# CS 598 HAZ Midterm Checkpoint for Course Project

Pratikshit Singh, Simran Sandhu, Siddharth Patra

TOTAL POINTS

## 10 / 10

QUESTION 1

*1* **10 / 10**

✓ **- 0 pts** *Correct*

**- 1 pts** Click here to replace this description.

**- 5 pts** Click here to replace this description.

*1* **10 / 10**

✓ **- 0 pts** *Correct*

**- 1 pts** Click here to replace this description.

**- 5 pts** Click here to replace this description.

# Smart Transfer Learning methods for time series classification

Pratikshit Singh
ps71@illinois.edu

Simran Sandhu
sandhu5@illinois.edu

Siddharth Patra
patra3@illinois.edu

## Abstract

The objective of this research project is to investigate the domain of transfer learning in Time Series Classification (TSC) problems. The focus is on exploring methods to analyze the best source dataset for improving accuracy for the Neural Network model trained on target time dataset. To achieve this, the project aims to leverage various similarity algorithms (measures) for time series datasets. The experiments are performed using the UCR archive, the largest publicly available TSC benchmark containing 85 datasets. This midterm project checkpoint focuses on implementation of similarity measures such that they can be leveraged to find the best source dataset for a given target dataset. Further, the project will compare the results of the similarity measures and select the best measure for the final combination of datasets to be used for transfer learning on the Nerual Network model. The final phase of the project will include testing to evaluate the performance of the Neural Network model after transfer learning, explain the results and compare the performance with the original DTW DBA based approach.

## 1 Introduction

Through this research project we aim to explore the transfer learning problem domain, particularly domain adaptation for Time Series classification(TSC) problems [1]. Transferring deep FCN for TSC seems to deliver high accuracy results by fine tuning a pre-trained model. The model prediction's improvements or degradations in performance depend on the choice of source dataset for the transfer learning approach. Thus, we aim to find methods to analyze the best source of time series dataset for pre-training FCN models that can be transferred to improve accuracy for the targeted time series dataset. We plan to explore and leverage various similarity measures between time series datasets to improve model predictions.

## 2 Motivation

Generalization with transfer learning has shown significant accuracy improvements in classification tasks[2]. Deep CNN image classification transfer models like ResNets [3] and others trained on large dataset - ImageNet dataset [4], containing 100 classes, gave substantial accuracy improvements.

However, time series data is highly dependent on the underlying trends, patterns, and dependencies over time. These patterns can vary significantly across different time series classification tasks, making it more challenging to transfer the knowledge learned from one task to another. Hence, it is more important to choose an optimum source dataset which can be used for pre-training deep FCN model. Thus, transfer learning and similarity measures with time series has many unexplored fields and has motivated us to research in this area.

## 3 Progress

- Latest Dataset procured from the UCR archived

- Data visualization for datapoints in dataset for better understanding of the data.
- Finding relevant similarity measures to work with time series datapoints.
- Implementing and getting results for similarity measures on the combination of dataset from the list of dataset provided below.
- Selection of datasets to transfer information on, since with the help of similarity measures we can propose to set combination of datasets (source, target) that may provide us with positive transfer learning.

# 4 Dataset

We perform experiments using the datsets from the UCR archive [5] which is the largest publicly available TSC benchmark containing 85 datasets. The datasets range from 200-2000 time steps, and we come up with different ways to align those time steps based on the requirements of the distance/similarity measures. Below are plots for an example dataset- Beef plotting all its time series(left) and only plotting the DBA generated prototypes(right) for it.

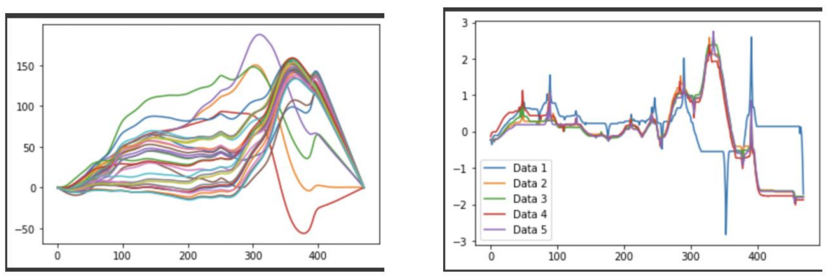


Figure 1: Plotting all the TS vs just 1 protoype for each class

# 5 Methodology

This section talks about in detail the experiments that we have perform as well as their expected performance evaluation. Any conclusions that we draw based on the intermediate in the next section is based on the implementation of the similarity metrics described in this section.

## 5.1 Per-Class Prototype

In order to compute the similarities between the datasets, we need a per-class representation time series in each dataset befoer we can start with comparisons. The per class representation/prototype is computed by using a measure of center for the set of time series in the corresponding class. The DTW Barycenter Averaging (DBA) method to average a set of time series appears to be appropriate. Specifically, the DBA algorithm is a summarizing function for averaging method in the induced space. To generate the similarity matrix between the UCR datasets, we compute a distance between each pair of datasets through the prototypes computed.

## 5.2 Similarity Measures

We do a literature survey in related work such as [6] and find some similarity measures that may give a performance benefit with transfer learning approaches. We've identified below potential candidates for measuring the similarity between datasets.

- Proxy A-Distance (PAD)[7]: Proxy-A-Distance (PAD) is a distance metric used for measuring the similarity between time series data. The proxy series is constructed by taking the absolute differences between consecutive values in the original time series and then summing those differences. Once the proxy series is constructed for each time series, the PAD distance between the two time series is calculated as the Euclidean distance between their respective proxy series.

- Pearson Correlation Coefficient: Pearson correlation is a commonly used measure of the linear relationship between two variables which can also be used as a similarity measure between two time series. To use Pearson correlation as a similarity measure for two time series we first need to align the two series so that they have the same time steps and then compare each value from both the time series on the same time step. The Pearson correlation coefficient ranges from -1 to +1, where -1 indicates a perfect negative correlation, 0 indicates no correlation, and +1 indicates a perfect positive correlation.
- DBA (DTW Barrycenter Averaging)[8]: We used DBA, which is a dynamic programming based algorithm to find the minimum distance or the most similarity between two time series based on the best alignment of the two time series, even the time series vary in speed and length. The similarity or dissimilarity of two-time series is typically calculated by converting the data into vectors and calculating the Euclidean distance between those points in vector space.
- Time lagged cross correlation: Time lagged cross correlation (TLCC) can identify directionality between two signals such as a leader-follower relationship in which the leader initiates a response which is repeated by the follower.

### 5.3 DBA- DTW Barycenter Averaging:

#### 5.3.1 Reasons to use DBA:

- Any distance (Euclidean, Manhattan, . . . ) between the two-time series can be used for the comparison
- DTW gives a non-linear (elastic) alignment between two-time series. Simply, it looks for the best alignment between the two-time series. This produces a more intuitive similarity measure, allowing similar shapes to match even if they are out of phase in the time axis.

#### 5.3.2 Algorithm:

The basis of DTW [9] is found on the computations of distance /confusion matrix between two-time series. A warping function is used to find the best alignment between A and B, we need to find the path through the grid.(fig below)

$P = p1, \ldots, ps, \ldots, pk$
$ps = (is, js)$

which minimizes the total distance between them. Here P is called a Warping Function.

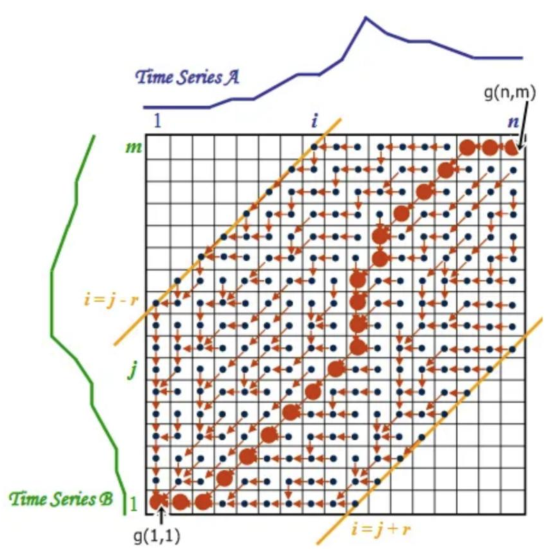


Illustration on How DTW Works

#### 5.3.3 Warping window:

A good alignment path is one that wanders too far from the diagonal.

$|is - js| \leq r$ where $r \geq 0$ is the window length.
Warping window guarantees that the alignment does not try to skip different features and gets stuck at similar features.

#### 5.3.4 Working algorithm:

- Start with the calculation of g(1,1) = d(1,1).
- Calculate the first row g(i, 1) =g(i–1, 1) + d(i, 1).
- Calculate the first column g(1, j) =g(1, j) + d(1, j).
- Move to the second row g(i, 2) = min(g(i, 1), g(i–1, 1), g(i –1, 2)) + d(i, 2). Book keep for each cell the index of this neighboring cell, which contributes the minimum score (red arrows).
- Carry on from left to right and from bottom to top with the rest of the grid g(i,j) = min(g(i, j–1), g(i–1, j–1), g(i –1, j)) + d(i, j).
- Trace back the best path through the grid starting from g(n, m)and moving towards g(1,1) by following the red arrows.

The complexity of computing DTW is O(m * n) where m and n represent the length of each sequence.

## 6 Intermediate Results

We calculate the similarity between datasets based on the different techniques described in section 5 and prepare the best source and target dataset combinations.

### 6.1 Proxy-A-Distance

Figure 2 provides a heat map of PAD values when source and target dataset are taken as combination. For faster computation, we computed the upper triangle matrix as the lower half computes to be the same value. For example, from the figure we observe that 'FaceFour' dataset has high similarity with 'Car' dataset, i.e. 'FaceFour' will be a great choice as a 'source' dataset for the target 'Car' dataset.

### 6.2 Time Lagged Cross Correlation

From the figure 3 you can see at lag of 20 time steps for 'beef' time series it correlates the highest with 'chicken' time series, meaning that 'beef' time series is leading the 'chicken' time series(follower in this case). But as expected there is no such relation between time series of chicken and beef in real world context is taken, hence TLCC is not exactly a great measure in this case.

### 6.3 Pearson's Coefficient

In order to calculate the pearson correlation similarity meausre, we started with 5 train datasets 'Meat', 'InlineSkate','50words','DiatomSizeReduction','Wine'. Figure 4 shows a heatmap with higher correlation amongst the data-sets shown by a darker color. For example, for '50words' as the target dataset, we choose 'DiatomSizeReduction' as the source dataset because it has the highest correlation.

### 6.4 Dynamic Time Warping

Results from DBA over different dataset combinations Nearest neighbours calculated using DBA for selected dataset and the distance calculated between some specific dataset combination are as:

- RefrigerationDevices-ShapeletSim: Most similar class(source-target): (1, 1); Minimum distance between them: 12.555812838330903 - This is NN pair

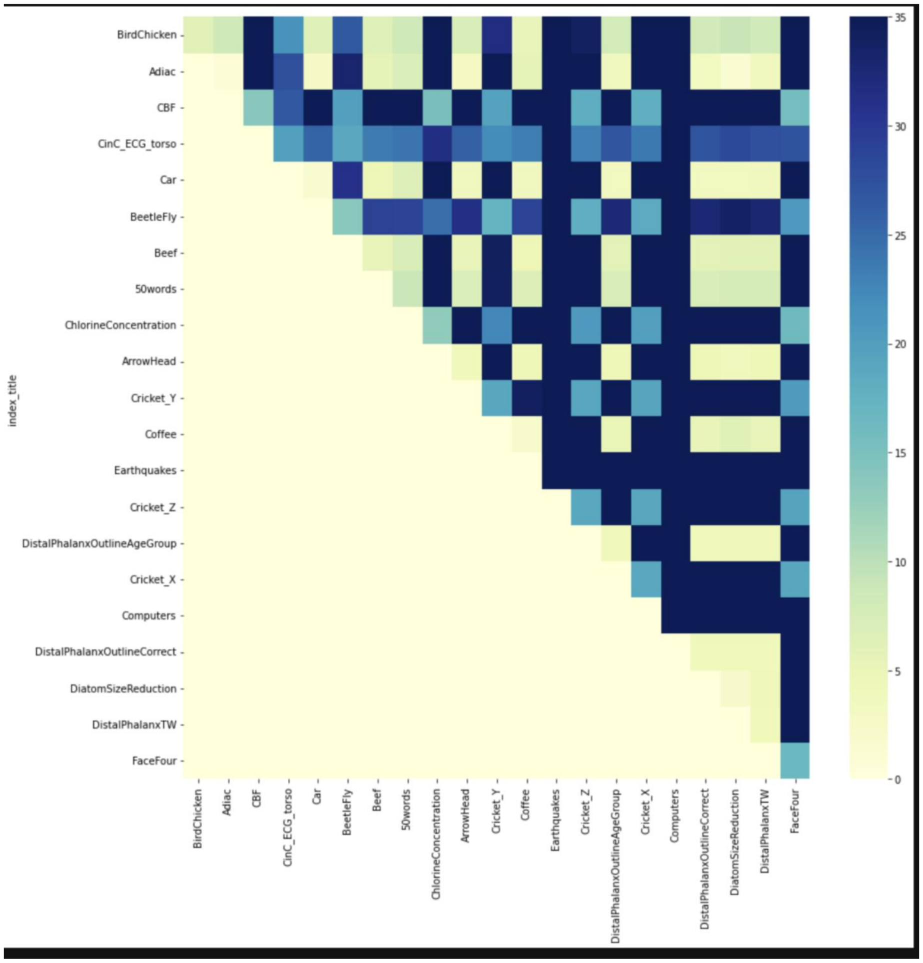


Figure 2: Heatmap showing the results of PAD distance metric experiments

- Car-ShapeletSim: Most similar class(source-target): (2, 1); Minimum distance between them: 16.738063197450185 - Not a NN pair
- 50words-HandOutlines: DBA caluclation not be performed due to Google Colab RAM reaching memory limit( 12.7GB)
- WordsSynonyms-HandOutlines: DBA caluclation not be performed due to Google Colab RAM reaching memory limit( 12.7GB)
- Strawberry-Meat: Most similar class(source-target): (2, 3); Minimum distance between them: 2.1764866573784327 - This is NN pair
- Beef-Meat: Most similar class(source-target): (3, 1); Minimum distance between them: 3.738592244173742
- ShapesAll-DiatomSizeReduction: Most similar class(source-target): (58, 3); Minimum distance between them: 0.3778433263677064 - This is NN pair
- Beef-Wine: Most similar class(source-target): (4, 1); Minimum distance between them: 5.365530898885693 - This is NN pair

# 7 Technical Challenges

- For running the DBA algorithm for large datasets like 'handoutliers' dataset, google colab is running out of memory for distance calculation, since it is a dynamic programming implementation. Google Colab RAM reaching memory limit( 12.7GB)

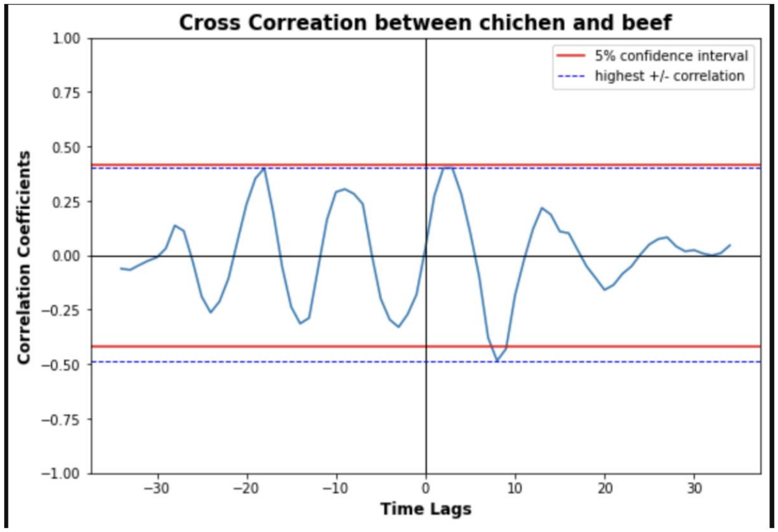


Figure 3: Graph showing time lagged cross correlation between 'chicken' and 'beef datasets

| | Meat_TRAIN | Wine_TRAIN | DiatomSizeReduction_ | InlineSkate_TRAIN | 50words_TRAIN |
|---|---|---|---|---|---|
| Meat_TRAIN | | 0.141 | 0.302 | 0.366 | 0.271 |
| Wine_TRAIN | 0.141 | | 0.514 | 0.506 | 0.626 |
| DiatomSizeReduction_T | 0.302 | 0.514 | | 0.253 | 0.753 |
| InlineSkate_TRAIN | 0.366 | 0.506 | 0.253 | | 0.439 |
| 50words_TRAIN | 0.271 | 0.626 | 0.753 | 0.439 | |

Figure 4: Heatmap showing pearson correlation values

- Another problem we encounter is that not all similarity measures can work well or provide extensive insight since the context they work on is different to the problem statement we have, for example Time lagged cross correlation is a good measure to find out the point (time step) at which max correlation between two time series is observed, but the actual correlation is of little significance.

## 8 Potential solution to overcome

- For google colab running out of memory issue, we plan to split our train and test datasets into smaller chunks and then merge the results

## 9 Conclusion and Future Work

By this midterm project checkpoint we have successfully completed the implementation of similarity measures such that they can be used to find the best source dataset for a given target dataset. Our next steps are to select the source dataset for the final combination of datasets (source and target combinations) which will be used for transfer learning on the Nerual Network model. We will test out the combinations of the datasets to find out relying on which similarity measure gives the best results. Finally, we analytically explain and critically analyze the obtained results.

## References


[1] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Transfer learning for time series classification. In *2018 IEEE international conference on big data (Big Data)*, pages 1367–1376. IEEE, 2018.

[2] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963, 2019.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[5] Dau, Hoang Anh and Keogh, Eamonn and Kamgar, Kaveh and Yeh, Chin-Chia Michael and Zhu, Yan and Gharghabi, Shaghayegh and Ratanamahatana, Chotirat Ann and Yanping and Hu, Bing and Begum, Nurjahan and Bagnall, Anthony and Mueen, Abdullah and Batista, Gustavo, and Hexagon-ML. The ucr time series classification archive, October 2018. `https://www.cs.ucr.edu/~eamonn/time_series_data_2018/`.

[6] Joan Serra and Josep Ll Arcos. An empirical evaluation of similarity measures for time series classification. *Knowledge-Based Systems*, 67:305–314, 2014.

[7] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006.

[8] François Petitjean, Alain Ketterlin, and Pierre Gançarski. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3):678–693, March 2011.

[9] *Dynamic Time Warping*, pages 69–84. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.