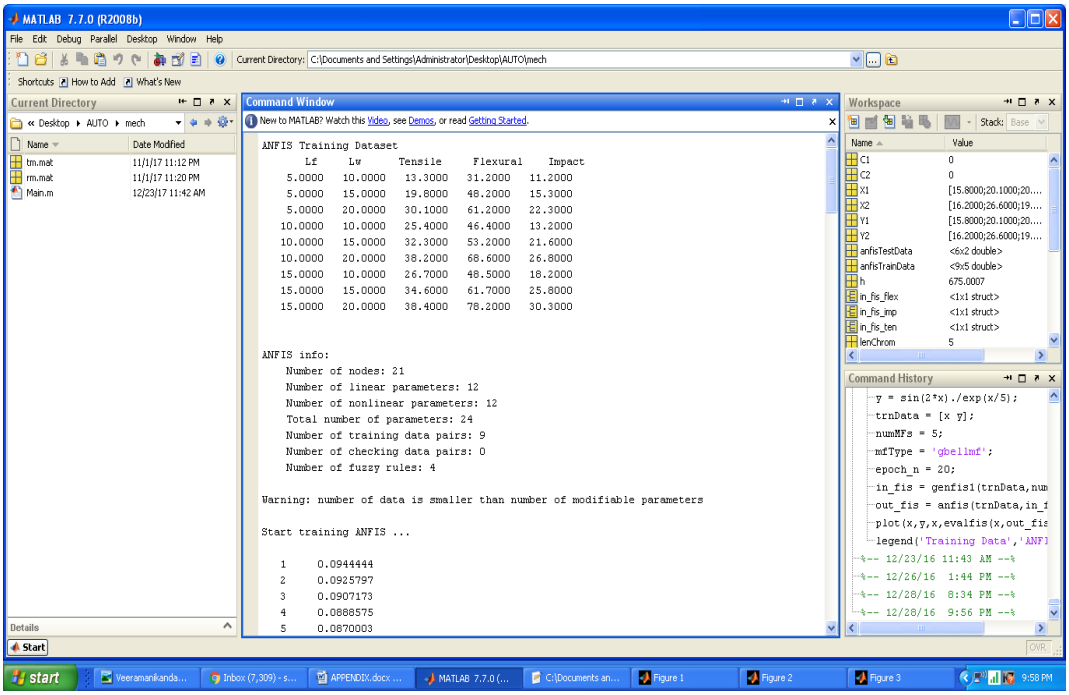

Figure shows the simulation screenshot of the proposed system which depicts the training samples inputs as fiber length, fiber content, tensile, flexural and impact strength values.

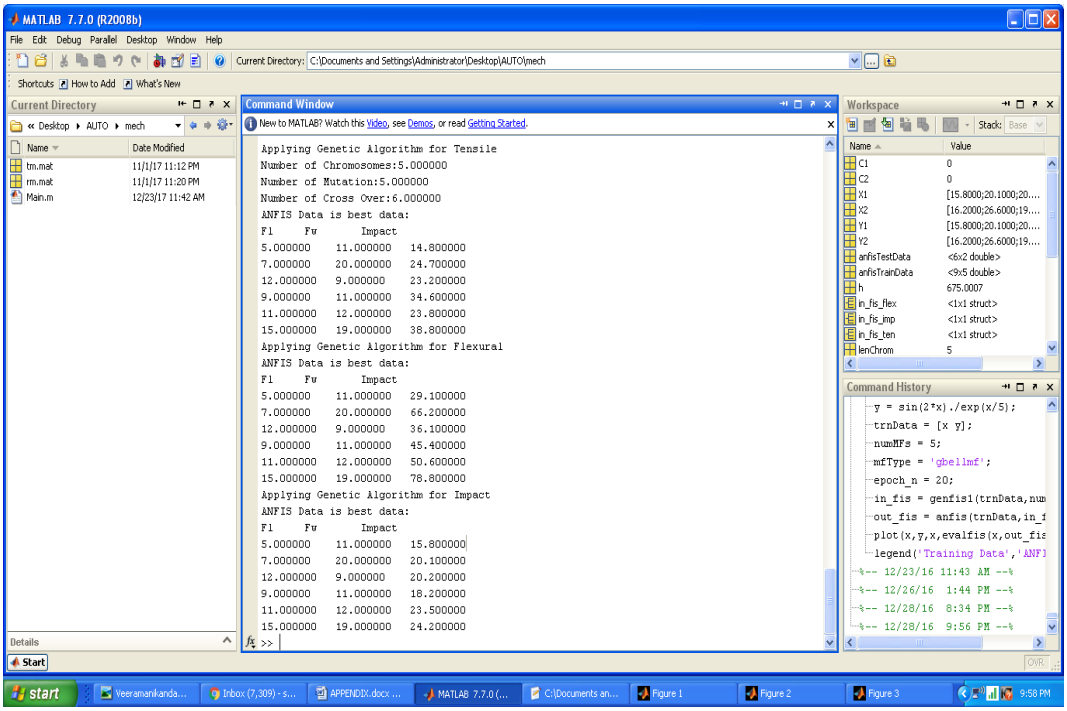

Figure shows the simulation screenshot of the proposed system which produces the testing values of the tensile, flexural and impact strength.

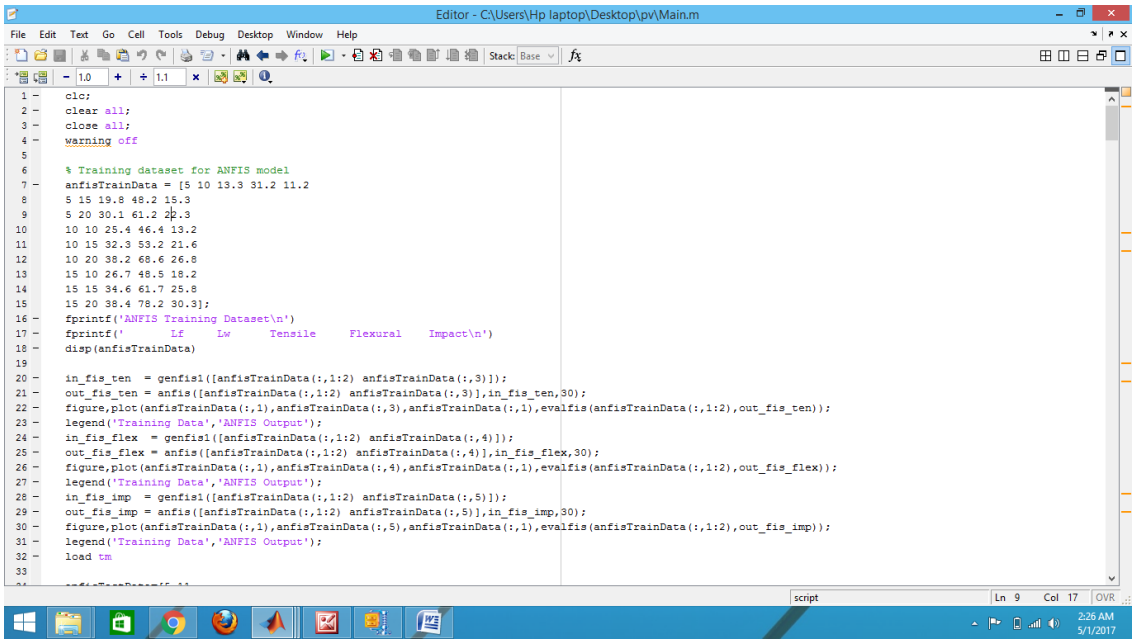

Figure shows the simulation screenshot of the proposed methodology with respect to training modes of tensile, flexural and impact strength. The training pattern of this Neuro Fuzzy model consists of fiber length, fiber content as training pattern inputs and tensile, flexural and impact strength as trained inputs of the corresponding training pattern inputs. Each input pattern is individually trained in Neuro Fuzzy classification approach, which produces corresponding trained pattern.