Applicability of Conformity Scores for Transfer Learning

Rohan Deshmukh

Submitted for the Degree of Master of Science in

Artificial Intelligence

Department of Computer Science
Royal Holloway University of London
Egham, Surrey TW20 0EX, UK

December 19, 2022

Declaration

This report has been prepared on the basis of my own work. Where other published and unpublished source materials have been used, these have been acknowledged.

Word Count: 8744

Student Name: Rohan Deshmukh

Date of Submission: 19th December 2022

Signature: Rohan Deshmukh

Abstract

The objective of this project is to develop a prediction model with conformal prediction that maintains reasonable accuracy, validity, and efficiency for different cases of transfer learning. We will use decision trees to implement conformal prediction on a classification problem. We will study the process of generating data, building a decision tree, tuning the decision tree, and implementing conformal prediction on it. In addition, we will also implement randomized conformal prediction and compare its results with standard conformal prediction. For each transfer, we will test the models on ten different test data sets by increasing the level of transfer step by step. We will make modifications to the conformal prediction by changing the conformity measure in order to maintain the property of validity and efficiency. This paper include multiple tables and graphs showing performances of all the models. We will also discuss the applications and future works.

Contents

1 Introduction

Transfer learning is a method that allows us to train a model on one set of data and reuse it on different sets of data when resources are limited or training multiple models is expensive. Human learners naturally transfer their knowledge from one activity to another. In other words, when faced with new challenges, we recognise and use the pertinent information from prior learning experiences. We learn new tasks more quickly the more closely they resemble our prior knowledge. Transfer learning seeks to advance classical machine learning by transferring the knowledge acquired in one or more source tasks and applying it to the learning of a related target task. The development of methods for knowledge transfer is a step toward making machine learning as effective as human learning[13].

Evaluation of a prediction's reliability is becoming increasingly important with the expanding use of prediction models across all industries. Conformal Predictions provides us with the level of certainty for each prediction. It measures the score of certainty of a sample with the distribution of the data from training set. Any point prediction technique for classification or regression, such as support-vector machines, decision trees, boosting, neural networks, and Bayesian prediction, can be combined with conformal prediction. Beginning with the point prediction approach, we create a conformity measure that gauges how out of the ordinary an sample appears in comparison to earlier data. The conformal prediction algorithm then converts this nonconformity measure into prediction areas[11].

A good conformal prediction model should always maintain its properties of validity and efficiency in all cases. But some models fail to maintain it while they are used for transfer learning. They tend to make poor performances when the difference between the training and testing sets increases. They start making mistakes like making accurate predictions with low confidence or false predictions with high confidence. But still, we can modify our models in such a way that their performance does not drop below a certain point.

2 Background Research

2.1 Assumptions

The sample data must include multiple elements, each of which must be measurable. Two main assumptions will be made regarding the sample points. Independent and identically distributed is the abbreviation for IID. The IID assumption states that all n+1 samples, including the test sample, are independently created using the same probability distribution. The second is the exchangeability assumption, which states that the test sample and the sequence

$$z_1, z_2, z_3, ..., z_n$$

are produced using the same exchangeable probability distribution P. For any permutation of the set 1, 2,..., n+1 the distribution of the original samples and the permuted samples are the same. A sequence is exchangeable if it has an IID [1].

2.2 Classification model

There are several models available for classification problem. In [8], KNN is implemented with conformal prediction and tested it against different transfers. Every model has it's advantage and disadvantage. Out of Support Vector Machines (SVM), Agglomerative Hierarchical clustering (AHC) and Decision Tree (DT), we decided to implement decision tree. The preparation of data for decision trees does not require the consumers of the data to exert superhuman efforts, in contrast to some methods that demand considerable data preparation before performing analysis and applying the algorithms. The data must be adjusted to scale with the model in order to fit a regression model or compute the coefficients. The decision trees don't need to undergo these changes either, as their structure stays the same throughout the study. The classification decision tree algorithm has several features such as pruning, unbiased splits, branches/splits, split type, user-specified priors, variable ranking, user-specified costs, missing values, bagging, and ensembles[9].

2.3 Data

Future performances of the induced model are significantly influenced by the relevance and quantity of examples in this training sample. A little manipulation in data directly impacts in the model's structure and performance. Using appropriate data for all cases of training and testing was a big task. It is quite challenging to find real life data with all the required variations. So we decided to generate artificial data with user-defined parameters.

3 Conformal Prediction

3.1 Concept

Suppose we have trained a prediction model, and the end user wants to know how confident we are in each prediction. The conformal prediction enters the picture at this point. We develop a model that computes the prediction's confidence level for each prediction. Each sample has a score, which is calculated using a method called as conformity measure. This score tells us how conforming the sample is to the distribution of training data, which is called the p-value of that sample. The performance of conformal prediction can be evaluated using its properties of validity and efficiency. We assess how conforming the sample is in relation to the data distribution. This is called the conformity score of that sample. We can select the suitable measure to calculate this conformity. We determine the sample's prediction's level of confidence, or p-value, based on this conformity score. P-values are in the 0–1 range. The reliability of the prediction is indicated by the p-value.

3.2 Algorithm

Assume that we have already trained a prediction model, and we are implementing conformal prediction.

- Compute conformity scores for all the samples in training data set. Store the list of this scores in ascending order.
- Take one test sample and assign all possible labels to it, then compute its conformity score for each assignment.
- Compare the chosen score with the ascendingly sorted list of scores from training samples and get its index number. This index number is the rank of the current sample. If there are repeated scores in the list, select the highest index of that score.
- Divide the rank by the number of samples in the training data set plus one. This will give us the p-value of the prediction.
- Choose the label that has the highest p-value as our prediction label for the current sample. That is our prediction label and it's confidence level for current sample.

$$p(y) = \frac{\#\{i = 1, ..., n+1 | \alpha_i^y <= \alpha_{n+1}^y\}}{n+1}$$

- Repeat the same process from step 2 for all the test samples.

3.3 Properties

We will see three properties. Accuracy is the basic property of a model. The two desiderata for inductive conformal predictors are their validity and efficiency. Validity requires that the coverage probability of the prediction sets should be at least equal to a preset confidence level, and efficiency requires that the prediction sets should be as small as possible[14].

3.3.1 Accuracy

Accuracy is one metric for evaluating classification models. The percentage of predictions that our model correctly predicts is known as accuracy[7].

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

3.3.2 Validity

To compute the validity, we calculate the average value of the p-value of the accurate prediction made by our model. In this research we tried to maintain this property near or above 0.5.

3.3.3 Efficiency

Unlike validity, this property depends on how confidently the model makes false predictions. We want it to be as minimal as possible. We calculate the average value of the p-values of false predictions made by our model. Good efficiency without good validity is not worth anything.

4 Data

The data we are using in this research has two attributes (X1 and X2) and a label (Y). X1 and X2 are continuous random variables, and the label Y is a discrete random variable. This data forms two clusters focused around two points. Cluster 1 is classified as -1, and cluster 2 is classified as +1. Both clusters are focused around different coordinates with different noise levels.

4.1 Generating Data

- We have defined a function *genData(x1,x2,s,label,n)* is used to generate artificial data.
- This function takes 5 parameters.
  - x1 : 1st co-ordinate on 2 dimensional plane.
  - x2 : 2nd co-ordinate on 2 dimensional plane.
  - s : Noise level of data.
  - n : Number of data points.
  - label : Class of the data.
- We import two Python libraries into *Numpy* and Pandas to use some predefined functions from them.
- We declare two empty lists before starting *for* loop. *for* loop iterates for *n* times which is given in parameter.
- Each iteration generates two random numbers in the normally distributed range of -1 to 1 and multiplies them with the noise *s*. Then these numbers are added to x1 and x2.
  - X1.append(x1 + s*np.random.normal(-1,1))
  - X2.append(x2 + s*np.random.normal(-1,1))
- This loop creates *n* coordinates focused around *x1* and *x2* with noise *s*.
- After the loop terminates, all the data is stored in a data structure called *DataFrame* which is available in *Pandas* library.
- Each function call generates data for one specified class. We need to call this function multiple times because one function call generates the data only for one class.

5 Decision Tree

5.1 Concept

Human decision-making is similar to that of a decision tree. Procedure and to make it simple to comprehend. Whether one has discrete or continuous data as input, it can solve in both cases. A decision tree is a tree-based machine learning model that can be used for both regression and classification problems. We are using this model for a classification problem. We use recursive binary splitting to build the decision tree[6]. The response variable is divided into primarily two kinds by the classification tree. Yes or No can also be expressed mathematically as 1 or 0. Each node splits the data into two sets, and these are further divided until the tree reaches its maximum depth or the data is split into the purest subsets.The data is split into multiple blocks recursively and the prediction model is fit on each of such partition of the prediction model. Now, each partition represents the data as a graphical decision tree[9]. We can use two methods to compute the impurity of the split.

- Gini Index:
$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}).[6]$$
- Entropy:
$$D = -\sum_{k=1}^{K} \hat{p}_{mk} log p_{mk}.[6]$$

Based on the value of one of the data set's fields or columns, each rule assigns a record or observation to a node in a branch or segment. Inputs are the fields or columns that are used to build the rule. Following the sequential application of splitting rules, a hierarchy of branches within branches is created, giving rise to the distinctive inverted decision tree structure. A decision tree is a layered hierarchy of branches, and each segment or branch is referred to as a node.A branch or additional segment of a node is formed by the node and all of its descendent segments. The leaves are the decision tree's bottom nodes (or terminal nodes). The decision rule offers a special way for data to enter the class that is designated as the leaf for each leaf. All nodes, including the bottom leaf nodes, have assignment rules that are mutually exclusive. As a result, just one node has records or observations from the original data collection. After the decision rules have been established, it is possible to forecast new node values using the rules in light of fresh or previously unobserved data. The decision rule produces the projected value in predictive modelling[3].

- Decision Node: This node contains a decision, which is used to split the data into its child nodes.

- Leaf Node: Data is not divided further once it reaches the leaf node. This node is used for assigning labels to the samples. Leaf nodes vote for all the possible labels; the label with the most votes is chosen as the prediction label for that leaf node.

A decision tree is like a flow chart with if-else statements. But the important part is to find the best conditions for those if-else statements, which is our task during training the model.

Nodes can be limited to a minimum size, and each category in the target classification can be limited to a specific size. For instance, a restriction might provide that no nodes with fewer than 50 observations or nodes with fewer than five observations for a category value will be found. The depth at which decision trees can grow can be limited. For instance, it is possible to confine decision trees to cease growing at three layers. The root node is used to calculate levels. [3]

But before starting the training, we need to set the parameters to tune the model according to the data for the best results. Below are the parameters we are setting-

- Maximum depth of the tree: Level 3
- Method to calculate impurity: Entropy
- Generating splits: From the available dataset of the current node, we take the minimum and maximum value of X and divide the range with an interval of 0.02.

5.2 Training the model

- The first step is to split the available data into two parts for training and testing purposes. Here, we are making a 60-40 split.
- We start from the root node, which contains all test data with maximum impurity.
- From the available data in the current node, make all possible splits.
- For each split, calculate its information gain. Information gain is calculated as follows:

$$IG = D(Parent) - \sum W_i D(Child).$$

  - D(Parent): Entropy of parent node.

  - D(Child) : Entropy of child node.
  - W : Weight of the node, that is length of child node divided by length of parent node.

- Out of all the splits, choose the split which has maximum information gain. That is the splitting condition of the current decision node.

- Repeat the process from step 3 for all child nodes. Do not split a node if it has no impurities.

- Stop when all the leaf nodes reach maximum depth or have no impurities.

- All the not-split nodes are leaf nodes. Each node votes for all the labels. A label with the most votes is the prediction label for that node.

$$\hat{y} = argmax \sum 1[y_n = i]_{i=0}^{m}.[6]$$

5.3 Testing the model

We have to check the performance of our model after training it. For that, we use the testing data set. We predict the label for each sample, and compare it with its true label. As we have seen above,

$$Accuracy = \frac{Number\ Of\ Accurate\ Predictions}{Number\ Of\ Total\ Predictions}$$

5.4 Advantages

- Decision trees are effective in both classification and regression applications because they may be used to predict both continuous and discrete values[5].

- When compared to KNN and other classification algorithms, they are incredibly quick and effective[5].

- One of the quickest ways to determine the most important factors and relationships between two or more variables is to use a decision tree. We can add new variables or features to the result variable more effectively using decision trees.[5]

5.5 Disadvantages

- From computational perspective, the time complexity for performing this operation is extremely high and keeps rising as the number of records rises. Training a decision tree with numerical variables can take a long time[5].
- Give the training-time complexity more time to rise as the input does[5].
- Small alterations in the data in a decision tree may result in the generation of a complicated new tree. In the decision tree, this is referred to as variance, and it can be reduced using techniques like bagging and boosting[5].

6 Implementing Conformal Prediction using Decision Tree

Before implementing the conformal prediction, we have to choose the right conformity measure for the model.

6.1 Conformity Measure

6.1.1 Vote of the node

As we have seen above, each leaf node votes for all labels. The value of the vote ranges between 0 to 1. The vote value is our conformity score. This measure causes repetitions in conformity scores as the number of nodes is limited, so the number of votes are also limited.

$$Vote\ of\ y = \frac{Number\ of\ available\ samples\ with\ label\ y}{Total\ number\ of\ available\ samples}$$

6.1.2 Distance from the centroid

To use this measure, we need to calculate the centroids of all the clusters. Each cluster is a class.

$$Conformity\ Score = \frac{Distance\ to\ the\ centroid\ of\ different\ class}{Distance\ to\ the\ centroid\ of\ same\ class}$$

There are multiple methods available to calculate the distance. Here we are using the Euclidean method. Suppose we have 2 points with co-ordinates (x1,y1) and (x2,y2), and we calculate the straight distance between them using:

$$Distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

6.2 Function

In the code, we have written separate functions to implement conformity prediction. The function called *std_CP(X_train,X_test,Y_test,model,CM)* is used for implementing standard conformal prediction. It takes four arguments.

- X_train: X values from training data.
- Y_train: Y values from training data.
- model: The model we have trained using X_train and Y_train.
- CM: This parameter takes only the integers 1 or 2 to specify which of the two conformity measures we want to use.

6.3 Algorithm

- Declare three empty lists to store predicted labels, p-values for accurate predictions, and p-values for false predictions.
- Using the received parameter *CM*, pass the X_train, X_test data and model to an appropriate function.
- The above function call will return a list of p-values for each element of X_test. Each element of the list contains two p-values. The first p-value is for the confidence level for class 1, and the second is for class -1.
- Start a loop to iterate from first to last element of the p-values and X_train list.
- For the current iteration, if the p-value of class 1 is greater than the p-value of class 2, assign class 1 to the current sample from X_test, otherwise assign class -1.
- Compare the assigned class of current sample with Y_test. If the assignment is correct, add the p-value for that class to the list of accurate p-values. Otherwise, add it to the list of false p-values.
- Repeat the same process as in the fifth step for all test samples.

6.4 Time Complexity

6.5 Randomized Version of Conformal Prediction

The IID assumption, also known as the randomness assumption, states that each observation in a sequence is produced randomly from the same probability distribution on the space of potential observations.

$$p(y, \tau) = \frac{\#\{i : \alpha_i^y < \alpha_{n+1}^y\} + \#\{i : \alpha_i^y = \alpha_{n+1}^y\}}{n+1}$$

Maintaining the validity of the model in a randomized version of conformal prediction is challenging.

7 Transfer Learning

According to H. Judd, experience is generalised as a result of learning to transfer. As long as a person generalises his experience, the transfer from one scenario to another can be realised. This idea states that a relationship between two learning activities is necessary for transfer to occur. Since the violin and the piano are both musical instruments and maybe share some common knowledge, someone who has studied the violin can learn the piano more quickly than others in practise.Transfer learning, which was inspired by humans' capacity to transfer knowledge across domains, tries to use knowledge from a related domain (known as the source domain) to enhance learning effectiveness or reduce the number of labelled instances needed in the target domain. It is important to note that applied knowledge does not necessarily have a beneficial effect on new activities. Knowledge transfer may not succeed if the domains have few characteristics. For instance, learning to ride a bike won't make us better piano players. Additionally, because some similarities between domains may be deceptive, learning is not always facilitated by them. For instance, despite the close relationship between Spanish and French and the fact that both languages are Romantic. Persons who learn Spanish may have trouble learning French because they use the incorrect vocabulary or conjugation. This happens because learning the word formation, usage, pronunciation, conjugation, and other aspects of the French language might be hindered by prior successful experience in Spanish[16].

In this paper, we are going to apply our model to three types of transfers. We will train the model one training set, and test it on similar but different data set. We will gradually increase the level of difference.

7.1 Increasing the noise in data

We train the model on the data with a smaller noise level and test it by gradually increasing the noise level of the test data while keeping the cluster's focused coordinates constant.

- Data split 60-40
- Training Data: Positive data points (y=1) clustered around x1 = 2, x2=3 with noise = 1 Negative data points (y=-1) clustered around x1 = 0, x2=0 with noise = 2 Number of data points:
- Testing Data: Positive data points (y=1) clustered around x1 = 2, x2=3 with noise = 1 to 10 Negative data points (y=-1) clustered around x1 = 0, x2=0 with noise = 2 to 11 Number of data points:

7.2 Decreasing the noise in data

We train the model on the data with a higher noise level, test it by gradually decreasing the noise level of the test data, and maintain the cluster's targeted coordinates.

- Data split 60-40
- Training Data: Positive data points (y=1) clustered around x1 = 2, x2=3 with noise = 10 Negative data points (y=-1) clustered around x1 = 0, x2=0 with noise = 11 Number of data points:
- Testing Data: Positive data points (y=1) clustered around x1 = 2, x2=3 with noise = 10 to 1 Negative data points (y=-1) clustered around x1 = 0, x2=0 with noise = 11 to 2 Number of data points:

7.3 Shift in the model

We train the model on the data with fixed target coordinates for the clusters. and gradually shift the target coordinates of the clusters of test data sets.

We train the model on the data with a higher noise level, test it by gradually decreasing the noise level of the test data, and maintain the cluster's targeted coordinates.

- Data split 60-40
- Training Data: Positive data points (y=1) clustered around x1 = 2, x2=3 with noise = 1 Negative data points (y=-1) clustered around x1 = 0, x2=0 with noise = 2 Number of data points:

- Testing Data: Positive data points (y=1) clustered around x1 = 2 to 11, x2 = 3 to 12 with noise = 1 Negative data points (y=-1) clustered around x1 = 0 to 9, x2 = 0 to 9 with noise = 2 Number of data points:

8 Observations

We observed the results for both versions of conformal prediction with both conformity measures. We compared four models across all three transfers. We compared the scores up to six decimal points for higher precision. Each table contains four columns. First is the level of transfer; second is validity; third is efficiency; and fourth is accuracy. We will gradually increase the transfer level by one level ten times.

8.1 Increasing the noise

- Standard conformal prediction with the conformity measure as the vote of the node. (Table 1; Fig.1,2,3)

Noise Level	Validity	Efficiency	Accuracy
0	0.832492	0.738542	0.93125
1	0.876752	0.897337	0.71875
2	0.938808	0.978212	0.5875
3	0.975488	0.952938	0.55
4	1	0.919051	0.5125
5	0.982881	0.958786	0.525
6	0.991123	0.987656	0.50625
7	0.974918	1	0.5375
8	1	0.991123	0.49375
9	0.991012	1	0.5

Table 1:

- Randomized conformal prediction with the conformity measure as vote of the node. (Table 2; Fig.1,2,3)
- Standard conformal prediction with the conformity measure as distance from the centroid. (Table 3; Fig.1,2,3)
- Randomized conformal prediction with the conformity measure as distance from the centroid. (Table 4; Fig.1,2,3)

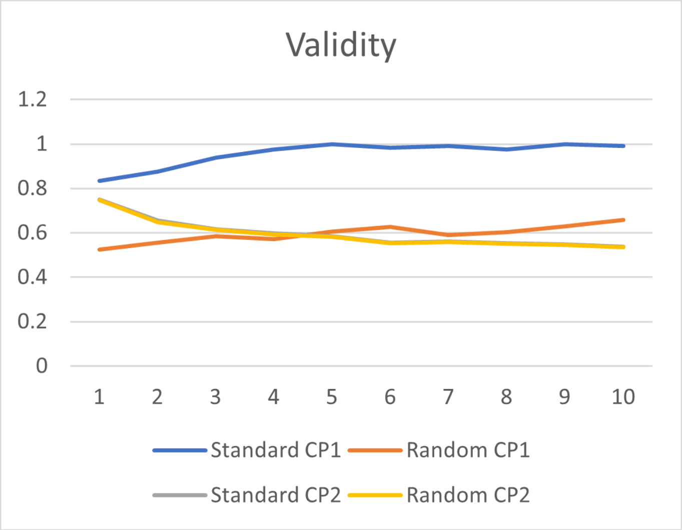

Figure 1:

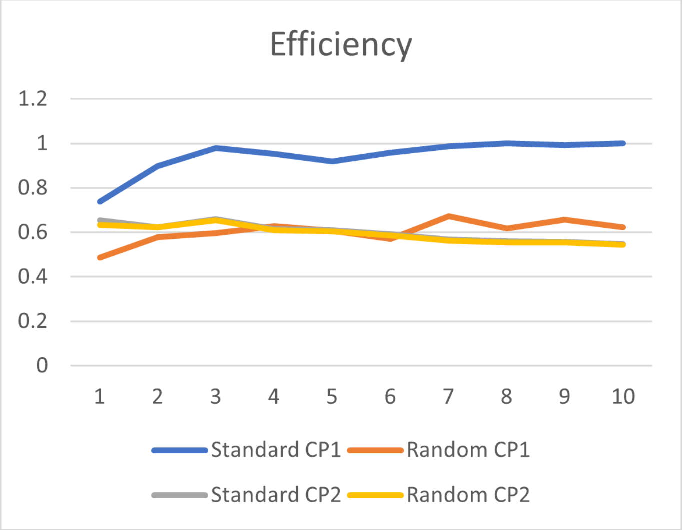

Figure 2:

Noise Level	Validity	Efficiency	Accuracy
0	0.524988	0.485349	0.93125
1	0.555839	0.577502	0.71875
2	0.584887	0.595229	0.5875
3	0.571455	0.627226	0.55625
4	0.60611	0.606459	0.525
5	0.627165	0.569976	0.525
6	0.589532	0.672089	0.50625
7	0.603162	0.616373	0.5375
8	0.629773	0.65779	0.49375
9	0.658006	0.621333	0.5

Table 2:

Noise Level	Validity	Efficiency	Accuracy
0	0.749946	0.652893	0.95
1	0.654184	0.62345	0.8
2	0.617028	0.658708	0.6625
3	0.596812	0.614864	0.56875
4	0.585564	0.610112	0.575
5	0.556783	0.590502	0.55625
6	0.561556	0.566795	0.54375
7	0.552651	0.560296	0.55625
8	0.548173	0.556966	0.475
9	0.538634	0.547869	0.51875

Table 3:

As shown in figure 1, the validity of randomized conformal prediction with the first conformity measure and standard conformal prediction with the second conformity measure is nearly identical. The validity of randomized conformal prediction with a second measure decreases slightly. However, with increasing noise, the validity of standard conformal prediction with node vote as measure approaches 1. Even though the validity is good, efficiency of the standard conformal prediction with vote of the node as measure approximates with 1, which is not a good case. It means that the model predicts both true and false prediction with high confidence. On the other side, all the three remaining models maintain the efficiency around

Noise Level	Validity	Efficiency	Accuracy
0	0.747222	0.633609	0.94375
1	0.648376	0.622634	0.80625
2	0.612701	0.654193	0.6625
3	0.592998	0.610732	0.56875
4	0.581782	0.604504	0.56875
5	0.55214	0.585904	0.55625
6	0.557659	0.562095	0.5375
7	0.550067	0.554277	0.5375
8	0.544006	0.553371	0.48125
9	0.534721	0.544077	0.5125

Table 4:

0.6. All the four models have similar trends for accuracy. Accuracy false from approx 0.94 with increasing noise and flattens to approx 0.5 after 4 levels of noise.

8.2 Decreasing noise

- Standard conformal prediction with the conformity measure as vote of the node. (Table 5)

Noise Level	Validity	Efficiency	Accuracy
9	0.71958	0.65978	0.5875
8	0.683884	0.634233	0.6
7	0.691585	0.669651	0.55
6	0.696387	0.691198	0.60625
5	0.693736	0.703051	0.59375
4	0.708659	0.728488	0.68125
3	0.721985	0.832645	0.7375
2	0.735798	0.866757	0.69375
1	0.762286	0.876095	0.58125
0	0.794409	0.853907	0.50625

Table 5:

- Randomized conformal prediction with the conformity measure as vote

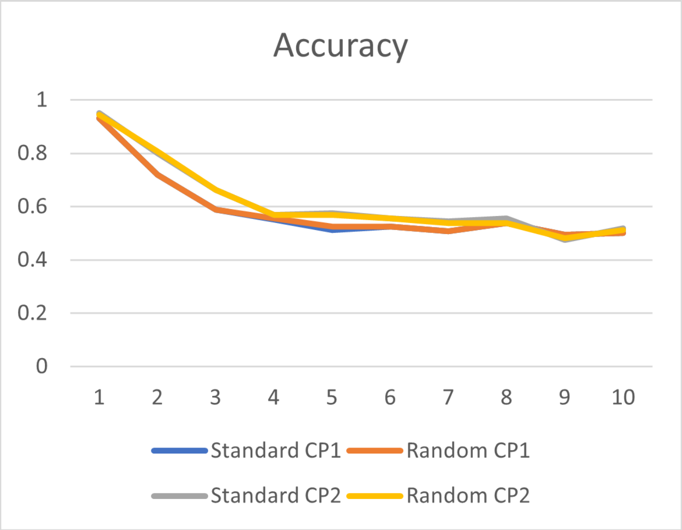

Figure 3:

of the node. (Table 6)

Noise Level	Validity	Efficiency	Accuracy
9	0.561632	0.487666	0.5875
8	0.47426	0.406121	0.6
7	0.514867	0.449055	0.54375
6	0.482923	0.479006	0.6125
5	0.506284	0.512526	0.6
4	0.513913	0.544158	0.68125
3	0.55491	0.702086	0.7375
2	0.567791	0.755608	0.69375
1	0.616724	0.773344	0.58125
0	0.654984	0.736217	0.50625

Table 6:

- Standard conformal prediction with the conformity measure as distance from the centroid. (Table 7)
- Randomized conformal prediction with the conformity measure as dis-

Noise Level	Validity	Efficiency	Accuracy
9	0.754537	0.758363	0.60625
8	0.785806	0.753589	0.64375
7	0.804785	0.757788	0.59375
6	0.820083	0.773278	0.625
5	0.816136	0.828549	0.64375
4	0.849862	0.834517	0.6
3	0.864284	0.870828	0.51875
2	0.856443	0.904279	0.50625
1	0.836209	0.906767	0.5
0	0.818182	0.89499	0.5

Table 7:

tance from the centroid. (Table 8)

Noise Level	Validity	Efficiency	Accuracy
9	0.75254	0.75071	0.6
8	0.781754	0.749529	0.64375
7	0.797435	0.757425	0.6
6	0.816529	0.769215	0.625
5	0.811683	0.823184	0.64375
4	0.84616	0.830385	0.6
3	0.860799	0.866642	0.51875
2	0.851903	0.900617	0.50625
1	0.832438	0.902738	0.5
0	0.813533	0.890134	0.5

Table 8:

Both versions of conformal prediction models with conformity measure as distance from the centroid are following same pattern for validity. It is maintained around 0.8. Standard version of conformal prediction with first measure has almost linear and slightly increasing trend. Randomized version of conformal prediction with second measure has the lowest but increasing validity which fluctuates a little bit initially. Overall, all models have successfully managed to keep the validity above 0.5. Same as validity, models with conformity measure as distance from the centroid are equivalent

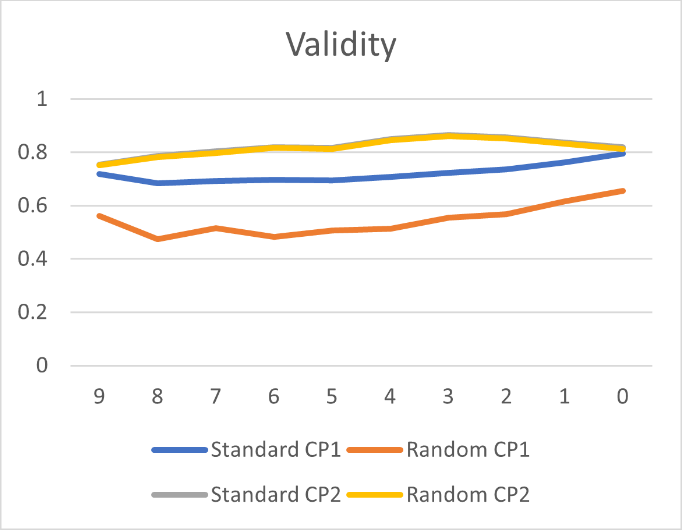

Figure 4:

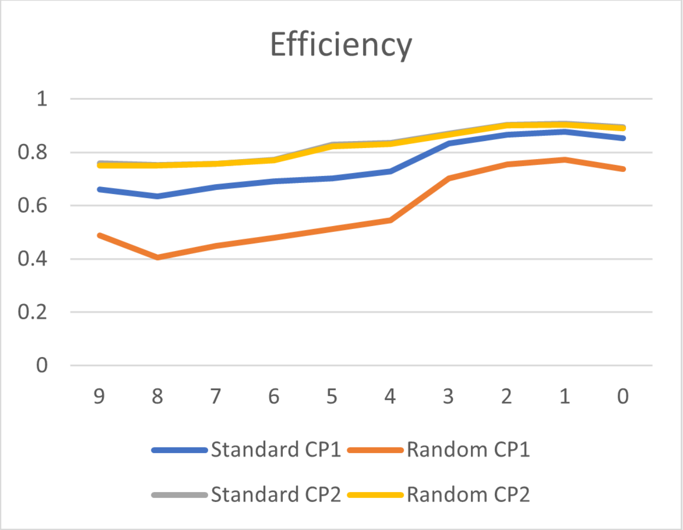

Figure 5:

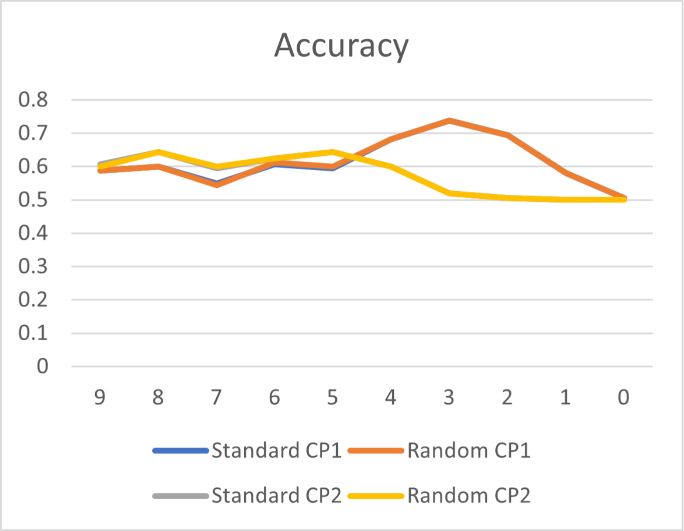

Figure 6:

and moving from 0.75 to 0.82. Models with first measure follows similar pattern but the random version managed to minimize the efficiency overall. All models are behaving similar with accuracy till noise is reduced till level 5. As the noise level drops below 5, models with first measure are showing bulge, and models with second measure flattening to 0.5. All the models meet the accuracy 0.5 at noise level reduced to 0.

8.3 Shifting models

- Standard conformal prediction with the conformity measure as vote of the node. (Table 9)
- Randomized conformal prediction with the conformity measure as vote of the node. (Table 10)
- Standard conformal prediction with the conformity measure as distance from the centroid. (Table 11)
- Randomized conformal prediction with the conformity measure as distance from the centroid. (Table 12)

All models has negative slope with shifting the co-ordinates. Standard version of conformal prediction with first measure is showing highest validity.

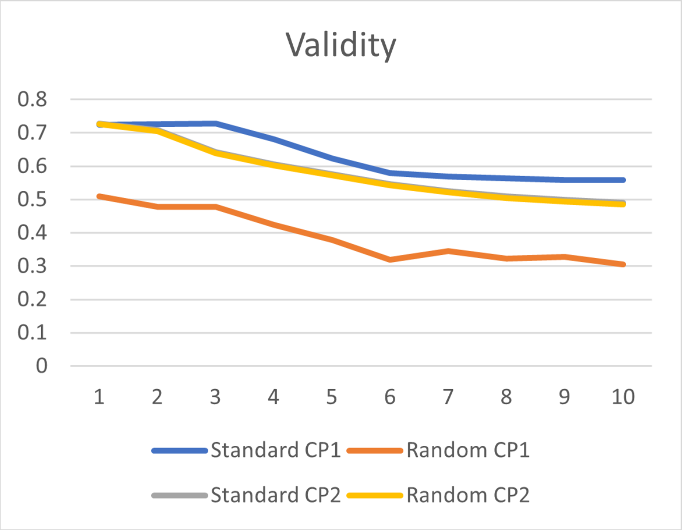

Figure 7:

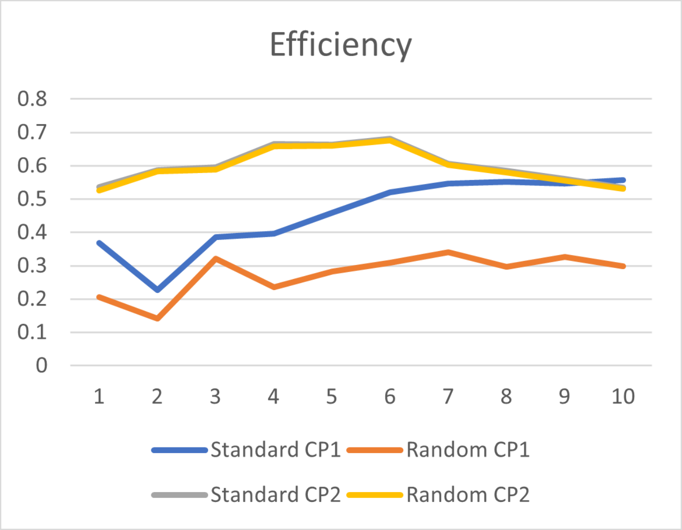

Figure 8:

Shift	Validity	Efficiency	Accuracy
0	0.723554	0.368595	0.9375
1	0.725799	0.226473	0.80625
2	0.727908	0.386639	0.8125
3	0.681334	0.395851	0.69375
4	0.623703	0.458803	0.5875
5	0.578906	0.520987	0.525
6	0.568635	0.545878	0.5125
7	0.56331	0.551941	0.50625
8	0.557851	0.546178	0.5
9	0.557851	0.557851	0.5

Table 9:

Shift	Validity	Efficiency	Accuracy
0	0.509493	0.205234	0.925
1	0.478582	0.142063	0.81875
2	0.478189	0.321243	0.83125
3	0.424329	0.23614	0.7
4	0.379132	0.2828	0.6
5	0.319105	0.309532	0.53125
6	0.344487	0.340962	0.5125
7	0.322059	0.296893	0.50625
8	0.328079	0.32608	0.50625
9	0.304545	0.298967	0.5

Table 10:

Both versions of conformal prediction with second measure are equivalent to each other but validity drops little more after third shift. The randomized version with first measure shows lowest validity starting to fall from 0.5 and flattens to approximately 0.3 after sixth shift. This is the lowest performance of among all the models in every transfer. Even though it is difficult to maintain validity in randomized versions of conformal prediction, randomized versions of conformal prediction with second measure has performed as good as it's standard version and successfully maintained the validity above 0.5. Same as validity, both versions of models with second measure has same efficiency starts from 0.53, increases up to 0.68 and again

Shift	Validity	Efficiency	Accuracy
0	0.728017	0.53595	0.9375
1	0.708904	0.586238	0.85625
2	0.641452	0.596207	0.75625
3	0.605099	0.665978	0.6625
4	0.575789	0.663854	0.55
5	0.546849	0.680638	0.51875
6	0.525967	0.606758	0.50625
7	0.508781	0.584556	0.5
8	0.498244	0.559762	0.5
9	0.489514	0.534659	0.5

Table 11:

Shift	Validity	Efficiency	Accuracy
0	0.725803	0.52592	0.93125
1	0.704862	0.582824	0.85625
2	0.639256	0.588017	0.75
3	0.602125	0.658903	0.65625
4	0.571844	0.65955	0.55
5	0.542119	0.676613	0.51875
6	0.521579	0.602626	0.50625
7	0.504752	0.580527	0.5
8	0.494112	0.555527	0.5
9	0.485486	0.529959	0.5

Table 12:

drops to 0.53. Where as both versioned models with first measure has similar patterns but randomized version has optimal efficiency, which starts from 0.2 and gets nearly flat at 0.3 after fifth shift. But the performance of the standard conformal prediction with first measure keeps on increasing after second shift and hits 0.56 which is little more than the models with second measure. In this transfer, all models shows similar trend for accuracy. All scores approximately 0.93 in first shift and drops and flattens to 0.5 after fifth shift. Models with first measure fluctuates a bit between second and third shift, whereas models with second measure linearly drops.

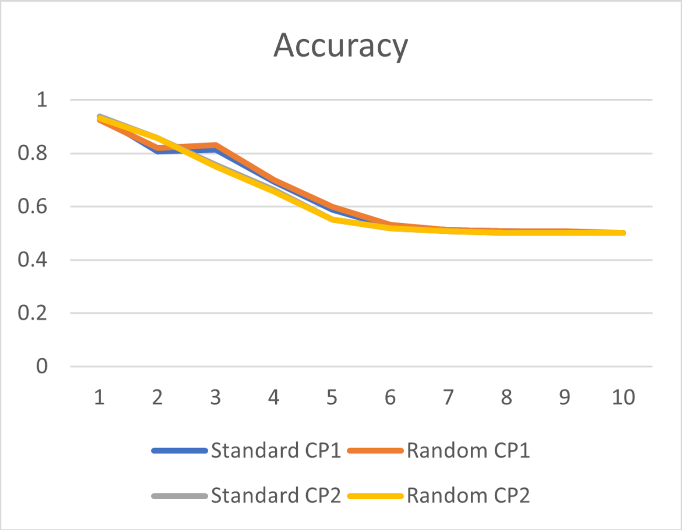

Figure 9:

9 Conclusion

Based on the graphs above, we can draw a few conclusions about the performance of each type of model on each type of transfer. All the models behave differently on different transfers. The randomized version of the conformal predictor with first measure of vote clearly produces lower validity scores than it's standard version but successfully minimizes efficiency. Whereas both versions of the conformal prediction with conformity measure as distance from the centroid does not have much difference in their performances. In the first transfer of increasing the noise level, randomized conformal prediction with first measure behaves similar to the models with second measure, all three are shows linear trend and approximates to 0.6. But in case of decreasing the noise level, standard version of first measure shows similar pattern as the models of second measure but provides less validity and efficiency. In same case, it's randomized version has lowest performance than others.

One interesting observation I found that in some cases as the level of transfer increase, accuracy decreases. But because of that decreasing accuracy, we see a rise in validity.

10 Applications

10.1 Conformal Prediction

10.1.1 Face Recognition

Uncontrolled circumstances that alter image quality, as well as the inadequate or misleading information associated with occlusion and/or disguise, present a problem for face identification[1]. Interoperability-related difficulties in uncontrolled environments include handling incomplete and incorrect information, image variability including stance, illumination, and expression, and temporal change. The conformal prediction framework helps recognition-by-parts and reidentification deal with uncontrolled environments. Interoperability is a challenge for mass screenings when enrollment and authentication happen at various places that are separated in space and time. As a result of training and tiredness, the performance of the various human operators differs. The state of operation of the various capture devices varies as well. Interoperability is impacted by proper protocol and result validation, including the cumulative impact of such diverse conditions that are difficult to estimate and predict[1].

10.1.2 Stock Market Prediction

Brokers would benefit from knowing what their market positions would be in the future in order to help meet the goals of making money and lowering risk. They would be greatly assisted in achieving this goal by predictive models that offer insight into potential future net position movements with a high degree of confidence[15].

10.1.3 Computational Drug Discovery

In the context of healthcare and personalised medicine, where accurately identifying, for example, which patients are likely to benefit from a particular drug treatment has strong ethical and legal implications, reliability estimates are crucial to increase the trust and application of artificial intelligence solutions to guide decision-making. In drug development, where both the prediction and the related uncertainty need to be taken into account for decision-making, estimating the dependability of predictive models is particularly important[2].

10.2 Transfer Learning

10.2.1 Visual Categorization

Due to environmental constraints and the high cost of human manual labelling in real-world applications, there may not always be enough training data with the same distribution or feature space as the testing data. In other instances, training samples are taken from a different angle when just one action template is offered for each action class. Regular machine learning algorithms are very likely to fail in such circumstances. This serves as a reminder of the power of the human visual system. Given the enormous geometry and intraclass variations of objects, the theory that humans develop such a competence through collected information and knowledge is supported by the fact that they are able to learn tens of thousands of visual categories in their lifetime[12].

10.2.2 Medical Imaging

The de facto approach for deep learning applications to medical imaging is transfer learning from natural image datasets, primarily ImageNet, using conventional big models and corresponding pretrained weights. However, there is limited knowledge on the effects of transfer because there are significant disparities between natural picture classification and the target medical tasks in terms of data volumes, features, and task specifications. Surprisingly, transfer delivers no boost to performance, and simple, lightweight models can perform comparably to ImageNet structures, according to a performance test on two large-scale medical imaging workloads. Investigating the learned representations and features reveals that rather than complex feature reuse, some of the deviations from transfer learning are caused by the over-parametrization of standard models[10].

11 Future Work

11.1 Multi-classed Data

The current data we have used in this research has only two classes, +1 and -1. This experiment can also be done with multi-classed data with some modifications to the model. Performance of the models with the measure of distance from the centroid might change significantly.

11.2 Regression Problem

As we produce a confidence for prediction in classification problem, we compute intervals with certain confidence for prediction in a regression problem. Finding the transferability of a conformal prediction model in a regression problem, is a topic of research.

11.3 Real World Data

The data we have used here is generated artificially with the parameters which was provided by us. It is quite challenging to find real life data with the level of differences of our choice. But this models can certainly be used on real data. For example, we can train the model for predicting the weather of London, and check it's performance if used for other cities.

11.4 Different Model

In [8], k-nearest neighbour is used; and in this paper, decision tree is used. Other models such as logistic regression, agglomerative hierarchical clustering, support vector machine can be tested against the same problem.

12 Professional Issues

12.1 Programming Language

There are numerous alternatives available for open-source programming languages like C, C++, Java, Python, etc. Every programming language has its own features. Python, R programming, and Matlab are very useful for building machine learning models and for statistical computations. Out of all these languages, I chose Python because of its functionality and my previous hands-on experience with it.

12.2 Integrated Development Environment (IDE)

programmesPython program can be developed on multiple open-source IDEs available on the internet, like PyCharm, Visual Studio, Atom, Spyder, and Jupyter Notebook. I decided to go with Jupyter Notebook because of its feature of executing the code in sections, where you can execute only a certain part of the code without running the entire programme.

12.3 Resources

Not all resources are freely available online. Most of the articles are freely available, but several important research papers and books are restricted. Many journals, however, grant access to their papers after verifying university student status. I accessed the books through the library provided by the university.

12.4 Citation

Citing all the content that is referred to and used is a very important responsibility to maintain ethics and integrity. Citing research papers, books is unalike citing online available content such as videos, websites, etc.

13 Self Assessment

A process with a sensible plan and disciplined execution produces the desired output. Before selecting the topic, I went through all the prerequisites required to complete this project. Though some of the main concepts used were not part of my course modules, I verified the availability of the resources before opting for the subject.

My supervisor provided me with the thesis, which has already worked on the same problem but with a different approach. It helped me understand the flow and scope of my project. Initially, we had in-person meetings and frequent communication over email to get clarity and establish a plan. As the process progressed, many hidden issues were resolved and decisions were taken. After every task, my professor suggested the changes, made corrections, and provided feedback. We agreed to move on to the next task if the results were satisfactory.

We started by choosing decision tree as our prediction model. Python has some libraries that provide in-built functions for training a decision tree, which could build the model with just one function call. But as I had to implement it from scratch because there were many modifications required in the model. Some of the code like what classes and functions to be used, I referred from [4]. I had a general sense of what could be the conformity measures for this model. The next relatively simple but very important task was to decide on different cases of transfer and generate the data for them. We produced three cases of transfer, and each transfer has ten levels.

We tested the transferability of the decision tree without conformal prediction in these cases and saw the clear impact of transfer on the model's performance. Implementing conformal prediction on this model was the next big task, but there were many visible, hidden, small, and complex tasks in between. We agreed on the method for measuring the conformity of the data, and I implemented it. Following the successful implementation of the standard version of conformal prediction, we decided to see the performance of the randomised version of the model.

Randomising the model was a bit tricky for me. After a few corrections, I developed the precise model and verified it with the supervisor. We found that the performance of this model drops and does not maintain its properties in some cases of transfer. We had to find appropriate conformity measures that could hold the properties. We chose a measure after some study and tested it against the same cases. So after this, we had a total of four different types of models, each tested against three different types of transfers. I produced the results multiple times to ensure their persistence.

In the last stage of the coding part, I started by writing the report for it. I sent the flow and contents to the supervisor, which I planned to include in the report. Though I was familiar with Microsoft Word, I decided to write the report with Latex as it is a great tool for writing a paper, and it was a learning opportunity for me. I quickly adapted to the platform and utilised as many features as possible to improve the quality of the report.

14 How To Use My Project

The folder *MyProj* contains three files; *Conformal_Prediction.ipynb* which contains code for the project, *Report_100998004.pdf* which contains the report, and *results.xls* which contains the results of the model.

To use the code, please follow below steps:

- This code is written in Python version 3.9.13. Make sour your system has same or higher version.
- Open the file *Conformal_Prediction.ipynb* in Jupyter Notebook.
- Execute the first cell which imports all the libraries required in the code.
- Program will throw an exception if any of the required library is not installed. Use below commands to install any missing library.
    - *pip install –user pandas* to install *pandas*
    - *pip install –user numpy* to install *NumPy*

  - *pip install –user sklearn* to install *sklearn*
  - *pip install –user random* to install *random*
  - *pip install –user math* to install *math*

- Run all the section from top to bottom in same order. The flow goes as importing libraries, then defining classes, then defining single functions, then code for executing classes and function.

- The second last section contains code with three function calls for three cases of transfer. Each function call runs all four models on that case of transfer. Initially all function calls are commented, uncomment the funciton call which you want to try.

- Last section contains code for four print statements, each assigned for one type of model. It prints the results of that model on the transfer case tested in previous section.

The .xls file contains numerical results and graphs. There are three sheets in the file for three types of transfer. Each sheet contains four tables for four types of models. Each table contains four columns, first column for noise level, second for validity, third for efficiency, fourth for accuracy. There are three graphs on each sheet used for visualizing three properties of conformal prediction. All these tables and graphs are also included in the report.

References

[1] Vineeth Balasubramanian, Shen-Shyang Ho, and Vladimir Vovk. *Conformal prediction for reliable machine learning: theory, adaptations and applications.* Newnes, 2014.

[2] Isidro Cortés-Ciriano and Andreas Bender. Concepts and applications of conformal prediction in computational drug discovery, 2019.

[3] P. SAS Institute. De Ville, Neville. *Decision trees for analytics: using SAS Enterprise Miner.* 2013.

[4] Sujan Dutta. Ml_from_scratch. `https://github.com/Suji04/ML_from_Scratch`, 2021.

[5] Educba.com. Decision tree advantages and disadvantages.

[6] Trevor Hastie Robert Tibshirani Gareth James, Daniela Witten. *An Introduction to Statistical Learning with Applications in R.* Springer, USA.

[7] Google. Or-tools, machine learning crash course.

[8] Sourabh Marne. Applicability of conformity scores for transfer learning with knn, 2021.

[9] Dr. GP Pulipaka. An essential guide to classification and regression trees in r language.

[10] Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. *Advances in neural information processing systems*, 32, 2019.

[11] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

[12] Ling Shao, Fan Zhu, and Xuelong Li. Transfer learning for visual categorization: A survey. *IEEE transactions on neural networks and learning systems*, 26(5):1019–1034, 2014.

[13] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010.

[14] Vladimir Vovk. Conditional validity of inductive conformal predictors. In *Asian conference on machine learning*, pages 475–490. PMLR, 2012.

[15] Wojciech Wisniewski, David Lindsay, and Sian Lindsay. Application of conformal prediction interval estimations to market makers' net positions. In Alexander Gammerman, Vladimir Vovk, Zhiyuan Luo, Evgueni Smirnov, and Giovanni Cherubin, editors, *Proceedings of the Ninth Symposium on Conformal and Probabilistic Prediction and Applications*, volume 128 of *Proceedings of Machine Learning Research*, pages 285–301. PMLR, 09–11 Sep 2020.

[16] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.