1. Multi source aggregated TCP Algorithm

Let us consider a binary classification problem, and suppose we have a training dataset Z and external test data set X, or we randomly partition the given dataset into training (80%) and external test set (20%). The algorithm for aggregated TCP from multiple sources is as follows (see Figure 1).

1. The training data set is randomly split into K parts (disjointly) with varying sizes. For example, Let $Z = \{z_1, ..., z_n\}$ be the data set, then we divide the dataset into $S_1, ..., S_K$ such that $Z = \bigcup_{i=1}^{K} S_i$, $k_i = |S_i|$ and $n = k_1 + ... + k_K$.
2. We compute p-values using (X, S_i) for each S_i, say p_i for each class, then we finally aggregate the k, p-values (weighted average).
3. We repeat the step 1 and step 2 with different values of K and $k_i's$ say for q times.
4. Then we analyze the q results obtained (this part is not clear yet).

Algorithm 1: Multi source aggregated TCP

Input: (training dataset:Z, test dataset:X, label set:Y, a nonconformity measure:$\mathcal{A}$)
Output: Aggregated p-values
Initialization;
Unequal size partition: Partition training.data randomly and unequally into K parts, $S_1, ..., S_K$;
Steps;
for *each* S_i, $i \in \{1, ..., K\}$ **do**
  for *each observation* $x_j \in X$ **do**
    Compute p-values by using **TCP** algorithm:
    $PValues_i = \mathbf{TCP}(S_i, x_i, Y, \mathcal{A})$;
  end
end
Aggregate $PValues_i$ from various sources into a set **p-values**
return p-values

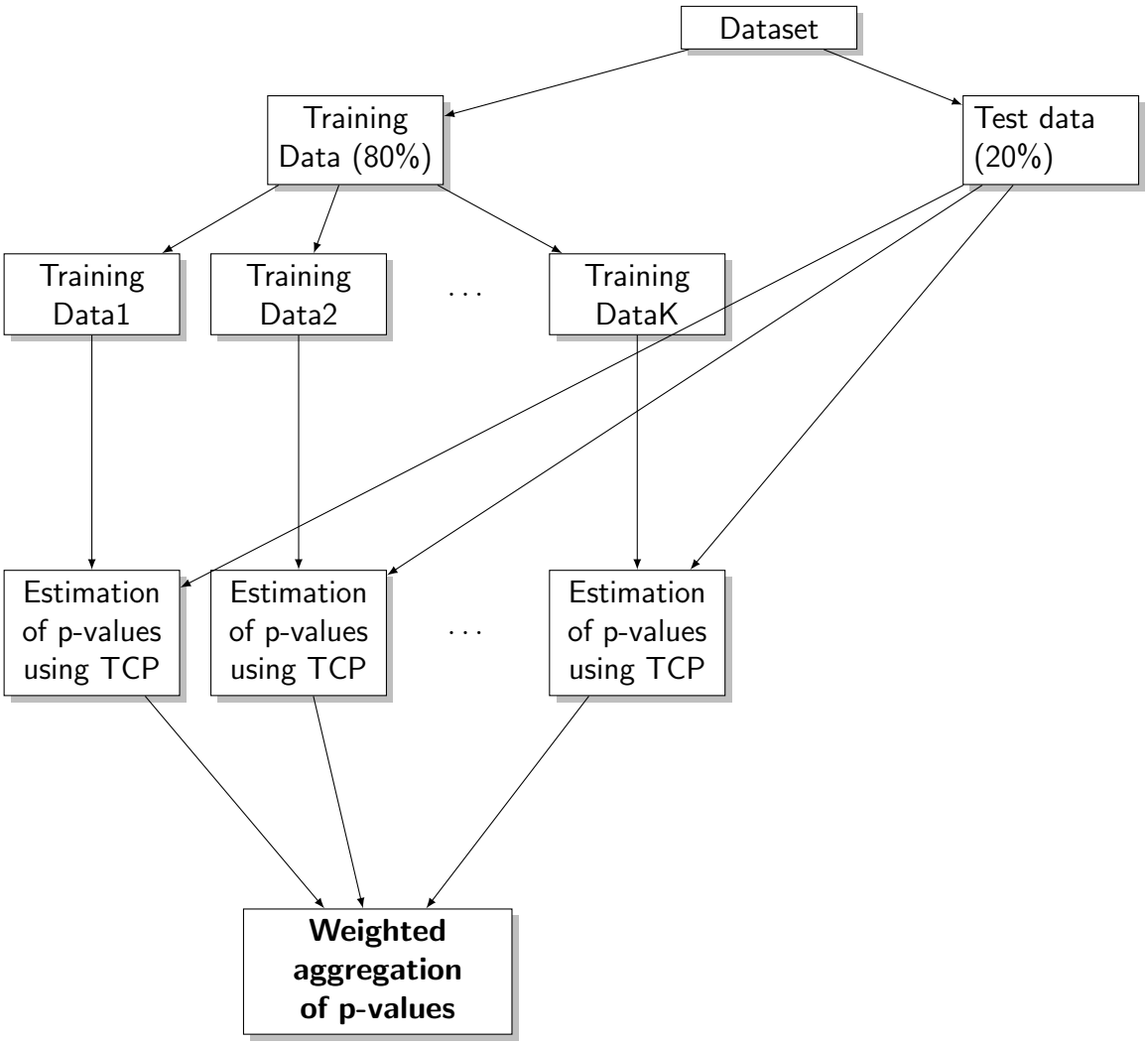

Figure 1: Multi source aggregated TCP Algorithm