Summary.

Dynamic time warping: dynamic time warping (DTW) is an algorithm for measuring similarity between two temporal sequences which may vary in time or speed

Classic/ Derivative may fail to align pair of sequences and also lack the capability of any supervised learning.

The proposed method is Feature Based Dynamic time warping. This method align two sequences based on each point's local and global features instead of its value and derivative.

It outperforms both the previous two techniques.

To improve upon the FBDTW, we design a method that gives the system a capacity for supervised learning. This is called Adaptive Feature Based Dynamic Time Warping. (AFBDTW)

The supervised learning algorithm is a dual learning process. The first fold learning process learns the significances of both the local and global feature towards classification, then the second fold learning process learns a classification model based on the pairwise distances generated by AFBDTW

DTW finds an optimal warping path between R and Q by using dynamic programming to calculate the minimal cumulative distance which is defined recursively.

Aligning two temporal sequences using the distance between the data points of two different sequences is valued based DTW.

When a data point in a sequence is compared with the data point in another sequence the relation to its neighbours should be taken into consideration.

Derivative DTW can be viewed as local feature of the data point that expresses its relationship into two adjacent neighbours.

Thus FBDTW is proposed as a better technique for evaluating similarities as it takes into account both the local and global features of the two points.

The second contribution in this paper is the enhancement of the supervised learning capacity of DTW through a learning algorithm

The AFBDTW is defined where the contributions of the global and local features are leveraged by weighting factors.

The method proposed in the paper. :

Two temporal sequences of M and N instances is taken.

A NxM matrix is then made in which each node is the distance between various data points of the two sequences.

Here the distance is then defined as a sum of the local feature distance and the global feature distance.

The local feature and global feature of a data point are explicitly mentioned

Then the optimal warping path is then found using dynamic programming to calculate the minimum cumulative distance which was defined recursively before.

Time complexity is O(MN)

Second contribution:

The learning capacity should be equipped to a time series classification approach to learn an optimized way to calculate pairwise distances from the training data.

$dist(r_i,q_i)= w1*dist_{local}(r_i,q_i)+w2*dist_{global}(r_i,q_i)$

An algorithm called In-Class-Range Weighting Algorithm is designed to learn the weighting factors w1 and w2 from the training data.

From each training sequence it calculates the difference between the number of same class sequences within the in-class range and the number of different class sequences within the in-class range

w_i=summation of ($numSameClass_x$-$NumDiffClassInRange_x$)

X states the class of the sequence.

Dynamic Programming Concept. :

Once you have your distance matrix where every node defines the distance (as was defined in the paper) of the data points in two sequences.

To create a mapping of the two signals we need to create a path in the above matrix. The path should start from (0,) and want to reach (M, N). Our aim is to find the path of the minimum distance

For this purpose we build another matrix similar to that of the distance matrix. This matrix would contain the minimum distance s to reach a point.

The node of this matrix could be the accumulated cost to reach that point.
The cost function is the matrix can be recursively defined as

//pseudo-code

```
for I in range(1,len(y)):
 for j in range(1,len(x)):
   accumulated_cost[i,j]=min(accumulated_cost[i-1,j-1],accumulated_cost[i,j-1],accumulated_cost[i,j-1])+distancce[i,j]
```

x and y are the temporal sequences and (i,j) correspond to the matrix.

Once we have this matrix we would be doing backtracking to find the path minimizing the distance.

Backtracking procedure is fairly simple and involves trying to move back from the last point (M, N) and finding which place we would reached there from and do this in repetitive fashion

Thus you will have your optimum warping path which minimizes the sum of the distance along the path.

Additional Doubts and Information that is required:

What all methodologies are we required to implement when you say a fully functional dynamic time warping library

I have extensively read the paper and have a thorough understanding of the system to be implemented. The summary/understanding of which I have stated above. I have also given a procedure to implement the methodology stated in the paper. I understand that the implementation of dynamic programming is similar to that of the edit distance problem

I am presently going through the DTW package is R
Some links that I am following are:

http://www.jstatsoft.org/v31/i07/paper

It would be highly appreciated if you give me more insight regarding the project and what would be the role of a person if he/she is selected for this project.

I would also appreciate links that can better my understanding regarding various topics.