In [1]:
```
# Import Module
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [3]:
```
data = pd.read_excel("E:\\data\\test\\test_2\\BIOMASS&LAI DATA\\R7_LAI_081622.xlsx")
data.columns
```

Out[3]:
```
Index(['Location', 'Plot #', 'Treatment', 'Cover Crop ', 'Date and Time',
       'Average Above PAR', 'Average Below PAR', 'Tau [T]',
       'Leaf Area Index [LAI]', 'Leaf Distribution [X]', 'Beam Fraction [Fb]',
       'Zenith Angle', 'Latitude', 'Longitude', 'Date', 'CC', 'Plot',
       'covercrop', 'Fert', 'NDVI', 'GNDVI', 'EVI2', 'NDRE', 'ARVI', 'CCCI',
       'GRRI', 'CARI', 'NAVI', 'SCCCI', 'CIRE', 'CVI', 'GCVI'],
      dtype='object')
```

In [4]:
```
# LR MODELLING
# Import Module
from sklearn.linear_model import LinearRegression
from sklearn import metrics
from sklearn.metrics import r2_score
import statsmodels.api as sm

from sklearn.model_selection import train_test_split
#column name as indexes
index=['CC','NDVI', 'GNDVI', 'EVI2', 'NDRE', 'ARVI', 'CCCI', 'GRRI', 'CARI', 'NAVI','SCCCI', 'CIRE', 'CVI', 'GCVI']
for i in index:
    #print(i)

    data=data
    x=np.array(data[[i]].values)
    y=np.array(data['Leaf Area Index [LAI]'].values).reshape(-1,1)

    model=LinearRegression()
    model.fit(x,y)
    y_pred=model.predict(x)

    metrics_dict = metrics.r2_score(y, y_pred) #(actual, predicted)
    #print(metrics_dict)
```

```
    r2=np.sqrt(metrics.mean_squared_error(y, y_pred))
    r1= metrics_dict
    res=[]
    res.append(i)
    res.append(r1)
    res.append(r2)

    print(res)
```

```
['CC', 0.5987420147867349, 0.43156493894722164]
['NDVI', 0.6335392029508576, 0.41242793290423524]
['GNDVI', 0.5708866060085767, 0.446293278179984]
['EVI2', 0.7229965592405869, 0.35857242834856934]
['NDRE', 0.3582100643084315, 0.5457965193302683]
['ARVI', 0.6694277971496696, 0.39171254136758377]
['CCCI', 0.3582100643084316, 0.5457965193302682]
['GRRI', 0.730694039658632, 0.35355525222698725]
['CARI', 0.7274268072921972, 0.35569346090273973]
['NAVI', 0.6171643798055626, 0.421541639308322]
['SCCCI', 0.054830420855623285, 0.6623524973686764]
['CIRE', 0.6099781207474959, 0.4254796524650508]
['CVI', 0.4424130997666472, 0.5087337984748882]
['GCVI', 0.5844295007704472, 0.43919427016898055]
```

In [5]:

```
# RF and SVR model

# Import Module

from sklearn.ensemble import RandomForestRegressor
from sklearn import metrics
from sklearn.metrics import r2_score
import statsmodels.api as sm
from sklearn import svm
from scipy import stats
from sklearn.model_selection import train_test_split

# Folder Path
#path="D:\\data\\excel_csv\\early_planting\\datewise_data\\06_08_2022"
#path="D:\\data\\excel_csv\\early_planting\\datewise_data\\07_08_2022\\EP\\New folder"
# Change the directory
#os.chdir(path)

# Read text File
```

```
# def read_text_file(file_path):
#     with open(file_path, 'r') as f:
#         print(f.read())
index=['CC','NDVI', 'GNDVI', 'EVI2', 'NDRE', 'ARVI', 'CCCI', 'GRRI', 'CARI', 'NAVI','SCCCI', 'CIRE', 'CVI', 'GCVI']
for i in index:

    data=data

    x=np.array(data[[i]].values)
    y=np.array(data['Leaf Area Index [LAI]'].values).reshape(-1,1)
    x=np.array(data[[i]].values)
    y=np.array(data['Leaf Area Index [LAI]'].values).reshape(-1)

    #RF MODEL
    regressor=RandomForestRegressor()
    regressor.fit(x, y)
    y_pred=regressor.predict(x)
    coeff=pd.Series(regressor.feature_importances_)
    #print("RF model result:")
    #print(coeff)
    #print ('Root Mean Squared Error in RF:', np.sqrt(metrics.mean_squared_error(y_test,y_pred)))
    #print("rsquare IN RF:", metrics.r2_score(y_test, y_pred))
    r2=np.sqrt(metrics.mean_squared_error(y, y_pred))
    r1= metrics.r2_score(y, y_pred)
    #SVM model
    regressor1=svm.SVR(kernel='rbf', C=100, gamma=0.1, epsilon=.1)

    regressor1.fit(x, y)
    #print("intercept:",regressor.intercept_)

    y_pred_svm=regressor1.predict(x)
    #print('R2 score in svm :', metrics.r2_score(y_test,y_pred_svm))
    #print ('Root Mean Squared Error in svm:', np.sqrt(metrics.mean_squared_error(y_test,y_pred_svm)))

    r3=metrics.r2_score(y,y_pred_svm)
    r4=np.sqrt(metrics.mean_squared_error(y,y_pred_svm))
    res=[]
    m1='rf'
    m2='svr'
    res.append(i)
    res.append(m1)
    res.append(r1)
    res.append(r2)
    res.append(m2)
    res.append(r3)
```

```
    res.append(r4)
    print(res)
```

```
['CC', 'rf', 0.9355321502148893, 0.17298398721654307, 'svr', 0.5886274204833326, 0.43697036107042747]
['NDVI', 'rf', 0.9246644118614263, 0.18699689792435936, 'svr', 0.6014783049832433, 0.4300909408609134]
['GNDVI', 'rf', 0.9143554114218639, 0.19938123156405638, 'svr', 0.4392660739689994, 0.5101674276789266]
['EVI2', 'rf', 0.9266004025782485, 0.1845785158137314, 'svr', 0.12242890303295095, 0.6382274309071552]
['NDRE', 'rf', 0.8517174257374196, 0.26234897337960594, 'svr', 0.21047626284276355, 0.6053644401775059]
['ARVI', 'rf', 0.9523391718450167, 0.14873565925269328, 'svr', 0.7433491197820361, 0.3451482610178693]
['CCCI', 'rf', 0.8604200319632889, 0.2545340324331238, 'svr', 0.32809642333656097, 0.5584544899843588]
['GRRI', 'rf', 0.9486390984449541, 0.15440118522861132, 'svr', 0.7259012084056315, 0.3566874844894033]
['CARI', 'rf', 0.9546153516545237, 0.14514056118122154, 'svr', 0.7328413292739413, 0.3521429089756089]
['NAVI', 'rf', 0.9202905836965686, 0.19234864222724987, 'svr', 0.5455446045914591, 0.45928254720211464]
['SCCCI', 'rf', 0.8440185208735358, 0.2690734379309855, 'svr', 0.04573534101366916, 0.6655316760849198]
['CIRE', 'rf', 0.9024994081615008, 0.2127345047236105, 'svr', 0.00634363886229683, 0.6791292238156681]
['CVI', 'rf', 0.8829580136680315, 0.23308007672614695, 'svr', 0.45075812350493294, 0.5049125106185308]
['GCVI', 'rf', 0.9113914271240314, 0.20280197566427507, 'svr', 0.6195902954099479, 0.4202039246904852]
```

In [ ]: