CSCI 8994
Directed Research in Computer Science
Final Report

Harsh Jha (4907153, jhaxx036@umn.edu)

Topic: Land cover classification based on time series clustering

Aim: Aim is to find an effective clustering/classification technique to separate bimodal land cover type from other type of areas in the Zimbabwe region. By bimodal land cover, we mean the land cover where farmers have grown two crops in a year. Other land covers includes human settlement, unimodal land covers, barren lands, etc.

Data:

Region considered: Though the region of focus is Zimbabwe (16° S to 23° S and 26° E to 33° E), but as we have data for 10.004178° S to 19.995856° S and 20.313047° E to 31.920095° E, we have focused on 16° S to 19.995856° S and 26° E to 31.920095° E, i.e. subset of available data which represents Zimbabwe.

Test Set:

Data is taken from NDVI time series from data_unfiltered_h20v10_EVI_1000m_16day.mat. It represents data for tile h20v10 and has the following matrices -

- data (1440000 x 285) - time series for 1440000 points with 285 timestamps
- dates (285 x 11) - dates for each of the 285 timestamps
- lat (1440000 x 1) - latitude values of 1440000 points varying from -19.9959 to -10.0042
- lon (1440000 x 1) - longitude values of 1440000 points varying from 20.3130 to 31.9201

Modified Test Data:

As we are using only a subset of the original test data available and also we are slicing the data year by year, following is the preprocessing done on the data -

A. data matrix of 285 timestamps is sliced to keep only 16 to 269 timestamps. We want to start the year when spring season starts in the region and move 23 timestamps from there to finish that year. As October is approximately the start of the spring season in that region, I sliced the data from 16th timestamp ('15-Oct-2000') (as this the earliest October timestamp in the dataset), till 269th timestamp ('30-Sep-2011') (as this the latest September timestamp in the dataset; year would end at September if it starts from October.)
B. Now, we have 253 timestamps and each year is represented by 23 timestamps which implies that in test set we have data for 11 years ranging from October, 2000 to September, 2011.
C. Slicing the data year by year yields the following matrices (just to confirm the integrity of data 1440000x11 = 15840000) -
    a. data - 15840000 x 23
    b. lat - 15840000 x 1

c. lon - 15840000 x 1

Train Set:
Training data is available in the following format -
<id><rank><lat><lon><label series(one value for each year><evi time series>
where -
label series - one label for each year(14 labels from 2000 to 2013)
evi time series - starts from Feb 18, 2000 and goes till 16 Dec, 2013 (319 time steps)

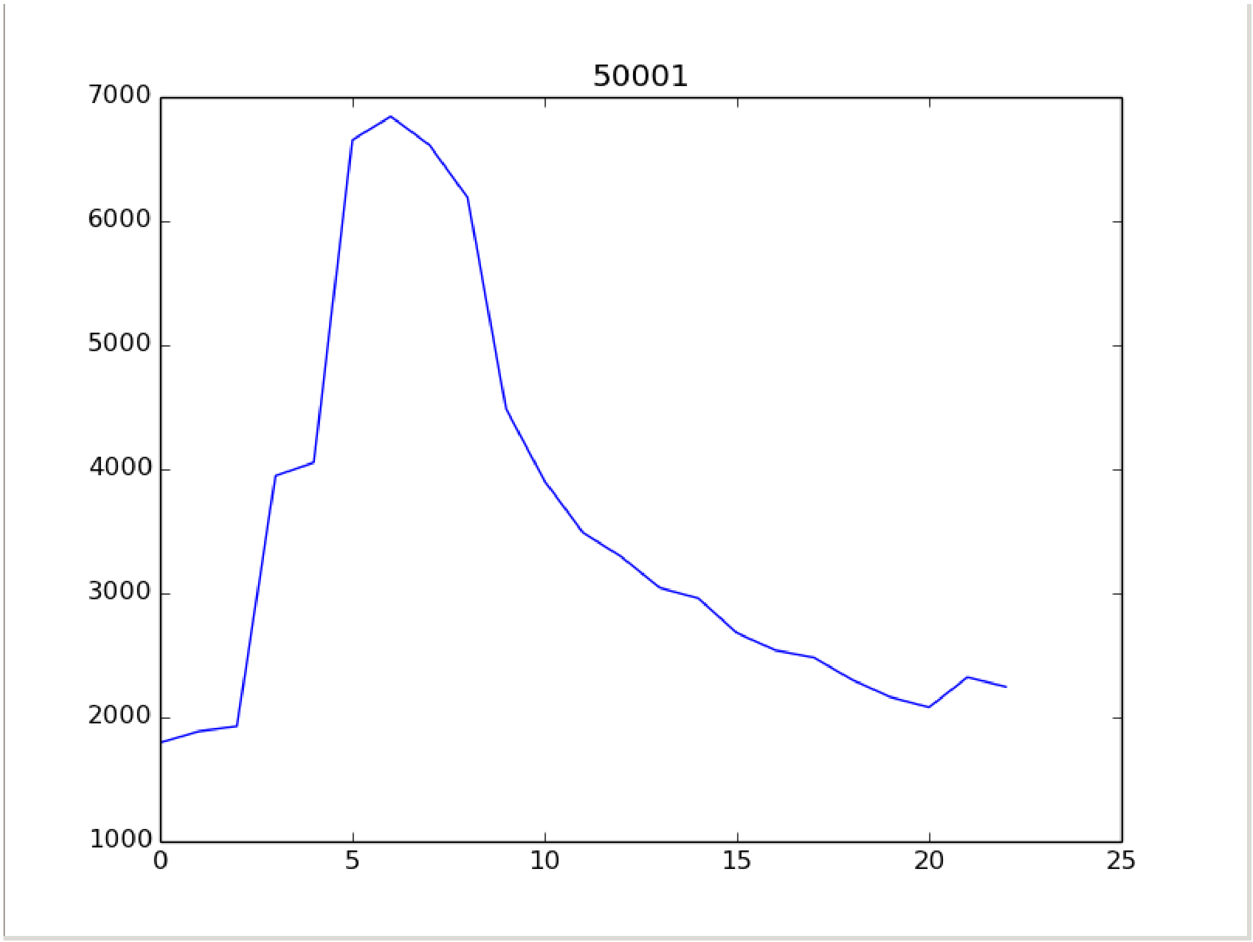
Labels are marked as 0 (unknown), 1 (single crop) and 2 (double crop) i.e. 2 is our desired class.

Modified Train Set:
As year 2000 is partially present (data for the month of Jan is missing), I removed it from the training set to keep the whole years. So first 20 timestamps are removed and training set data now ranges from 2011 to 2013.

Classification methods attempted:
I) As our main intention is to separate the bimodal time series from other regions, we tried to concentrate on the 3rd coefficient of fourier transform of time series. 3rd coefficient represents the energy corresponding to 2nd harmonic of fourier transform. As 2nd harmonic corresponds to two peaks per cycle, more energy corresponding to 2nd harmonic as compared to other harmonics can mean a bimodal time series. Linear relationship between the 2nd harmonic amplitude and bimodal behavior has been shown before in [6].
We conducted experiment where we took fourier transform of all the time series and sorted them on the basis of 2nd harmonic in the descending order. As per our assumption we should see more bimodal time series in the upper part of this sorted matrix of time series and less at the bottom where energy corresponding to 2nd harmonic is less.
This implies that we could find a threshold to separate the bimodal time series from others. But we didn't see the expected results. Plots for these sorted time series can be seen in the directory "plots_2ndharmonic". We can see that in many cases even if the time series has high energy corresponding to 2nd harmonic doesn't show bimodal behavior. From inspection of few such time series we see that these are the cases where the 1st harmonic also has significantly high energy as compared to 2nd harmonic. In these cases instead of boosting one of the peaks of 2nd harmonic, 1st harmonic mitigates the bimodal behavior.

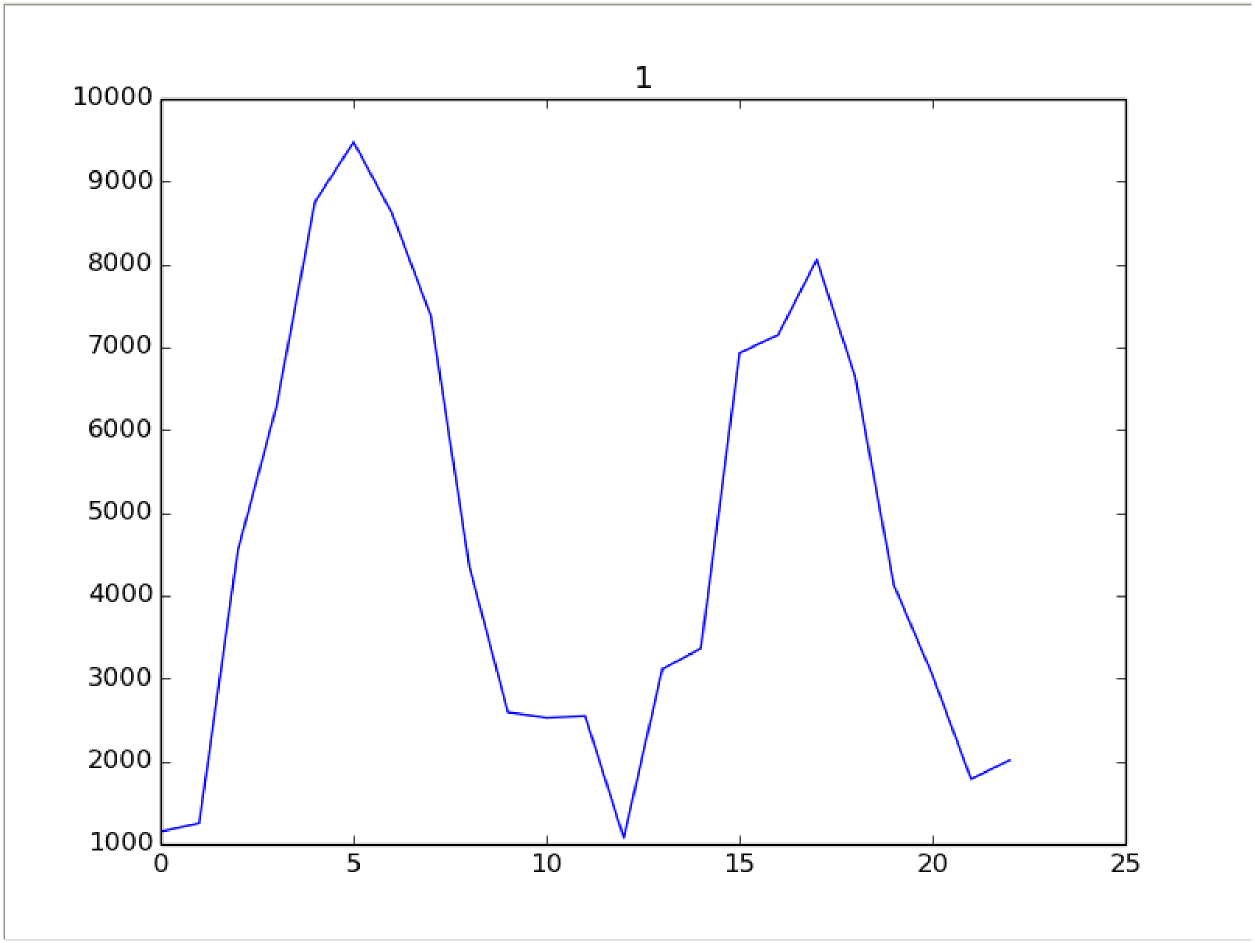
Eg. -
DFT coefficients of time series 1 - 1794, 1886, 1926, 3946, 4054, 6654, 6843, 6615, 6189, 4485, 3905, 3488, 3293, 3043, 2959, 2680, 2540, 2482, 2302, 2162, 2080, 2322, 2246

Plot for time series 1 -



DFT coefficients of time series 2 - 1055, 2008, 5186, 8112, 8557, 9638, 9633, 7549, 3010, 935, 1066, 1914, 4017, 6013, 6734, 6860, 6443, 5433, 4857, 2822, 1979, 745, 936

Plot for time series 2 -



We can see that in first case difference in 3rd coefficient and the 1st coefficient is 40 and in 2nd case it is 3178. Similar behavior was seen for several other time series. This implies that larger the 3rd coefficient as compared to 2nd coefficient, higher are the chances of finding bimodal behavior in the time series. This tempted us to move to the next method.

II)
As seen in the previous method, 3rd coefficient should have a higher value as compared to the second coefficient to find the bimodal behavior. Hence, we tried the ratio of 3rd coefficient and 2nd coefficient as a measure to separate bimodal time series. So, we sorted the time series based on this ratio and tried to follow the same threshold based method as we tried before. Resultant plots are present in directory “plots_2ndby1stharmonic”. We could see that although above issues were resolved but we could still see misclassified time series. This indicates that that more than two coefficients have a role to play to classify it properly with a different combination. So we moved on to different classification methods - k-NN and SVM with different feature selection methods. Below are those methods discusses -

Feature selection:
Using time series directly for feature selection is not a practical idea because of two reasons -

A. High dimensionality
B. High correlation among consecutive values
C. It is hard to separate noise from the original signal in time domain

Following feature selection measures were tried on the train data. For each of the experiments below I have used 5 fold cross validation to calculate the value of the parameter -

1) Using Discrete Fourier Transform -
Using DFT of the time series we get sinusoids (harmonics) of different frequencies (which can be considered as independent features). As shown in [1] and [2] that most of the phenology related information is stored in the first few harmonics. Hence, I tried to consider only 1 to 7 harmonics as features to see which DFT coefficients are the most discriminative in this case. This not only helps us in better selection of features but also helps in reducing noise which mostly comes from the high frequency harmonics of the time series [1].
I have used KNN (k=1) to classify the time series and efficiency of classification is checked using F-measure with 5 fold cross validation. I tried values of k from 1 to 6 and distance measure as "minkowski" with values of q from 1 to 4, but I couldn't see much difference in the values of F-measure. Although k=4 produced best value of F-measure but the improvement was of approximately 0.0004 (from k=1). So to keep things simple and calculations fast, I went ahead with k=1.

kNN (k=1)				
No. of coefficients	F-measure	Precision	Recall	Accuracy
1	0.690928356118	0.691531360885	0.690479189787	0.778347315221
2	0.967844680665	0.968020317835	0.967674301278	0.899643397727
3	**0.999468661624**	0.999498099792	0.999439341406	0.970471279407
4	0.999332763558	0.999347839951	0.999317693727	0.969193668386
5	0.999238968887	0.999134875973	0.999343099288	0.967869878653
6	0.999075422676	0.999017442541	0.999133487506	0.965786705662
only 2nd	0.831502127443	0.83025781113	0.83276605701	0.826101233998

SVM (linear)				
No. of coefficients	F-measure	Precision	Recall	Accuracy

2	0.937724617089	0.939354150541	0.940976329621	0.901069806819
3	**0.999510160321**	0.999855896341	0.999164957588	0.963031375869
4	0.995727762097	0.999741118606	0.991855142147	0.946155622258
5	0.99875989544	1.0	0.997530437921	0.959444829267
6	0.998270995214	0.998718547679	0.997825621583	0.934964981144

We see that keeping first two coefficients give us the best results. As we expect that most of the desired information for this case is stored in the second coefficient. So I tried keeping only the second coefficient (which is similar to threshold method) to see how discriminative that is. But we can see F-measure decreases by huge amount. This is in sync with the previous experiments that I performed which is -

First harmonic can be in phase with second harmonic or out of phase with it. If it is in phase then it would boost a peak of second harmonic component and bimodal behavior (if exists) is not disrupted. In another case, if it is out of phase with the second harmonic then it would try to cancel bimodal behavior and work in the opposite direction. Hence, when we try to use threshold method to classify time series based only on the second coefficients, we don’t see desirable results. We need both the coefficients.

2) Using Discrete Cosine Transform -
DCT also provides a DFT like transformation from time to frequency domain but the basis functions here are only cosines, hence we don’t get the complex numbers as in case of DFT. Following are the advantages of using DCT -
a) DCT provides better energy compression as compared to DFT [3]
b) DCT coefficients don’t have complex values, hence easier to deal with

kNN (k=1)				
No. of coefficients	F-measure	Precision	Recall	Accuracy
1	0.694366572562	0.692883500379	0.695890744153	0.779794248185
2	0.773051772999	0.772857757795	0.773252053398	0.807429641602
3	0.926187499935	0.925756370756	0.926643138739	0.872038790118
4	0.995530205569	0.995198481163	0.995862620629	0.944046794428
5	0.998136481141	0.997999643558	0.998273640531	0.953174786423

6	**0.998991297783**	0.999066179567	0.998916518424	0.964493701737
7	0.998532057873	0.998577369263	0.998486921705	0.961835860335

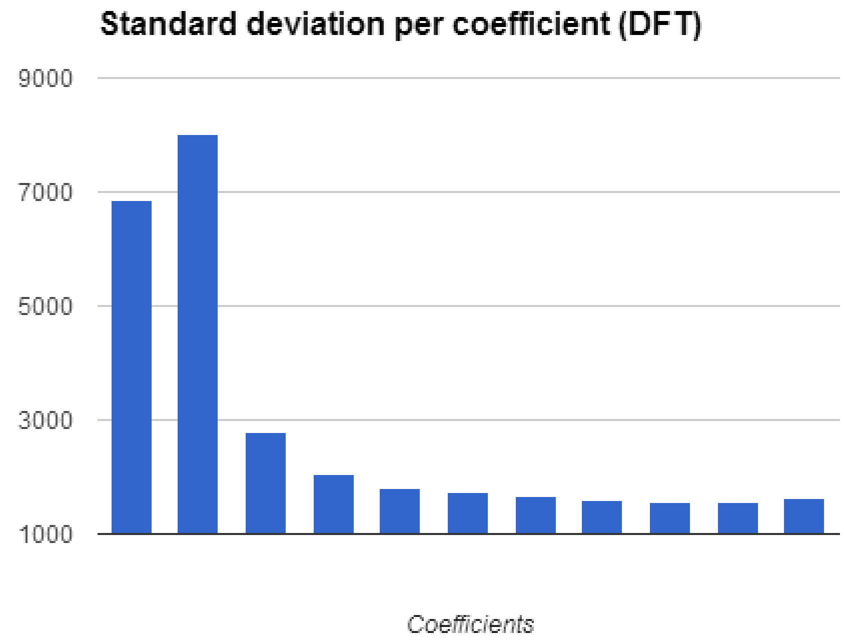
SVM (linear)				
No. of coefficients	F-measure	Precision	Recall	Accuracy
2	0.292449816225	0.552254886675	0.387325979945	0.65005772339
3	0.718241793257	0.793495755746	0.745639163992	0.858611046974
4	0.983022985993	0.973790197163	0.992747629361	0.89802714282
5	0.979655638797	0.989003360917	0.970952072095	0.895913184022
6	**0.987742755539**	0.987289891416	0.988278980228	0.918694681752
7	0.961420844344	0.966699121347	0.960280360261	0.904815413428

3) <u>Using Discrete Wavelet Transform</u> -
DFT and DCT assume that the input time series is periodic which may not be case always. With DWT we have the following advantages -
a) DWT doesn’t make the assumption of input being periodic
b) DWT has complexity of O(N) as compared to O(N log N) of DFT [5]
c) As we have also have space factor in wavelet transform, hence fewer coefficients are needed to represent the original time series [5]

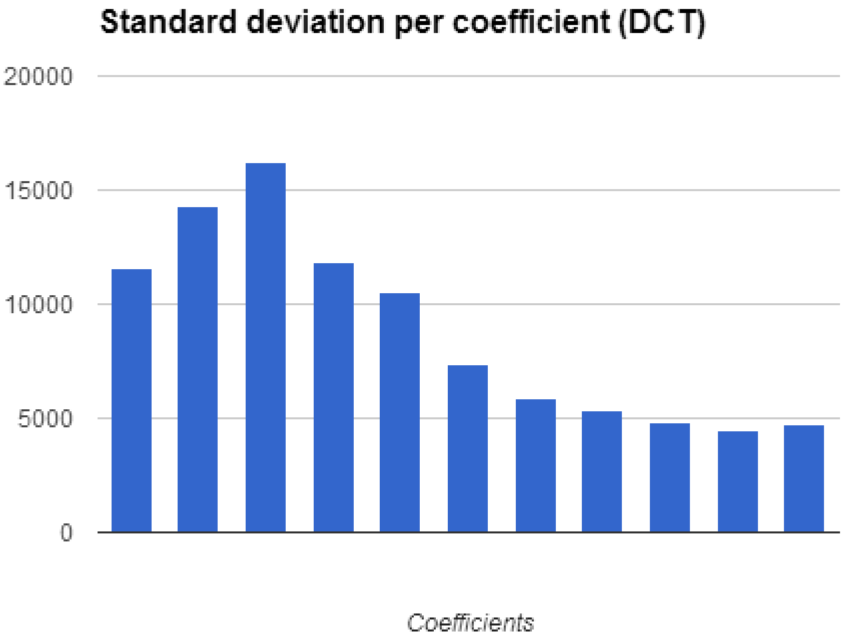
Keeping a subset of coefficients or putting a threshold on the coefficients to reduce the number of features, none of them yield results close to the results seen in case of DFT and DCT as coefficients. Hence, I won’t be using DWT as feature extraction method.

To support the claim that first few coefficients are most discriminative, we see the standard deviation of first few coefficients -

1) DFT - We can see that first two coefficients have wide range of values, hence high standard deviation, and therefore more discriminative power.



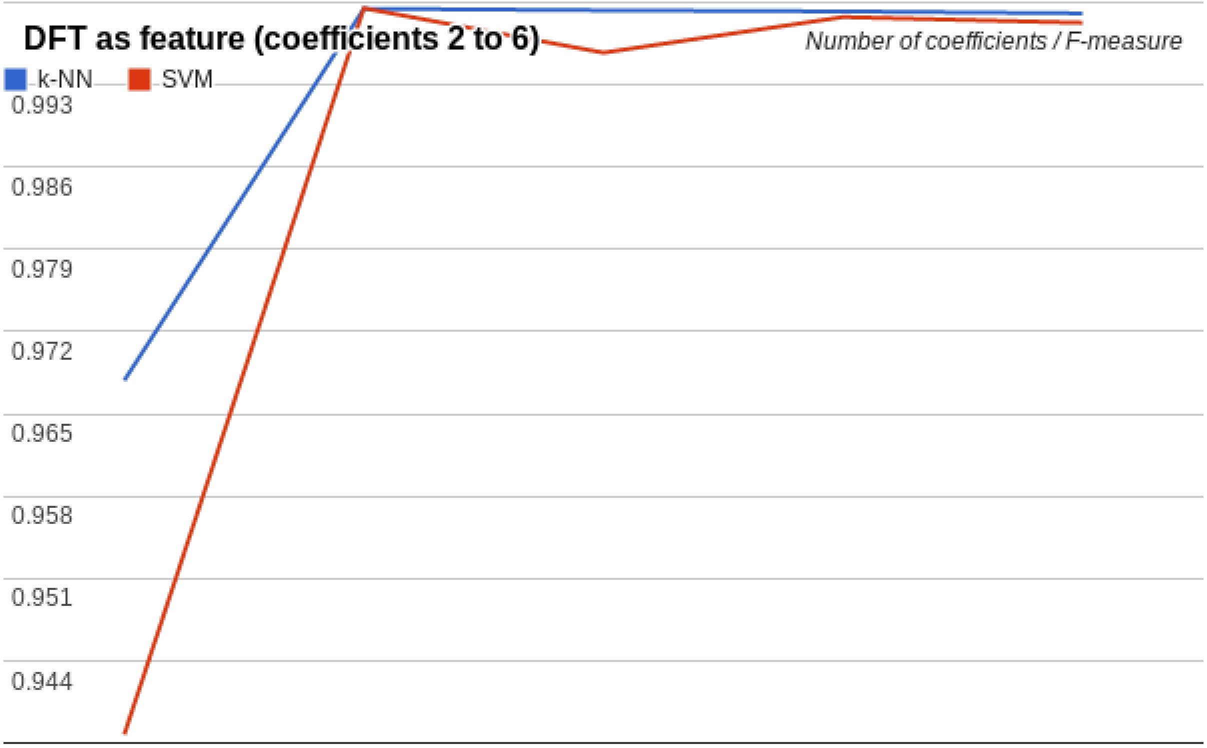
2) DCT - In case of DCT, few 5 coefficients have high standard deviation values which is also reflected from our previous results where we see that highest F-measure value is obtained when we choose first 5 DCT coefficients.



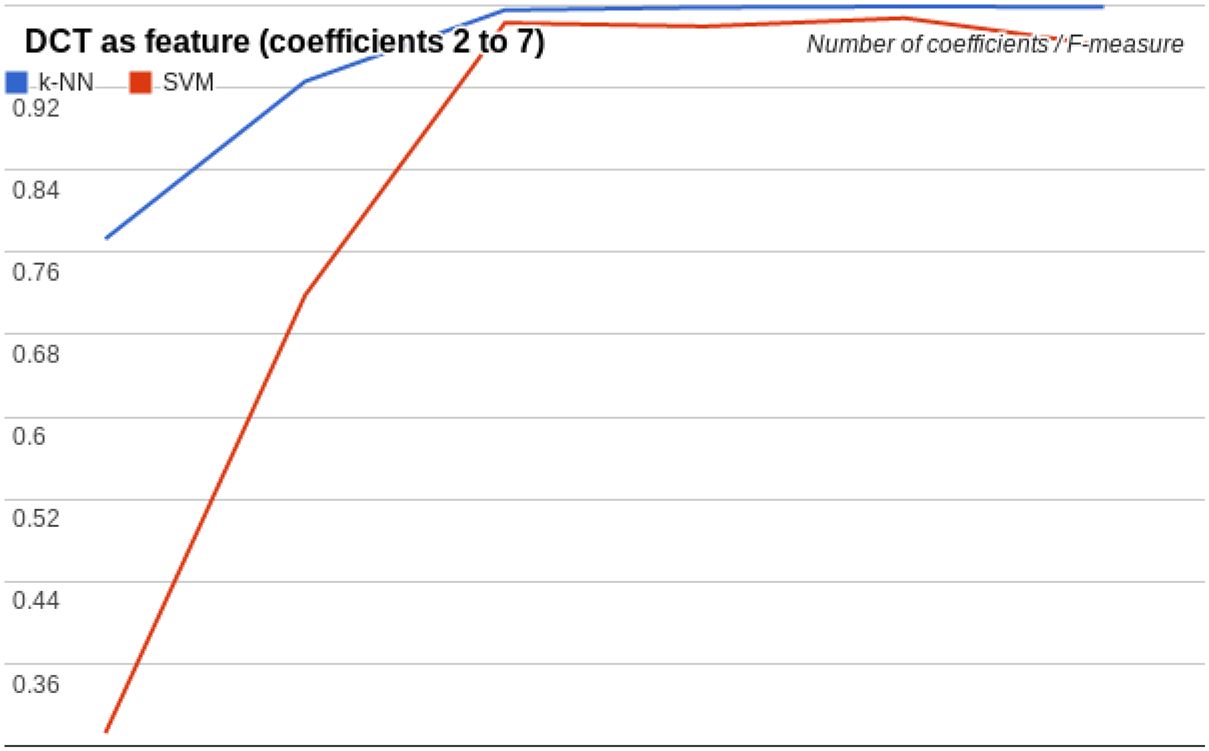
How different classifiers stand against each other?

We see that irrespective of feature selection, k-NN (with k=1) always outperforms SVM in terms of F-measure -

1) Below graph shows how F-measure varies when we use DFT coefficients as features. Features are varied from first 2 to first 6 DFT coefficients -



2) DCT as feature - Below graph shows how F-measure varies when we use DCT coefficients as features. Features are varied from first 2 to first 7 DFT coefficients -

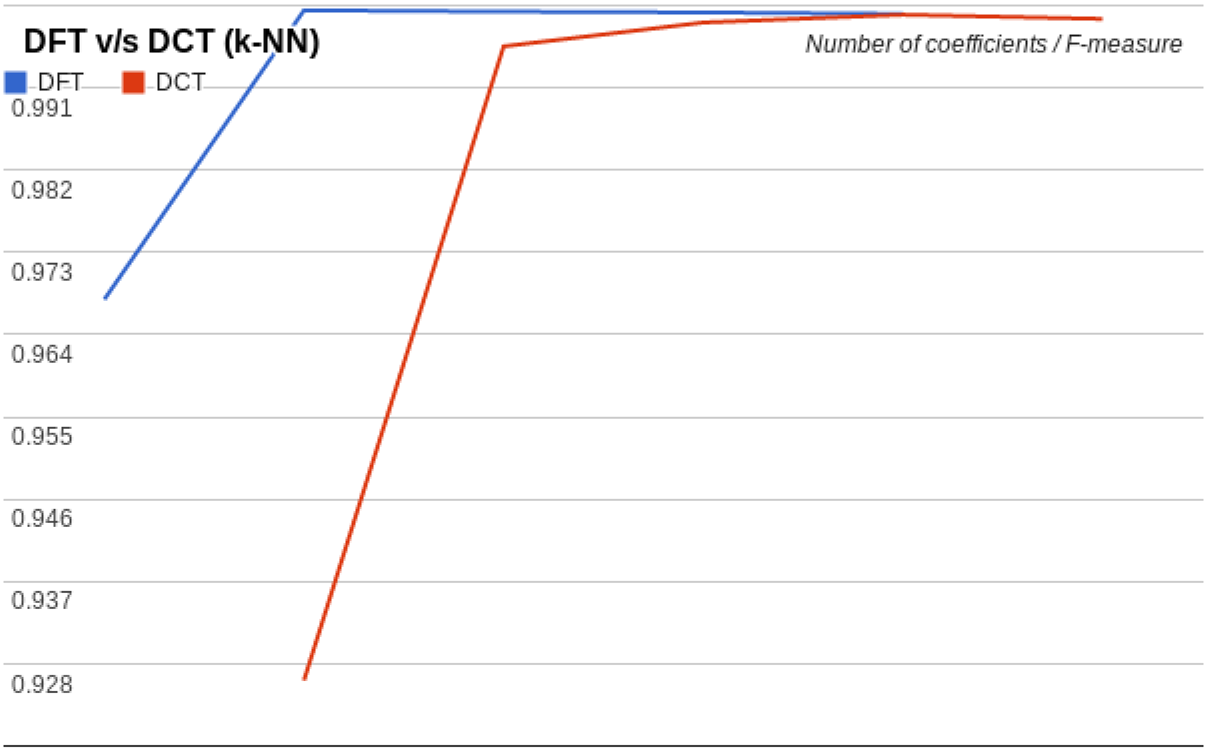


How different features stand against each other?

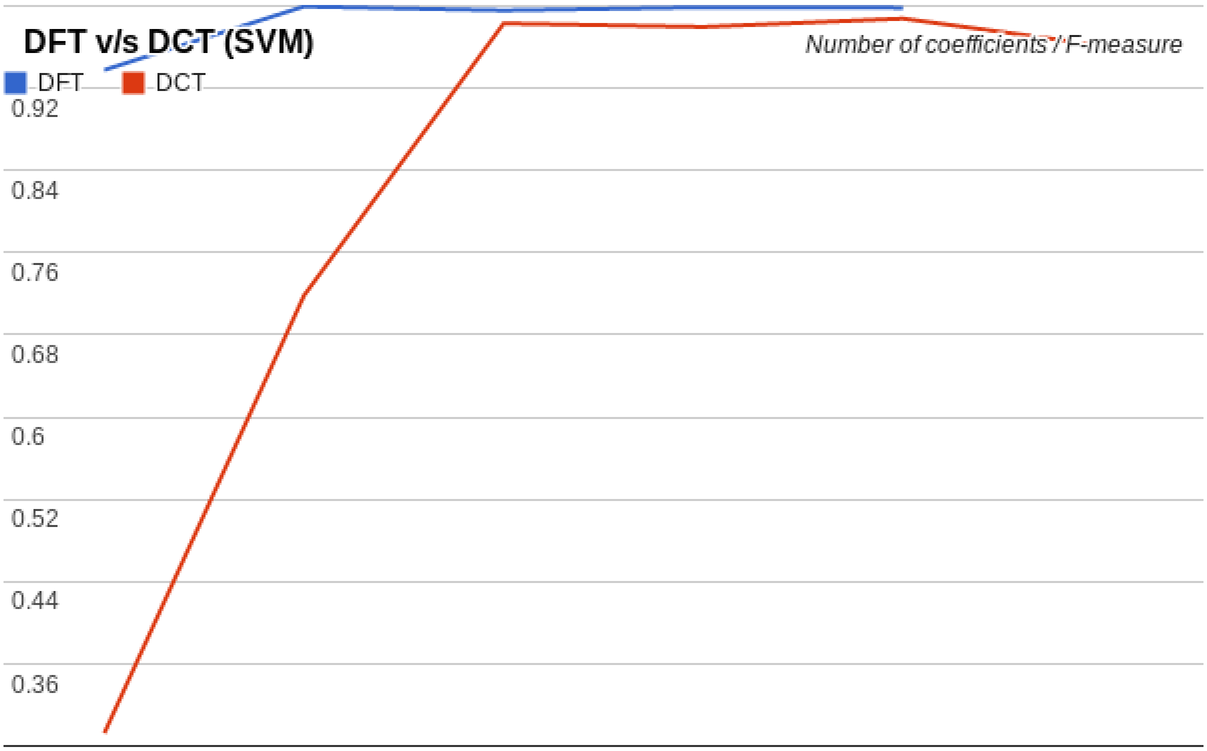
We see that irrespective of classifier used, DFT as feature selection always outperforms DCT as feature selection in terms of F-measure -

Below graphs testify this statement -

1) Variation of F-measure for DFT v/s DCT when number of coefficients are varied and classifier is fixed as 1-NN -



2) Variation of F-measure for DFT v/s DCT when number of coefficients are varied and classifier is fixed as SVM -

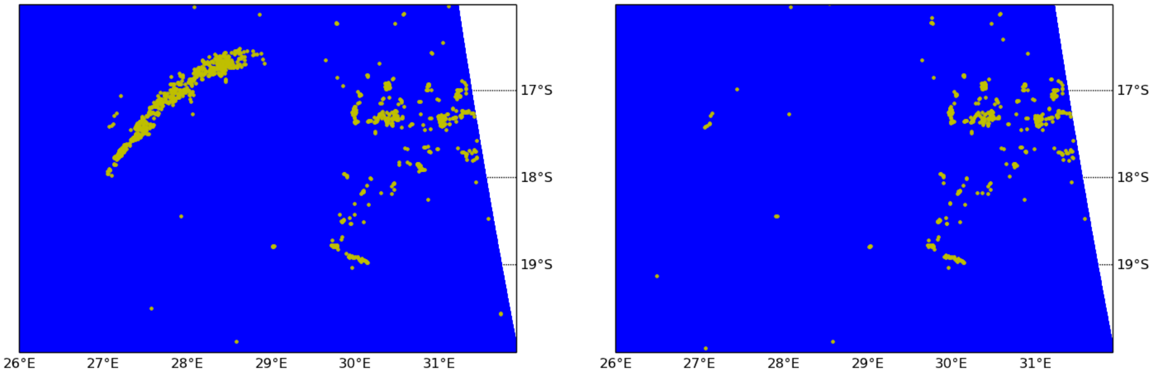


From the above experiments we see that there is significant difference between the performances when feature selection is done using DFT v/s DCT. As DFT has always outperformed we would keep the feature selection method as DFT (with first three coefficients, as they have provided us with the best F-measure values in the train dataset) while dealing with test set.

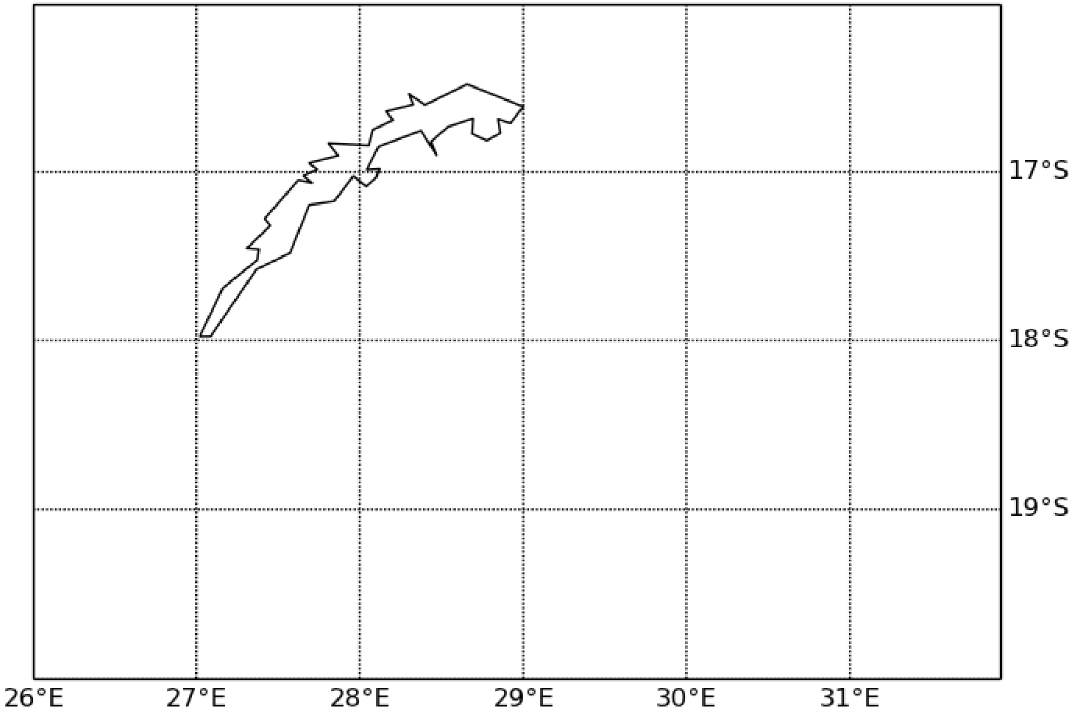
Results:
Below are the SVM and k-NN configurations used -
<u>SVM</u> -
Linear SVC
Loss function - Squared hinge loss
Penalty parameter (C) - 1.0
For penalty calculation - Euclidean distance
Tolerance for stopping criteria - 0.0001

<u>k-NN</u> -
k - 1
Distance metric - Euclidean

Plot for 1st year can is shown below -

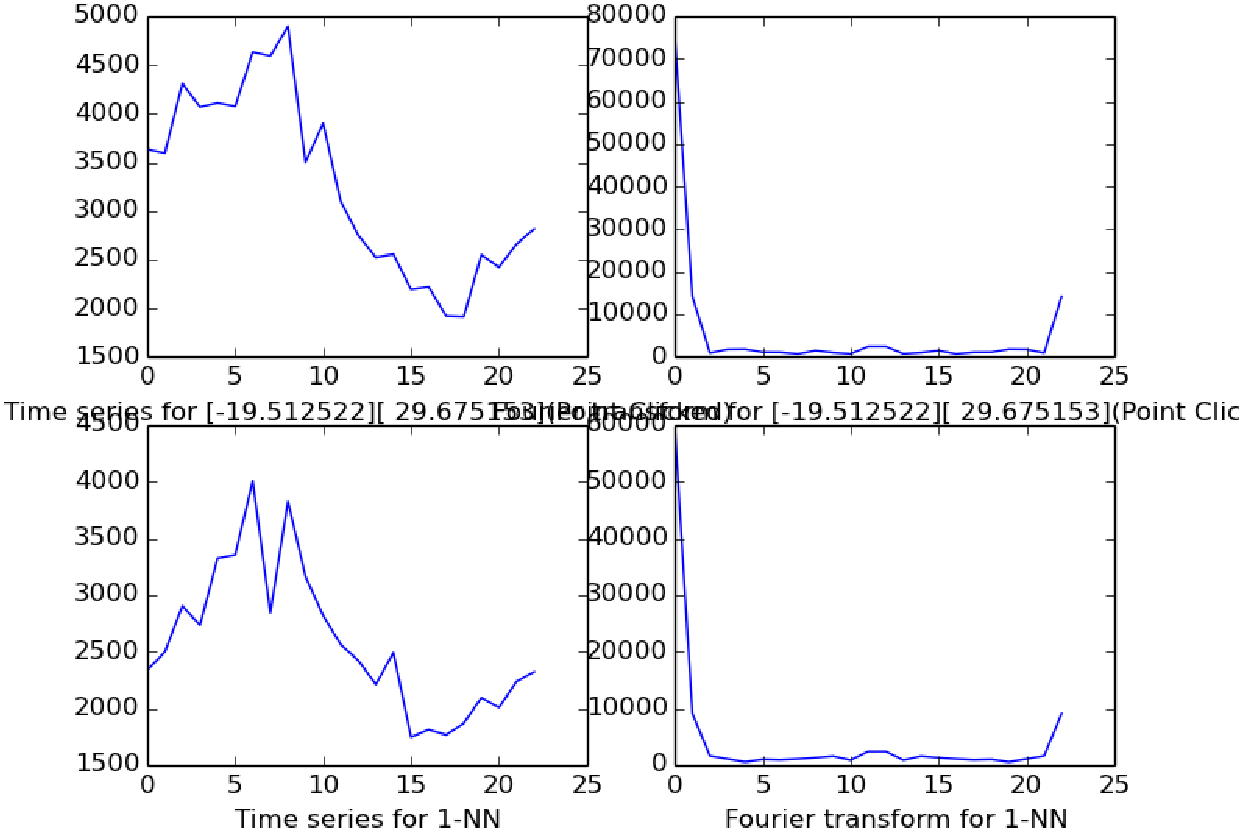


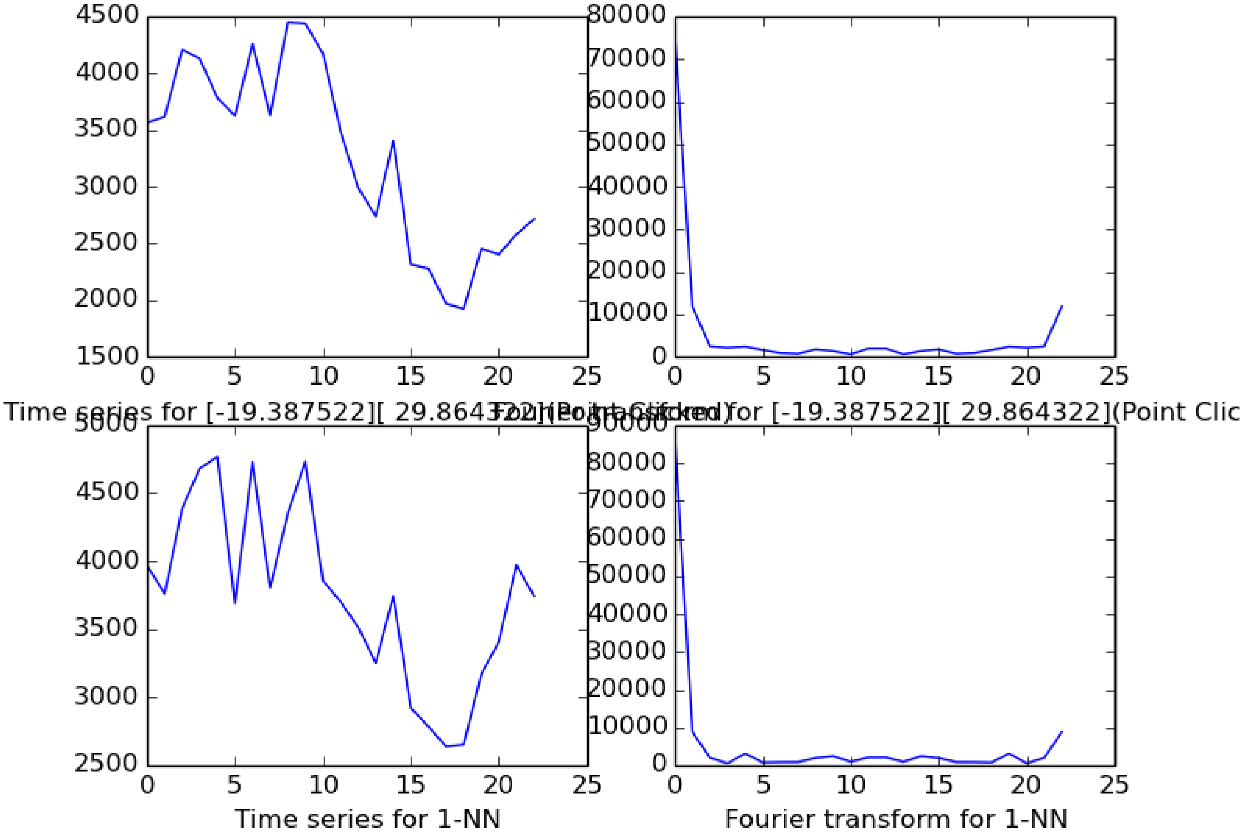
Left side plot is for the classification by SVM and right one is for 1-NN. Bimodal points are colored in yellow, unimodal points are colored in blue and unknown points are marked in white. Map for this region is shown below -



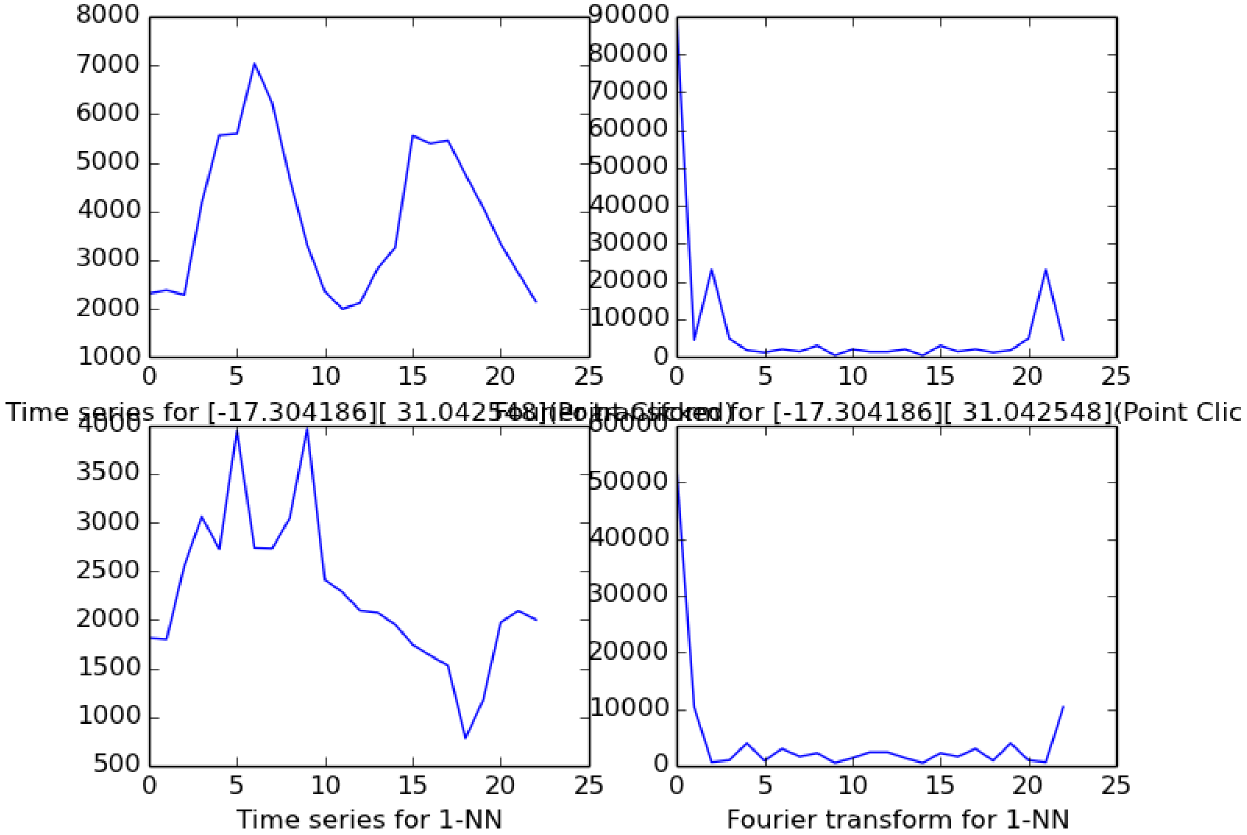
We can see that SVM misclassified points in the water body as bimodal. The water body represented in the above map is "Lake Kariba". This is a major misclassification which is also seen in the plots of subsequent years. These points are not marked as bimodal in k-NN, though they also don't belong to unimodal class as classified by k-NN. But if we consider one v/s rest classification considering bimodal class as our main class and other classes are kept in "rest" we are less concerned about the misclassification done by k-NN.

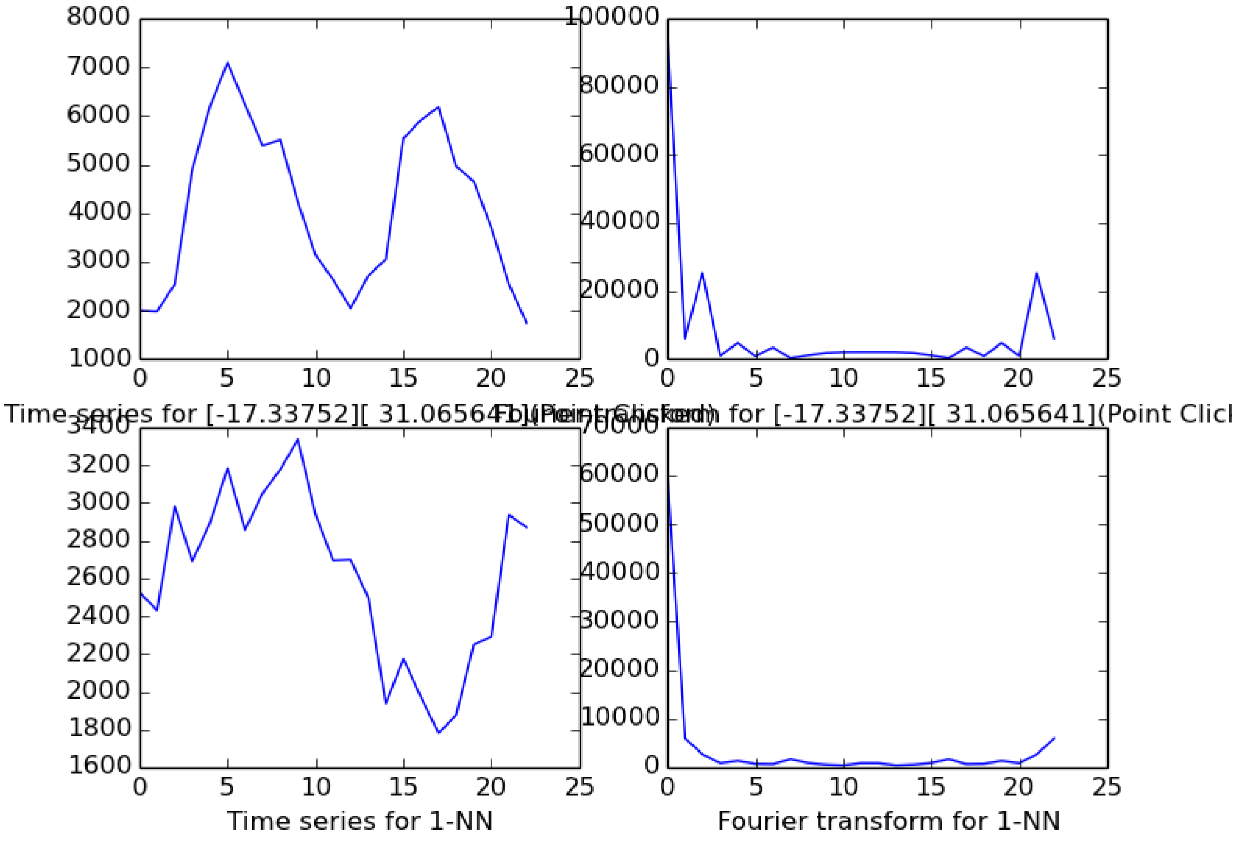
Based on the above statements made we see that k-NN has better performance as compared to SVM. Therefore, moving forward we would concentrate on k-NN and see the misclassification done by it. First let's see some points which are correctly classified by k-NN and their corresponding neighbors. Below two graphs correspond to points (in (lat, lon) format) - (-19.512522, 29.675153) and (-19.387522, 29.864322) which are correctly classified by the classifier as unimodal. We can also see that noise (high frequency peaks) is being correctly handled in the second case. In a plot, the four sub-plots refer to - time series of the point to be classified (upper left), its fourier transform (upper right), time series of 1 nearest neighbor (bottom left) and its fourier transform (bottom right).


5000
4500
4000
3500
3000
2500
2000
1500
0
5
10
15
20
25
Time series for [-19.512522][29.675153](Point Clicked)
80000
70000
60000
50000
40000
30000
20000
10000
0
Fourier transform for [-19.512522][29.675153](Point Clic
4500
4000
3500
3000
2500
2000
1500
Time series for 1-NN
60000
50000
40000
30000
20000
10000
0
Fourier transform for 1-NN



Below two graphs correspond to points (in (lat, lon) format) - (-17.304186, 31.042548) and (-17.33752, 31.065641) which are correctly classified by the classifier as bimodal. We can see the points which are being classified as perfectly bimodal but their 1 nearest neighbors aren't, though they are close to bimodal. This gives us an indication that the training set might not have good representative points for bimodal class.

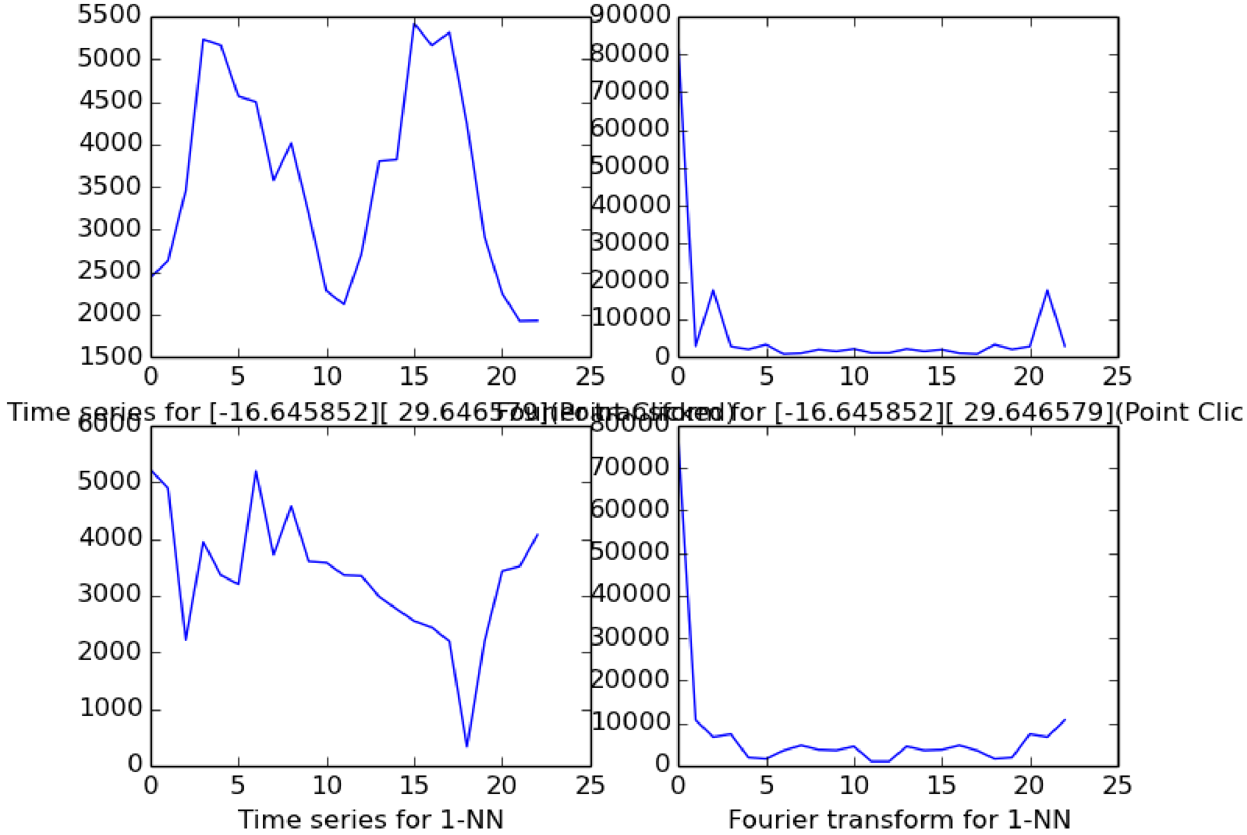

8000
7000
6000
5000
4000
3000
2000
1000
0
5
10
15
20
25
Time series for [-17.304186][31.042548](Point Clicked)
90000
80000
70000
60000
50000
40000
30000
20000
10000
0
Fourier transform for [-17.304186][31.042548](Point Clic
4000
3500
3000
2500
2000
1500
1000
500
Time series for 1-NN
60000
50000
40000
30000
20000
10000
Fourier transform for 1-NN

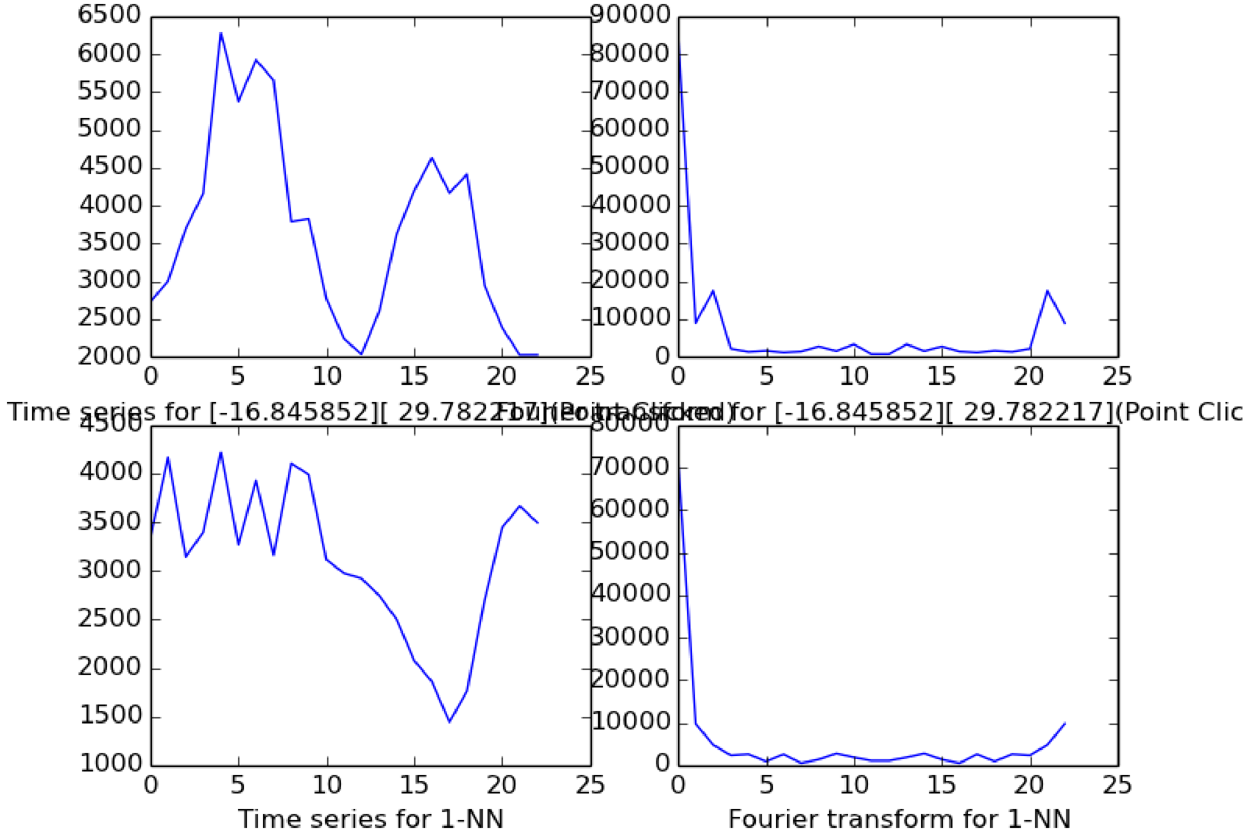


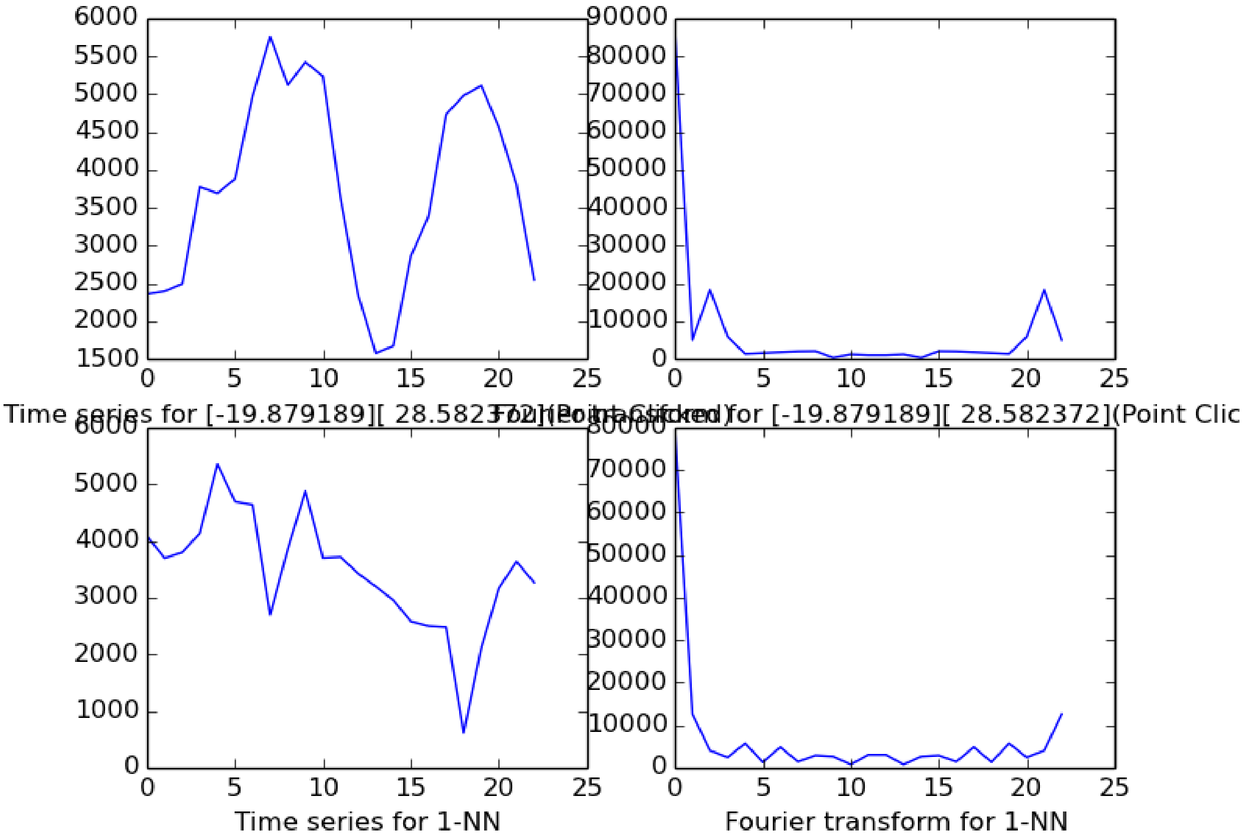
While looking at the correctly classified points I was looking at points surrounded by similar points which means that the points are surrounded by similar class points which represents a land cover of certain type in that region (unimodal or bimodal). Now I would show some misclassified points, which can be outliers or can be misclassified by the classifier. These points are unimodal points surrounded by bimodal points or vice versa.

As said before there can be two cases -

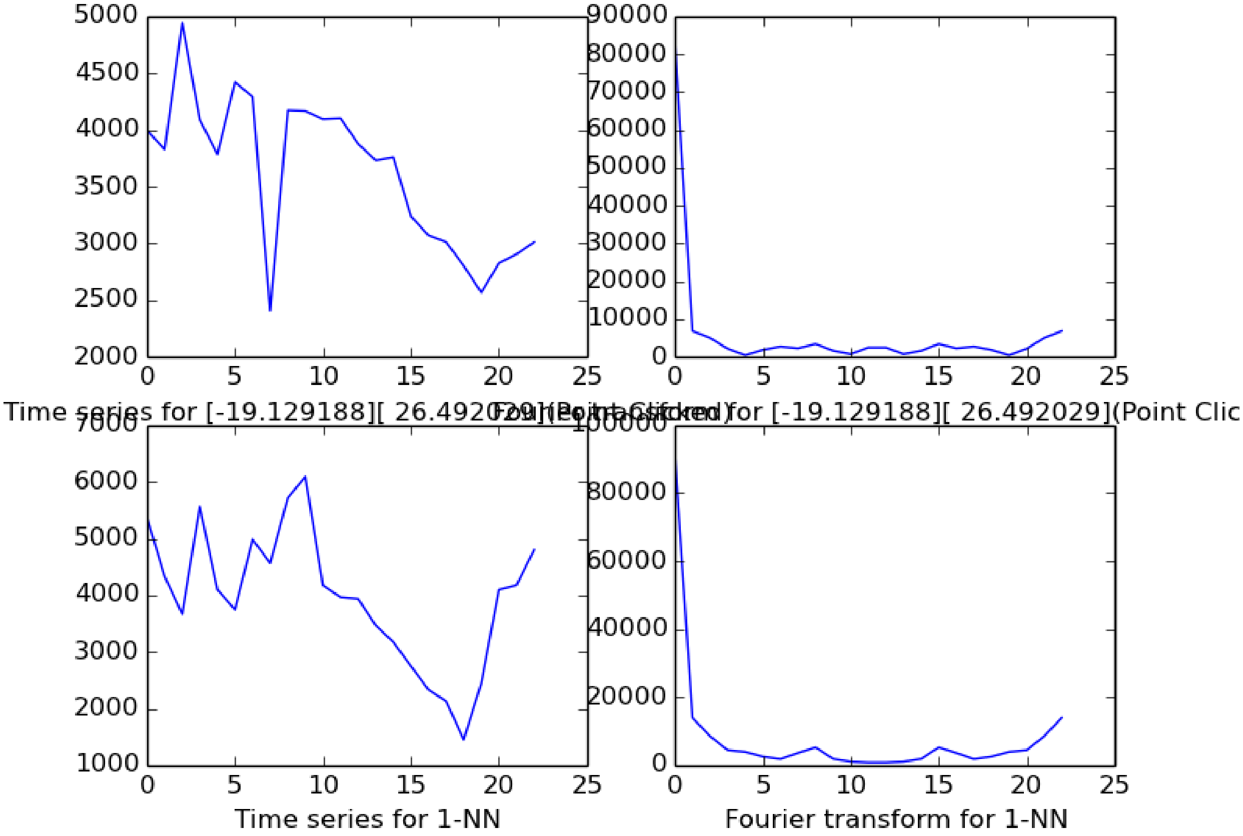
a) Point is an outlier - In this case point isn't misclassified, even though it is surrounded by points of other class. Below are few cases where this is seen. Plots are for points (-16.645852, 29.646579), (-16.845852, 29.782217) and (-19.879189, 28.582372).

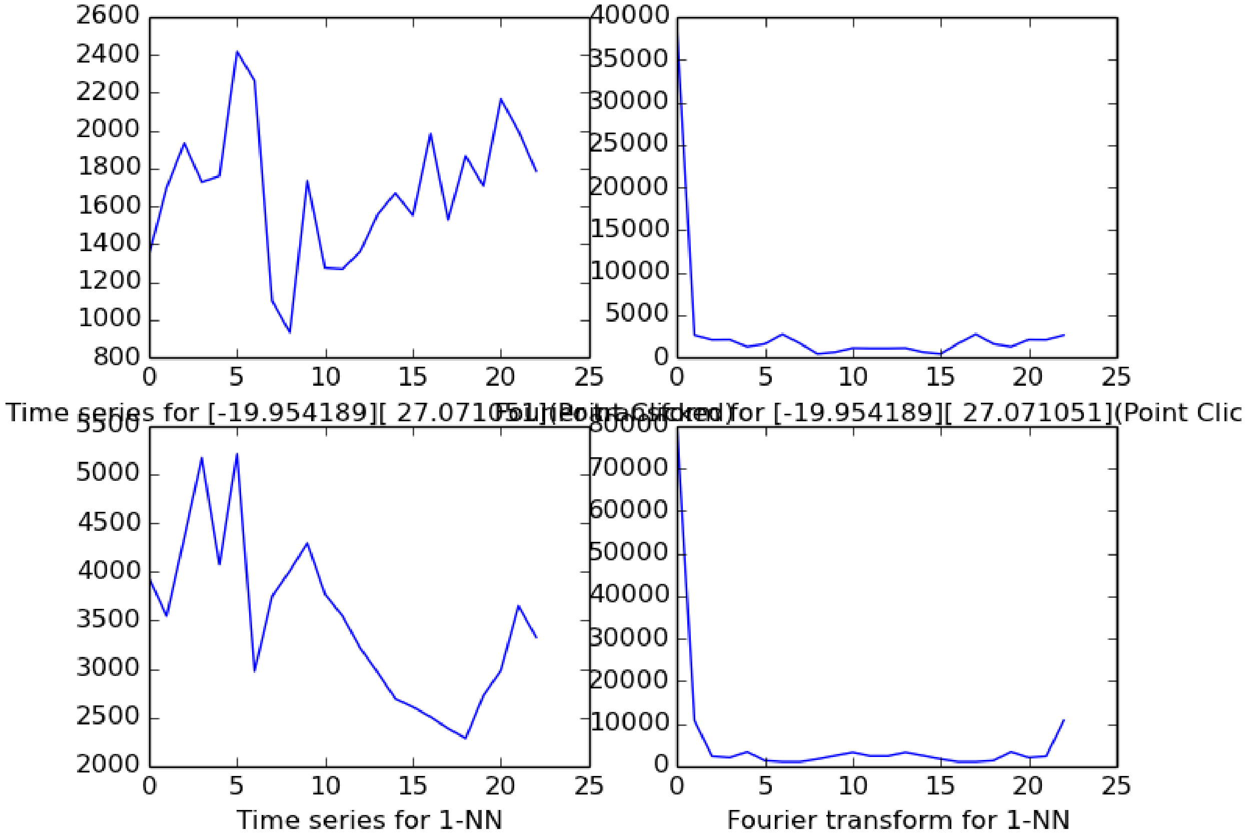

5500
5000
4500
4000
3500
3000
2500
2000
1500
0
5
10
15
20
25
Time series for [-16.645852][29.646579](Point Clicked)
90000
80000
70000
60000
50000
40000
30000
20000
10000
0
0
5
10
15
20
25
Fourier transform for [-16.645852][29.646579](Point Clic
6000
5000
4000
3000
2000
1000
0
0
5
10
15
20
25
Time series for 1-NN
80000
70000
60000
50000
40000
30000
20000
10000
0
0
5
10
15
20
25
Fourier transform for 1-NN


6500
6000
5500
5000
4500
4000
3500
3000
2500
2000
0
5
10
15
20
25
Time series for [-16.845852][29.782217](Point Clicked)
90000
80000
70000
60000
50000
40000
30000
20000
10000
0
0
5
10
15
20
25
Fourier transform for [-16.845852][29.782217](Point Clic
4500
4000
3500
3000
2500
2000
1500
1000
0
5
10
15
20
25
Time series for 1-NN
80000
70000
60000
50000
40000
30000
20000
10000
0
0
5
10
15
20
25
Fourier transform for 1-NN



b) Point is misclassified - In this case after looking at the time series of the corresponding points we see that they actually don’t belong to the bimodal class. Below are few such cases. Plots are for points (-19.129188, 26.492029) and (-19.954189, 27.071051).


5000
4500
4000
3500
3000
2500
2000
0
5
10
15
20
25
Time series for [-19.129188][26.492029](Point Clicked)
90000
80000
70000
60000
50000
40000
30000
20000
10000
0
0
5
10
15
20
25
Fourier transform for [-19.129188][26.492029](Point Clic
7000
6000
5000
4000
3000
2000
1000
0
5
10
15
20
25
Time series for 1-NN
100000
80000
60000
40000
20000
0
0
5
10
15
20
25
Fourier transform for 1-NN



Why misclassification and proposed solution? - We see that the points which are misclassified don't have 1 nearest neighbor as bimodal, which means that there might be some issue with the train set. One solution could be to improve the train set. Another solution could be to run clustering (k-means) on the points classified as bimodal. As we are more interested in finding the dense clusters of bimodal points, we can detect and remove outliers using a clustering technique and separate out the desired bimodal regions.

How the code is organized?

Step 1: Execute "process_train.py". This would generate a pickle file of processed train data (train_data.pickle) from "raw_train_data" which we get from http://gofc.cs.umn.edu/ankush/ModelChangeResults/Version2/FAL_4_40_75_400_700_0P8_h20v10_PointsInfo.txt

Step 2: Execute "features_train.py". This would train the two classifiers, k-NN and SVM. Default feature selection method is DFT. If you want to change the feature selection method to DCT, change "get_features_fourier" to "get_features_dct" in line number 100. By default the number of coefficients selected are 3 and 6 for DFT and DCT respectively (which gave best F-measure in evaluation). To change number of coefficients to be used with DFT, change value of n in line

number 13, to change it for DCT, change value of n in line number 23. Once you run this script, it would generate trained classifier pickles - “train_knn.pickle” and “train_linearsvm.pickle”.
Above experimental results are also calculated from this script. Just uncomment line 97 and the part below line 118 and comment the training part (from line 102 to 116) and execute the code. Default classifier for evaluation is set as k-NN, to change it to SVM, change “apply_knn” in line 136 to “apply_svm”. Executing the script after these changes would do cross validation on train set and give four measures as listed in tables above.

Step 3: Execute “create_fourier_dict.py”. This would create a dictionary where keys are the first three fourier coefficients and values would be a tuple of original time series and its fourier transform. That dictionary is saved in “fourier_dict.pickle”.

Step 4: Execute “main.py”. This would preprocess the test data and and run trained classifiers on test data to generate results from classifiers - “knn_results.pickle” and “linearsvm_results.pickle”

Step 5: Execute “get_plots.py”. This would start generating plots year-wise. For each year plot, there would be two subplots. Left side plot corresponds to the classification of points done using SVM and right side plot corresponds to the classification of points done using k-NN. Points with label 0 (unknown) are kept as white, with label 1 (unimodal) are kept as blue and with label 2 (bimodal) are kept as yellow.
Zoom in before clicking a point. Clicking a point would generate the time series corresponding to that point. In the new plot (point coordinates would appear in the title bar) that pops up, there would be four subplots - upper left plot corresponds to the time series of the point that is clicked, upp

PS: If the script outputs a message stating an operation is finished and doesn’t say to press enter to move onto next operation, please press “Ctrl + C” to exit. This is because there are other experiments that has been done for this research not mentioned in the report.

Plots:
Below is the directory structure of plots which reside in the parent directory “plots” -

- plots_2ndharmonic/ - plots of time series sorted by the third fourier coefficient
- plots_2ndby1stharmonic/ - plots of time series sorted by third divided by second fourier coefficient
- plots_svmknn_yearwise/ - scatter plots of the concerned region year by year. Plots has classification done by both SVM and k-NN
- plots_knn/
    - bimodal_correct/ - Plots of few bimodal points correctly classified
    - bimodal_correct/ - Plots of few unimodal points correctly classified
    - misclassified/
        - bimodal/ - Plots of points classified as bimodal and should be unimodal
        - unimodal/ - Plots of points classified as unimodal and should be bimodal

References:
[1] Classifying Wetland Vegetation Type from MODIS NDVI Time Series Using Fourier Analysis - Xiaodong Na and Shuying Zang
[2] Classifying rangeland vegetation type and coverage from NDVI time series using Fourier Filtered Cycle Similarity - R. Geerken, B. Zaitchik and J. P. Evans
[3] Discrete Cosine Transform - N. Ahmed, T. Natarajan, K. R. Rao
[4] http://astro.berkeley.edu/~jrg/ngst/fft/leakage.html
[5] A comparison of DFT and DWT based similarity search in time series databases - Yi-Leh Wu, Divyakant Agrawal, Amr El Abbadi
[6] Fourier analysis of historical NOAA time series data to estimate bimodal agriculture - F. Canisius, H. Turral, D. Molden

PS: Whole list of papers read for this directed research is available at - http://www.mendeley.com/groups/4515441/directed-research/papers/