

Concept Drift Detection: Dealing with Missing Values via Fuzzy Distance Estimations

Anjin Liu, *Member, IEEE*, Jie Lu, *Fellow, IEEE*, and Guangquan Zhang

Abstract—In data streams, the data distribution of arriving observations at different time points may change – a phenomenon called concept drift. While detecting concept drift is a relatively mature area of study, solutions to the uncertainty introduced by observations with missing values have only been studied in isolation. No one has yet explored whether or how these solutions might impact drift detection performance. We, however, believe that data imputation methods may actually increase uncertainty in the data rather than reducing it. We also conjecture that imputation can introduce bias into the process of estimating distribution changes during drift detection, which can make it more difficult to train a learning model. Our idea is to focus on estimating the distance between observations rather than estimating the missing values, and to define membership functions that allocate observations to histogram bins according to the estimation errors. Our solution comprises a novel masked distance learning (MDL) algorithm to reduce the cumulative errors caused by iteratively estimating each missing value in an observation and a fuzzy-weighted frequency (FWF) method for identifying discrepancies in the data distribution. The concept drift detection algorithm proposed in this paper is a singular and unified algorithm that can handle missing values, but not an imputation algorithm combined with a concept drift detection algorithm. Experiments on both synthetic and real-world data sets demonstrate the advantages of this method and show its robustness in detecting drift in data with missing values. The results show that, compared to the best-performing algorithm that handles imputation and drift detection separately, MDL-FWF reduced the average drift detection difference from 10.75% to 5.83%. This is a nearly 46% improvement. These findings reveal that missing values exert a profound impact on concept drift detection, but using fuzzy set theory to model observations can produce more reliable results than imputation.

Index Terms—concept drift, machine learning, fuzzy distance, fuzzy clustering, fuzzy weighting, missing value

I. INTRODUCTION

In life, ideas constantly change and evolve. In data streams, these evolving ideas are reflected as changes in the data distribution of arriving observations. Areas where monitoring, control, and security are important are particularly dependent on fast and accurate data distribution change detection [1], [2] – for example, mobile tracking systems that monitor user behavior change, intrusion detection systems that look for unusual activity [3], [4]. The change of distribution in data streams is referred to concept drift in the literature [5], [6].

Drift detection systems typically infer changes in a situation by compiling sets of observations at different time points, estimating the discrepancies in the data distributions for each set, and comparing the results [1]. For machine learning models, the inevitability of concept drift means keeping up-to-date and adapting to the current state of ideas through incremental

training as new observations arrive [7], [8]. Without these continual self-refinements to the model, classification accuracy will eventually degrade beyond usefulness [9]–[12].

However, like learning with streaming data, a common problem in exploratory data analysis is handling missing values [13], [14]. Missing values in data are extremely common for dozens of reasons – human error, faulty sensors, power outages, etc. – and can cause serious implications for machine learning [15]. One of the most serious of these implications is the increased levels of uncertainty an incomplete data set brings [16]. This makes concept drift detection a much more complex task.

A common practice to mitigate the problem is to apply an imputation method to fill in the missing values with plausible substitutes [15]. As discussed in [17], the state-of-the-art imputation methods fall into two major categories: discriminative and generative. Discriminative methods include MICE [18], MissForest [19], and matrix completion [20]–[22]. Generative methods include algorithms based on either expectation maximization [23] or deep learning [17], [24], [25]. Regardless of the category, these algorithms were proposed to resolve missing values under different circumstances. These imputation algorithms have achieved remarkable success and can be easily combined with any learning algorithms. However, most are based on value-oriented imputation, which means they focus on minimizing the difference between the estimated values and the ground truths. Yet, the flexibility is a double-edged sword, they do not care about what the data is going to be used for. In other words, they do not take a task-oriented perspective.

Similarly, most drift detection researchers consider imputation to be an independent research task and so intentionally leave mechanisms for handling missing values out of their frameworks. We, however, believe that data imputation methods can have a significant impact on the accuracy and robustness of subsequent drift detection processes. An inappropriate value estimation will increase the bias of the statistics when comparing distribution, resulting in increased errors, such as shown in Fig. 1. Therefore, we propose a new concept, namely task-oriented imputation, that is, we directly minimize the difference between the ground truth of the statistics we need and the statistics acquired by imputation. In our solution, the statistics are the distances between observations.

The fundamental reason that data imputation impairs the performance of a drift detection method is because the replaced values decrease data variance (**Problem 1**). And, further, errors with the missing value estimations are not considered in the distribution estimations (**Problem 2**). For example,

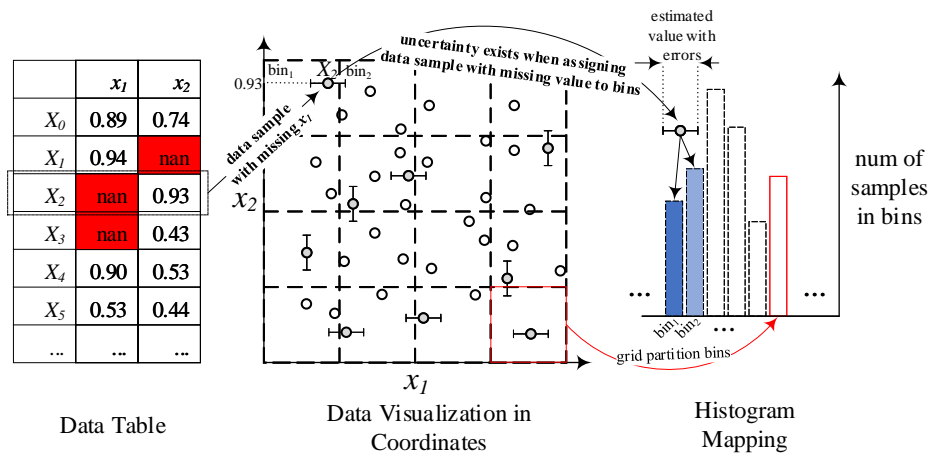


Fig. 1. A demonstration of uncertainty when allocating data observations with missing values to histogram bins. The grey points are the samples with missing values. I-o-I indicates the confidence interval of the imputation result. The *nan* stands for not a number. In this example, we want to use a histogram to estimate the distribution of the data set. Without considering the confidence interval, the top-left grey sample will be counted in bin₁. Assigning X_2 to bin₁ may introduce bias because there is a small chance that X_2 belongs to bin₂. Therefore, we suggest splitting this sample and counting it in both bin₁ and bin₂ where the fuzzy set theory is the best tool for this task.

using the mean of data as a replacement for the missing values forces the data toward a central focal point and reduces sample diversity [26], [27]. Small errors aggregate into large errors, introducing bias into the hypothesis testing for concept drift detection. Meanwhile, considering data imputation and concept drift detection as two independent tasks subjectively ignores the likelihood between the replaced value and the true value. For instance, assume we have a set of 1-dimensional observations $\{X_i\}_{i=1}^m$, where m is the size, and among these observations, there are z missing values $\{X_i^{miss}\}_{i=1}^z$. The imputation is the mean of the non-missing observations denoted as μ . Then the error that $\epsilon = \sum_{i=1}^z |X_i^{miss} - \mu|$ would not be considered in concept drift detection. Uncertainty in data and its impact on learning models has been well studied. For instance, in [28], [29], the authors evaluate the performance of classifiers in terms of their fuzziness and the problem's complexity, and discover that the generalization ability of a classifier is closely related to both its fuzziness and the target problem. Extrapolating on this conclusion, we suspect that the accuracy of a drift detection algorithm may also be closely related to data uncertainty and the fuzziness of the drift detection methodology.

To address Problem 1, we proposed a masked distance learning algorithm. The idea is to convert the distances calculated by different data imputation algorithms into a new feature set, and considering the new feature set and the true distance as a regression learning task, thereby, minimizing the difference between the estimated distance and the true distance, keeping the variance of data. To address Problem 2, the estimated distances are considered to be fuzzy to account for possible errors. This is coupled with a novel weighted frequency measuring method that tallies observations based on their degree of membership to histogram bins. Membership considers estimation errors in the data imputation process.

To detect drifts, we apply Pearson's chi-square test on the observed frequency as a hypothesis test [30], [31], and the significance level is governed by a parameter α , which

determines the sensitiveness of the hypothesis test to concept drift. In summary, the main contributions of this paper are:

- The notion of estimating the distance between samples rather than estimating the missing values, which reduces the cumulative errors caused by iterative estimations.
- A novel concept drift detection algorithm that is robust to missing values, along with a masked distance learning algorithm and a fuzzy-weighted frequency observing method – MLD-FWF when combined.
- A ablation study was conducted to evaluate the improvement of using fuzzy set theory and not using fuzzy set theory. The results indicate that applying fuzzy set theory for drift detection with missing values can achieve more reliable result.
- A comprehensive evaluation of the proposed algorithms and framework on both synthetic and real-world data sets with results revealing that applying fuzzy set theory to handle missing values is beneficial to reduce false alarms in concept drift detection.

The rest of this paper is organized as follows. In Section II, we discuss the problem of concept drift detection and data imputation, followed by the preliminaries relating to our drift detection method. Section III presents the proposed masked distance learning and the fuzzy-weighted frequency drift detection algorithms. Section IV evaluates the proposed distance learning performance and drift detection accuracy. Section V concludes this study with a discussion of future work.

II. LITERATURE REVIEW AND PRELIMINARIES

This section presents the problem of concept drift detection (Section II-A) and missing values (Section II-B). The preliminaries of our solution - Pearson's chi-square test is also introduced, Section II-C

A. Concept Drift Detection

In an evolving data stream, the distribution of available training samples may vary over time [1]. Consider a topological space feature space denoted as $\mathcal{X} \subseteq \mathbb{R}^n$, where n is the dimensionality of the feature space. A tuple (X, y) denotes a data instance, where $X \in \mathcal{X}$ is the feature vector, $y \in \{y_1, \dots, y_c\}$ is the class label, and c is the number of classes. A data stream, denoted as \mathcal{D} , can then be represented as a sequence of data instances. Time windows strategy chunks the sequence in a time interval as a batch. These batches are denoted as $D_{T_i} \in \mathcal{D}$, where T_i is a given time interval that defines the time window. A concept drift has occurred between two time windows T_i and T_{i+1} if the joint probability of X and y is different, that is, $p_{T_i}(X, y) \neq p_{T_{i+1}}(X, y)$ [5], [6].

There are two approaches to estimating the density discrepancy: parametric and nonparametric [31], [32]. The parametric approach assumes the data is drawn from a known distribution, with the major challenge being how to estimate the parameters to reach the maximum likelihood [31]. Nonparametric approaches assume that it is too difficult to devise an analytic expression of the target distribution's probability density function [33]. Therefore, the density is estimated empirically rather than by matching the data with any given parametric family. Given that concept drift detection inherently concerns evolving data distributions, it is difficult to rely on one distribution family for an accurate description. Therefore, most drift detection methods are nonparametric [1].

To quantify the distribution discrepancy between two multivariate sample sets, some researchers have proposed new test statistics, while others have proposed novel mapping methodologies that convert multivariate observations into univariate observations. For example, NN-DVI [34] and MMD [35] construct new statistical variables for comparing distributions to determine the differences. The performance of these methods is excellent, but that performance usually comes with a high computational cost. Alternatively, if the distribution of the proposed statistics is unclear, a Monto Carlo method can be applied to approximate the significance level. Examples include kdqTree combines with Kullback-Leibler divergence [36], competence model combined with total variation [7], QuantTree combined with total variation, or QuantTree combined with Chi-square statistics [37]. On yet another track, the solution involves building a model to map the discrepancy in distributions with powerful statistical tools, such as Pearson's Chi-square [31]. The general premise of these approaches is to convert multivariable data samples into a univariable two-sample test problem, such as an incremental Kolmogorov-Smirnov test [38] or a multi-Wald-Wolfowitz test [39]. However, no research has considered the missing values for concept drift detection.

B. Types of Missing Values and Handling Methodologies

Missing values are merely observations that were not recorded, perhaps due to human error or a sensor malfunction, which is very common in real-world data [15], [16], [40]. There are three main categories of missing values based on the correlation between the observations and the values that are

missing. These are: missing completely at random (MCAR), missing at random (MAR) and missing not at random (MNAR) [27], [41]. A recent survey by Santos et al. [15] provides a set of notations and definitions that demonstrate the difference between each category quite well, as follows:

Assume we have a data set with m number of instances and n number of features. The values of the data set are represented as an $m \times n$ matrix as $X = \{x_{i,j}\}_{i,j}^{m,n}$. Plus, a missing data indicator matrix is defined as an $m \times n$ zero-one matrix, denoted as M . M is represented as $M = \{I_{i,j}^{miss}\}_{i,j}^{m,n}$, where $I_{i,j}^{miss}$ is the missing value indicator and equals 1 if $x_{i,j}$ is missing, and 0 otherwise. The relationship describing the missing data is defined as a conditional probability function $p(M|X, \xi)$, where ξ are the parameters of the missing values [26]. In practice, ξ is not important, only the relation between M and the components of X are important, i.e., $X = (X^{complete}, X^{miss})$, where $X^{complete}$ is the observations without missing values, and X^{miss} is the the observations with missing values.

MCAR means that a missing value has nothing to do with its hypothetical value or with the values of other variables [15], [16]. The probability of missingness depends only on the parameters ξ , i.e., $p(M = 1|X, \xi) = p(M = 1|\xi)$ [15].

MAR means the missing values are not related to themselves, but they do depend on other features somehow. In other words, the conditional probability of missing values is $p(M = 1|X, \xi) = p(M = 1|X^{complete}, \xi)$ [5].

With MNAR, the missing values depend on both the observed and the unobserved information – $X^{complete}$ and X^{miss} – and the conditional probability of the missing values cannot be simplified, i.e., $p(M = 1|X, \xi) = p(M = 1|X^{complete}, X^{miss}, \xi)$ [15]. Conventional methods of handling missing values include complete case analysis, available case analysis, dummy variable adjustment, and imputation [26].

Imputation is currently the most widely used method for handling missing values [16]. Like dummy variable adjustment, missing values are replaced but this time with plausible values, not constants or a mean. The model is then trained just as though no values were missing [26]. However, there are two serious problems with most imputation methods. First, variances tend to be underestimated, which can lead to bias in the learning parameters [26]. Second, the standard error calculations presume that all data are real, which means the inherent uncertainty and sampling variability in the imputed values is not taken into account [26]. Of course, this results in confidence intervals that are too narrow.

C. Pearson's Chi-square Test

Pearson's Chi-square Test has been widely used to determine whether there are significant differences between expected frequencies and observed frequencies [30]. When there is no significant difference (i.e., when the null hypothesis is true), the statistic follows a chi-square distribution. Thus, the premise of the test is to assume the null hypothesis is true and then evaluate how likely a specific observation would be.

The standard process of the chi-square test is to use sample data to find: the degrees of freedom, the expected frequencies,

the test statistic, and the p-value associated with the test statistic [30]. Given a contingency table with i rows and j columns of categorical variables, the degrees of freedom are equal to

$$DF = (i - 1)(j - 1).$$

The expected frequency counts are computed separately for each level of one categorical variable at each level of the other categorical variable. The i -th and j -th expected frequencies of the contingency table are calculated with the equation

$$E_{i,j} = (n_i \times n_j)/n,$$

where n_i is the sum of the frequencies for all columns in row i , n_j is the sum of the frequencies for all rows of columns j , and n is the sum of all rows and columns. The test statistic is a chi-square random variable χ^2 defined by

$$\chi^2 = \sum \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}, \quad (1)$$

where $O_{i,j}$ is the observed frequency count at row i and column j , and $E_{i,j}$ is the expected frequency count at row i and column j . The p-value is the probability of observing a sample statistic as extreme as the test statistic. Since the p-value is a χ^2 test statistic, it can be computed with the chi-square probability distribution function.

Pearson's chi-square test should be used with a sufficiently large sample set. According to the central limit theorem, an χ^2 distribution is the sum of $O_{i,j}$ independent random variables with a finite mean and variance that converges to a normal distribution for large $O_{i,j}$. For practical purposes, Box et al. [30] claim that, for $O_{i,j} > 50$ and $E_{i,j} > 5$, the distribution of the estimated test statistics is sufficiently close to a normal distribution to ignore the difference.

III. CONCEPT DRIFT DETECTION WITH MISSING VALUES VIA FUZZY DISTANCE ESTIMATIONS

This section formally presents our proposed solution for concept drift with missing values. In Section III-A, we present a histogram-based concept drift detection method. Section III-B introduces our strategy for handling missing values. Last, Section III-C discusses how we leverage fuzzy distance estimations to address the uncertainty of missing values in drift detection.

A. Concept Drift Detection via Histogram Distribution Estimation

Problem Statement. Recall the drift detection problem outlined in Section II-A, i.e., Let \mathcal{A} and \mathcal{B} be random variables defined on a topological space $\mathcal{X} \subseteq \mathbb{R}^n$ at different periods in a data stream with respect to $p_{\mathcal{A}}, p_{\mathcal{B}} \in \mathcal{P}(\mathcal{X})$, where $\mathcal{P}(\mathcal{X})$ consists of all Borel probability measures on \mathcal{X} . Given two observation sets drawn from \mathcal{A} and \mathcal{B} , $A = \{X_i^A\}_{i=1}^{m_1}$ and $B = \{X_i^B\}_{i=1}^{m_2}$, the problem is: How confident are we to claim that $\mathcal{A} = \mathcal{B}$ based on A and B ?

To simplify this problem, it is commonly assumed that the observations A, B are i.i.d., which makes the research objective equivalent to a two-sample test problem. The general

procedure is to measure the difference between the observations A and B (sample sets), then infer the significance of the difference between \mathcal{A} and \mathcal{B} (population).

Remark: Recall the definition of concept drift, $p_{T_i}(X, y) \neq p_{T_{i+1}}(X, y)$, a data instance is defined as a tuple of (X, y) . For supervised learning setting, the target variable y can be considered as the $(n + 1)$ -th feature of the observations while for unsupervised setting, the drift detection will be equivalent to covariate shift detection [42].

The most intuitive method for capturing this difference is a histogram. A histogram is a set of intervals, i.e., bins, and density is estimated by counting the number of observations in each bin. A general framework for using a histogram and Pearson's chi-square test to detect concept drift is shown in Fig. 2.

The next question is how to partition the feature space as bins to build a histogram. In the literature, the partitions of most multivariate histograms are based on a grid or a tree structure [31], [37]. We have used clusters because, given our problem, this method makes it easy to determine which bin an observation belongs to. More specifically, with grid- and tree-based histograms, observations are allocated to a bin according to the value of each feature, whereas, with cluster-based histograms, the bin is determined by the distance between the observations and the centroids, as demonstrated in Fig. 3.

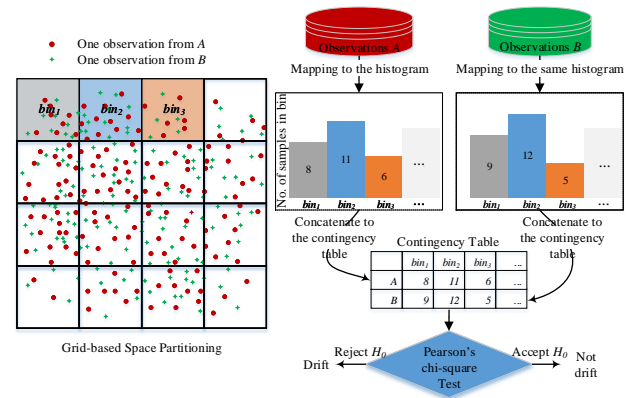


Fig. 2. The process of histogram-based concept drift detection. The number of observations in each partition are counted and then transformed into a contingency table. Pearson's chi-square test outputs the significance of the difference, which is not drift below a threshold and drift above. Note that this diagram is for demonstration purposes only; the number of observations in each bin would need to be much higher (e.g., 50 per bin minimum) for the chi-square approximation to be valid [30].

In practice, there are three advantages to using clusters as histogram bins.

1) The shape of the bins is flexible. Both grid and tree-based histograms have hyper-rectangular-shaped bins. However, the shape of the bins in a cluster-based histogram dynamically adapts to the data distribution, which is beneficial when trying to maintain a similar number of observations per bin and not leaving any empty bins.

2) Missing values in an observation do not need to be estimated one-by-one with cluster-based histograms. Rather, only the observation-to-centroid distances need to be modeled

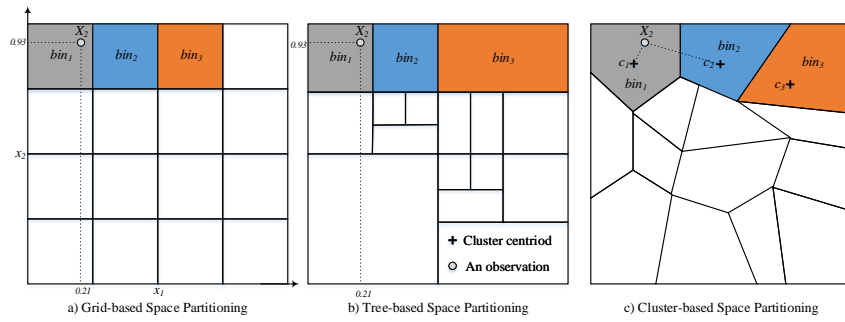


Fig. 3. A demonstration of different space partitioning algorithms: Consider an observation $X = (x_1 = 0.21, x_2 = 0.93)$. Grid- and tree-based algorithms match the value of the features to the bin intervals, whereas cluster-based algorithms compare the observation's distance to each centroid.

to determine the appropriate observation-to-cluster membership. In addition, the error of the distance estimation can be characterized by fuzzifying the distance then using that fuzzy distance to calculate the observation's degree of membership to a cluster.

3) Since our algorithm is task-oriented imputation rather than value-oriented, our drift detection algorithm has no assumption on the data type, i.e., both numerical and categorical data can be addressed.

B. Masked Distance Learning

This section sets out our formulation for a distance learning method to estimate observation-to-observation distances with missing values. The intuition behind the idea is to retrieve the observations without missing values, denoted as $X^{complete}$, and artificially introduce null values as a mask to simulate the missing values, that is,

$$X^{complete} \xrightarrow{f_{mask}(X)} X^{mask},$$

where $f_{mask}(X)$ denotes the missing value masking function. Since our target variable d is known, i.e., the actual paired distances between the retrieved observations, we can use the masked samples X^{mask} and their actual distances d as a training set to build a regression model, denoted as L_{mask} . The standard deviation σ_{mask} of L_{mask} can be used as a coefficient for fuzzifying observation-to-observation distances.

The workflow of this masked distance learning method comprises three major steps, as shown in Fig. 4. Put simply, masked distance learning asks and answers three questions: 1) How must the function $f_{mask}(X)$ be defined so that the observations with missing values have the same distribution as the original sample set? 2) Which method of building the learning model L_{mask} will minimize the prediction errors? 3) How should the error of the L_{mask} be estimated? Step 1-3 are proposed to address these questions.

Step 1 is a sample selection process. Retrieving observations with no missing values can be complete by a for loop over the sample set. The runtime complexity is $\mathcal{O}(mn)$, where m is the number of observations, and n is the dimension of the feature space. The number of observations with missing values is then counted and divided by the total number of observations to produce a missing rate vector, denoted as $V_r = \{r_i\}_{i=1}^n$,

where r_i is the missing rate of the i -th feature. At this stage, we assume the missing values are MCAR. Based on this assumption, the probability mass function of the masking variable for each feature is

$$P(x_i^{mask}) = \begin{cases} r_i, & \text{if } x_i^{mask} \text{ is null} \\ 1 - r_i, & \text{if } x_i^{mask} = x_i \end{cases}, \quad (2)$$

and we have an masked observations as $X_1^{mask} = \{x_i^{mask}\}_{i=1}^n$. The actual pairwise distances of $X^{complete}$ for Step 2 is also calculated at this time.

Step 2 is the prediction model training. In this step, we want to build a learning model that maps the masked observations to the actual pairwise distance while minimizing the error between the predictions and the actual values, i.e.,

$$\min_{h \in \mathcal{H}} \arg |h(X^{mask}) - d|,$$

where \mathcal{H} is the hypothesis space, and $h \in \mathcal{H}$ is the hypothesis of mapping observations to a real value $h : X \rightarrow d$. Few existing data imputation methods are applied to the masked observations, and the masked distances are calculated as new training features. Then these new training features, along with the actual distances prior to masking, are used to train a masked distance learning model. In our solution, we use the gradient boosted decision tree model to perform the learning task, which is a non-linear regression model. We consider that non-linear regression models often outperform linear models in complicated cases.

The basic assumption is that the true distance is close to the imputed distances but which imputation method provides the best results depends on the data. For example, mean replacement is the best option for or Gaussian-distributed features, whereas most frequent fills could be the best for Poisson-distributed features. So, to make the optimal choice, the imputed distances are passed to a regression model and learned. The masked training set contains $\frac{|X^{mask}|^2}{2}$ instances for training, where $|\cdot|$ denotes the cardinality of the set. On the MCAR assumption, the masked distance learning model should perform similarly across the entire sample set.

Step 3 is to estimate the performance of the trained distance learning model L_{mask} . According to the assumption of linear regression, i.e., that $d_i = a + \theta X_i + \epsilon_i$ where the error item $\epsilon \sim \mathcal{N}(0, \sigma^2)$, we can use the error item ϵ (residual) to fuzzify the predicted distance. To estimate the variance, the

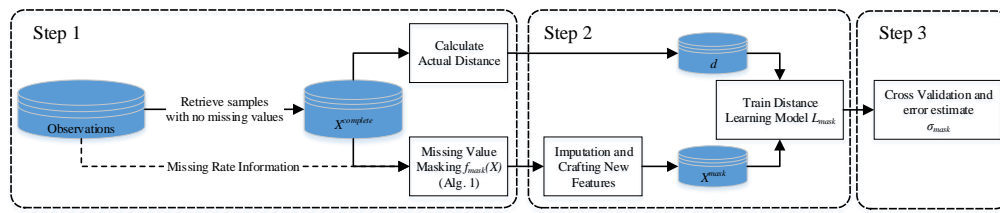


Fig. 4. The workflow of fuzzy distance learning by masking missing values. Step 1. Define the masking function, $f_{mask}(X)$. Step 2. Craft the features and train the distance learning model L_{mask} . Step 3. Estimate the distance prediction error, such as the standard deviation of the residual σ_{mask} .

residual standard error calculated by the equation below and then stored,

$$\sigma_{mask} = \sqrt{\frac{\sum_{i=1}^m (d_i - \hat{d}_i)^2}{m-2}}. \quad (3)$$

For each predicted distance \hat{d}_i , we have high confidence that the true distance is located in the interval $[\hat{d}_i - 3\sigma_{mask}, \hat{d}_i + 3\sigma_{mask}]$ according to the three-sigma rule. To avoid sampling bias, we apply cross-validation to the X^{mask} , which splits X^{mask} into kFold paired subsets $\{X_{train_i}^{mask}, X_{test_i}^{mask}\}_{i=1}^{n_{split}}$, after which the σ_{mask} is computed, where the n_{split} is the number of split.

Alg. 1. presents the pseudocode of masked distance learning. The inputs are the observations, a default regression model, a parameter for cross-validation, and a set of data imputation methods. The outputs are a trained L_{mask} and its residual standard deviation σ_{mask} . The overall runtime complexity is $\mathcal{O}(n_{split}m^2n)$ broken down as follows. The complexity for Lines 1-3 to initialize the data set statistics is $\mathcal{O}(mn)$. The pairwise distance has a complexity of $\mathcal{O}(m^2n)$ without any optimization. The worst case for the observation masking, i.e., when every value is missing, is $\mathcal{O}(mn)$. The worst case for the data imputation is multiple iterative imputations, which would be $\mathcal{O}(m^2n)$. The kFold training-testing complexity is $\mathcal{O}(n_{split}m^2n)$. Calculating standard deviations is $\mathcal{O}(m)$. Therefore, the runtime complexity for Alg. 1. is feasible within the complexity of $\mathcal{O}(n_{split}m^2n)$.

C. Fuzzy-weighted Frequency

After building the missing value distance prediction model L_{mask} and obtaining the standard deviation σ_{mask} of the errors, we can now start estimating the frequency of observations in histogram bins. For any two observations with missing values, we use the same masking function to convert them then pass them to L_{mask} . The returned result \hat{d} with the standard deviation σ_{mask} will be used to construct the fuzzy distance to present its membership to a bin. In fact, there are two options with the predicted distance \hat{d} . One is simply to use the predictions \hat{d} as crisp distances without considering the errors of L_{mask} . The observations would then be assigned to histogram bins in a many-to-one manner, which means each observation would only be counted once by a bin. Another option is to fuzzify the distance according to the σ_{mask} , that is, to consider the estimated distance as a fuzzy number, as shown in Fig. 5.

Algorithm 1: Masked Distance Learning (MDL)

input : 1. Observations, X
 2. Regression model, L_{mask} , default GBDT regressor
 3. Cross-validation times, $n_{split} = 5$ as default
 4. Data imputation algorithm pool, default $\Xi = (\xi_{zero}, \xi_{mean}, \xi_{med}, \xi_{mfr}, \xi_{iter})$

output: 1. Trained distance learning model, L_{mask}
 2. Prediction error, σ_{mask}

```

1 estimate the missing rate vector  $V_r$ ;
2 retrieve non-missing observations,  $X^{complete}$ ;
3 calculate pairwise distances of  $X^{complete}$  as  $d^{mask}$ ;
4 for  $i$  in range( $n$ ) do
5   for  $j$  in range( $m$ ) do
6     mask value  $x_{ji}$  according to  $r_i$ ; // Eq. (2)
7   end
8 end
9 for  $\xi$  in  $\Xi$  do
10  apply data imputation  $X^{impu} = \xi(X^{mask})$ ;
11  calculate pairwise distances of  $X^{impu}$  as  $x_{2n+i}$ ;
12  concatenate  $x_{2n+i}$  to  $X^{mask}$ ;
13 end
14 for  $i$  in range( $n_{split}$ ) do
15  split  $X^{mask}, d^{mask}$  into  $(X_{train}^{mask}, d_{train}^{mask}, X_{test}^{mask}, d_{test}^{mask})$ ;
16  train  $L_{mask}$  with  $X_{train}^{mask}, d_{train}^{mask}$ ;
17  valid  $L_{mask}$  with  $X_{valid}^{mask}, d_{valid}^{mask}$  and store the residuals;
18 end
19 compute the standard deviation  $\sigma_{mask}$ ; // Eq. (3)
20 train  $L_{mask}$  with  $X^{mask}, d^{mask}$ ;
21 return  $(L_{mask}, \sigma_{mask})$ ;

```

The workflow of fuzzy-weighted frequency is as follows. Step 1) model the estimated distance between X_i and C_z as fuzzy distance, $\tilde{d}_{i,z}$; Step 2) calculate the degree of membership $\mu_{i,k}$ of every observation to each bin based on the fuzzy distances. Step 3) normalize the degree of membership as observation's weight. Step 4) Accumulate the weights in each bin as the frequency. The novelty of the FWF compare to fuzzy clustering is that the degree of membership is calculated based on fuzzy distance rather than crisp distance.

In this case, the sample set X is the universe of discourse. The value $\mu(X_i)$ for each $X_i \in X$ is the grade of membership of X_i in (X, μ) . The function $\mu(X_i)$ is the membership function of a fuzzy set $C = (X, \mu)$, where a fuzzy set is a cluster denoted as C_k . When measuring the frequency of observations, the fuzzy distance will be used to determine the degree of membership of an observation to each clusters.

We propose two types of fuzzy numbers to present the estimated distance: the Triangular fuzzy numbers (TFN) and

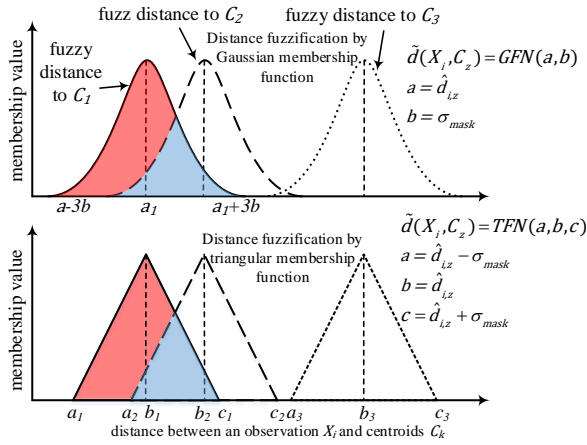


Fig. 5. A demonstration of Gaussian fuzzy distance and triangular fuzzy distance. For an observation X_i with missing values, the distances between X_i and the centroids of the clusters C_1, C_2, C_3 are fuzzified. In this case, X_i might belong to both the C_1 and C_2 clusters but not to C_3 since the upper bound of the estimated distance $\tilde{d}(X_i, C_1)$ is less than the lower bound of $\tilde{d}(X_i, C_3)$ that is $\tilde{d}(X_i, C_1) < \underline{d}(X_i, C_3)$. For simplicity, we use $\hat{d}_{i,1}$ to denote $\tilde{d}(X_i, C_1)$, where the fuzzy distance can be presented by Gaussian fuzzy number (GFN(a, b)) and Triangular fuzzy number (TFN(a, b, c)).

Gaussian fuzzy number (GFN). We chose the TFN as a baseline for comparison because it is the most widely used [12], and we chose the GFN as it is the most reasonable since the predictions and errors are assumed to follow a Gaussian distribution. Once the fuzzy distance formulation is determined, we can leverage them to calculate the degree of membership. The contingency table is then built with the degree of membership.

In general, the membership degree of an observation with missing values to a cluster is determined by its estimated distances to the cluster centroids and the standard deviation of the errors. This is formulated by the membership function $\mu(X_i, C_z)$, namely $\mu_{i,z}$. Without any prior knowledge of the sample set, we assume the observations all have equal weights of 1. Therefore, the fuzzy weighting function can be formulated as a normalized degree of membership, denoted as

$$w_{C_z}(X_i) = \frac{\mu_{i,z}}{\sum_{k=1}^K \mu_{i,k}}. \quad (4)$$

Fig. 6 illustrates how to calculate the weights based on the Triangular fuzzy distance. In this example, the estimated distance between an observation X_i and the cluster centroids $\{C_k\}_{k=1}^4$ are represented with a Triangular fuzzy number. As shown, X_i is closest to the centroid of C_1 and furthest from the centroid of C_4 . Even considering a possible error by L_{mask} , it is still very unlikely that X_i should fall into cluster C_4 , which can be formulated by the condition that $a_4 \geq c_1$. To calculate how likely it is that observation should fall into C_1, C_2, C_3 and C_4 , we use the overlapping areas as a membership function. The most likely cluster for X_i is C_1 with the overlapping area $\mu_{i,1}$, as the areas for $\mu_{i,2}$ and $\mu_{i,3}$ are smaller, and $\mu_{i,4} = 0$. The sum of the total overlapped area is $\mu_{i,1} + \mu_{i,2} + \mu_{i,3} + \mu_{i,4}$. Therefore, we can normalize the weight of X_i to each cluster as $w_{C_1}(X_i) = \frac{\mu_{i,1}}{\mu_{i,1} + \mu_{i,2} + \mu_{i,3} + \mu_{i,4}}$, $w_{C_2}(X_i) =$

$$\frac{\mu_{i,2}}{\mu_{i,1} + \mu_{i,2} + \mu_{i,3} + \mu_{i,4}}, w_{C_3}(X_i) = \frac{\mu_{i,3}}{\mu_{i,1} + \mu_{i,2} + \mu_{i,3} + \mu_{i,4}}, \text{ and } w_{C_4}(X_i) = 0.$$

The degree of membership of $\mu_{i,k}$ is calculated based on the fuzzy distance. We define the triangular fuzzy distance as a TFN(a, b, c), that is,

$$\begin{cases} a = \hat{d}_{i,z} - \sigma_{mask} \\ b = \hat{d}_{i,z} \\ c = \hat{d}_{i,z} + \sigma_{mask} \end{cases},$$

where $\hat{d}_{i,z}$ is the predicted distance between observation X_i and the cluster C_z . Since all triangles are congruent triangles, the triangles of $\mu_{i,z}$ and $\mu_{i,1}$ must be similar. Hence, for simplicity in explaining how the degrees of membership are normalized, we have assumed that the degree of membership of an observation to the closet cluster is equal to 1, i.e., $\mu_{i,1} = 1$. Based on the area theorem of similar triangles, if two triangles are similar, then the ratio of the area of both triangles is proportional to the square of the ratio of their corresponding sides. Therefore, the degree of membership of observation i to its z -th closest cluster is

$$\begin{aligned} \mu_{i,z} &= \left(\frac{c_1 - a_z}{c_1 - a_1} \right)^2 \mu_{i,1} \\ &= \left(\frac{\hat{d}_{i,1} + \sigma_{mask} - (\hat{d}_{i,z} - \sigma_{mask})}{\hat{d}_{i,1} + \sigma_{mask} - (\hat{d}_{i,1} - \sigma_{mask})} \right)^2, \\ &= \frac{(2\sigma_{mask} + \hat{d}_{i,1} - \hat{d}_{i,z})^2}{4\sigma_{mask}^2} \end{aligned} \quad (5)$$

where $\left(\frac{c_1 - a_z}{c_1 - a_1} \right)^2$ is the square of the ratio of their corresponding sides. Substitute to Eq. (4) we have the triangular fuzzy-weighted frequency represented as:

$$w_{C_z}^{Tri}(X_i) = \frac{(c_1 - a_z)^2}{Kc_1^2 - 2c_1 \sum_{i=1}^K a_i + \sum_{i=1}^K a_i^2}. \quad (6)$$

Because of their smoothness and concise notation, Gaussian and bell membership functions are popular methods for specifying fuzzy sets. With these functions, the distance between the cluster centroids and the observations is a Gaussian fuzzy number, GFN(a, b), and the prediction value \hat{d} plus standard deviation σ_{mask} are used to model the distance, that is,

$$\begin{cases} a = \hat{d}_{i,z} \\ b = \sigma_{mask} \end{cases}.$$

The degree of membership of X_i to C_z is the area under the curve and is formulated as

$$\begin{aligned} \mu_{i,z} &= 2 \left(\frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{\frac{a_z + a_1}{2} - a_z}{\sqrt{2}b_z} \right) \right] \right) \\ &= 1 + \operatorname{erf} \left(\frac{\frac{\hat{d}_{i,z} + \hat{d}_{i,1}}{2} - \hat{d}_{i,z}}{\sqrt{2}\sigma_{mask}} \right) \\ &= 1 + \operatorname{erf} \left(\frac{\hat{d}_{i,1} - \hat{d}_{i,z}}{2\sqrt{2}\sigma_{mask}} \right) \end{aligned} \quad (7)$$

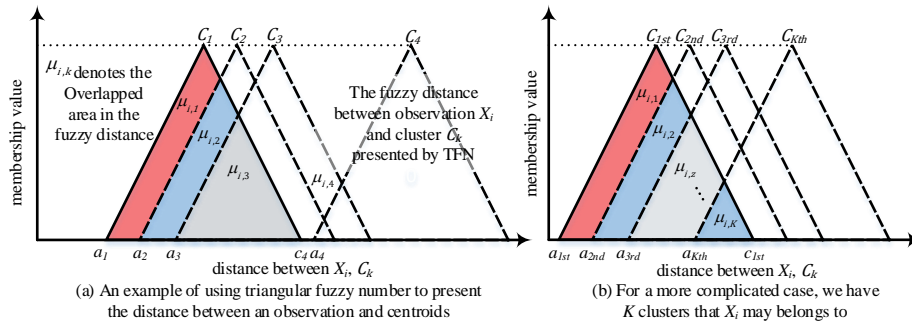


Fig. 6. An illustration of normalizing the degree of membership. K denotes the total number of clusters. $\mu_{i,z}$ denotes the overlapping triangular area used for the weight calculation.

And, similar to the triangular fuzzy number, the Gaussian fuzzy-weighted frequency is

$$w_{C_z}^{Gau}(X) = \frac{1 + \operatorname{erf}\left(\frac{a_1 - a_z}{2\sqrt{2}b_z}\right)}{K + \sum_{i=1}^K \operatorname{erf}\left(\frac{a_1 - a_i}{2\sqrt{2}b_i}\right)}. \quad (8)$$

In contrast to distance fuzzification, the crisp frequency is measured based on \hat{d} , and the frequency measuring is proceeded by finding the closest cluster.

To simplify the computation and to avoid the standard deviation σ_{mask} become too large to provide reasonable fuzzy distances. We propose a Top-Q cluster constrain for normalization the weight, that is, only the closest Q clusters will be considered for frequency measuring, that is, the Eq. (4) is updated as follow.

$$w_{C_z}(X_i, Q) = \frac{\mu_{i,z}}{\sum_{k=1}^Q \mu_{i,k}} I_{z \leq Q}. \quad (9)$$

where, $Q \leq K$, the $I_{z \leq Q}$ is the indicator function that $I_{z \leq Q} = 1$ if $z \leq Q$, otherwise 0. When $Q = 1$, the measured frequency will be equal to the crisp distance frequency. The Alg 2 presents the pseudocode of fuzzy-weighted frequency measuring. With the frequency, we can apply the chi-square test to accept or reject the null hypothesis.

Alg. 2 presents the pseudocode of the fuzzy-weighted frequency. The runtime complexity is determined by k -means clustering (Line 1-3), imputation (Line 4-8) and measuring the fuzzy-weighted frequency (Line 9-15). For the clustering, according to Lloyd's k -means algorithm, the runtime complexity of the clustering procedure in our algorithm is $\mathcal{O}(m^2n)$. The complexity of imputing the data is based on the worst case of the imputation algorithm in Ξ . Given our default settings, the worst case is an iterative imputation with $\mathcal{O}(m^2n)$ on a $m \times n$ data set. The fuzzy weighting process has linear runtime complexity of $\mathcal{O}(mn)$. Therefore, the runtime complexity of Alg. 2 is $\mathcal{O}(m^2n)$. Combining Algs. 1 and 2, the overall runtime complexity of MDL-FWF is $\mathcal{O}(n_{split}m^2n + m^2n) = \mathcal{O}(n_{split}m^2n)$, which is bounded by the MDL algorithm.

IV. EXPERIMENTS AND EVALUATION

In this section, we evaluate the proposed algorithms in three respects. Section IV-A evaluates the distance estimation algo-

Algorithm 2: Fuzzy-weighted Frequency (FWF)

input : 1. Observations, X
 2. Data imputation algorithm pool, default $\Xi = (\xi_{zero}, \xi_{mean}, \xi_{med}, \xi_{mfr}, \xi_{iter})$
 3. Number of clusters (bins), default $K = m/50$
 4. Top-Q constrain, default $Q = 3$
 5. Trained distance learning model, L_{mask}
 6. Prediction error, σ_{mask}
 7. Distance fuzzification model, default TFN

output: 1. Frequency vector, V_k
 2. Cluster centroids, $\mathcal{C} = \{C_k\}_{k=1}^K$

```

1 retrieve non-missing observations,  $X^{complete}$ ;
2  $k$ -means cluster  $X^{complete}$  and get the centroids  $\{C_k\}_{k=1}^K$ ;
3 retrieve missing observations,  $X^{miss}$ ;
4 for  $\xi_i$  in  $\Xi$  do
5   apply data imputation  $X^{impu} = \xi(X^{miss})$ ;
6   calculate distances between  $X^{impu}$  and  $C_k$  as  $x_{2n+i}$ ;
7   concatenate horizontally  $x_{2n+i}$  to  $X^{miss}$ ;
8 end
9 predict distance on  $X^{miss}$ ,  $\hat{d} = L_{mask}(X^{miss})$ ;
10 initial frequency vector  $V_k = \{0\}_{1,...,K}$ ;
11 for  $v_k$  in  $V_k$  do
12   for  $X_i$  in  $X$  do
13     update frequency  $v_k \leftarrow v_k + w_{C_k}(X_i)$ ; // Eqs.
14     (9) (5)(6) or (7)(8) depends on the
15     membership function
16   end
17 end
18 return  $V_k, \mathcal{C}$ 

```

gorithm in comparison to various data imputation methods. Section IV-B tests MDL-FWF's performance at concept drift detection on five synthetic data sets with MCAR missing values. In this set of experiments, we also examine whether fuzzifying the observations improves performance. Section IV-C and IV-D present comparisons with five state-of-the-art concept drift detection algorithms.

A. Masked Distance Learning vs. Calculating Distance from Imputed Values

Experiment 1 - Distance estimation with MCAR.

This experiment evaluate the extent to which masked distance learning can reduce estimation errors.

Data sets.

TABLE I
SYNTHETIC DATA SET CHARACTERISTICS FOR DISTANCE ESTIMATION.

Distribution	Distribution Config	MV Config
Uniform	$x_i \in [0, 10], X = \{x_i\}_{i=1}^{10}$	$V_{r_{1,\dots,5}} = 0.2$ $V_{r_{6,\dots,10}} = 0.0$
Gaussian	$\mu = \text{rand}(0, 1), \sigma = \text{rand}(0, 1)$	Same as above
Exponential	$\beta = \text{rand}(0, 1)$	Same as above
Poisson	$\lambda = \text{randint}(5, 10)$	$V_{r_{1,\dots,3}} = 0.2$ $V_{r_{4,5}} = 0.0$
Categorical	$\text{randint}(0, 10)$	Same as above

For each distribution, we generated 500 observations and randomly removed some values in different configurations, as summarized in Table I. For the uniform, Gaussian, and exponential distributions, we generated $n = 10$ dimensional data. The first 5 dimensions had a missing value rate of 20%. For the Poisson, and uniform categorical distributions, we generated $n = 5$ dimensional data where the first 3 dimensions had a missing value rate of 20%.

Imputation methods and configurations.

Missing values were replaced with five kinds of imputed values: zeros, means, medians, most frequent, and iterative. We generated five sample sets with different distributions and stored the true pairwise distance matrix for each. Then we randomly masked some values as null and applied different distance estimation methods to estimate the pairwise distance matrix. The error between the estimated distance and the true distance was used to evaluate performance.

Evaluation metric.

We used both root mean squared error (RMSE) and mean absolute error (MAE) as the metric for distance estimation evaluation. RMSE gives a relatively high weight to large errors. This means the RMSE should be more useful when large errors are undesirable. MSE does not penalize huge errors as badly as RMSE, but it is a good metric to reflect the overall prediction accuracy.

Findings and discussion.

The distance estimation results are shown in Table III, and the feature importance is plotted in Fig. 7. In this experiment, we find that combining multiple imputation methods and training a regression model improves the overall estimations. It is easy to understand that regression considers the different imputation method will reach the best result. Compared to calculating distance from imputed values, using masked distance learning to predict the distance had lower RMSE and MAE. This inspired us to think that combining multiple instances of uncertainty into one problem and searching for the best optimization solution may produce a better result than solving each uncertainty issue individually.

Regarding the feature importance returned by the gradient boosting decision tree, we noticed that different imputation methods resulted in different importance ranks for different distributions. For example, mean imputation contributed the most for the distance estimation with missing values on uniform and Poisson distributions while, for exponential distributions, zero imputation was more important. The overall estimation results show that integrating diverse imputation method

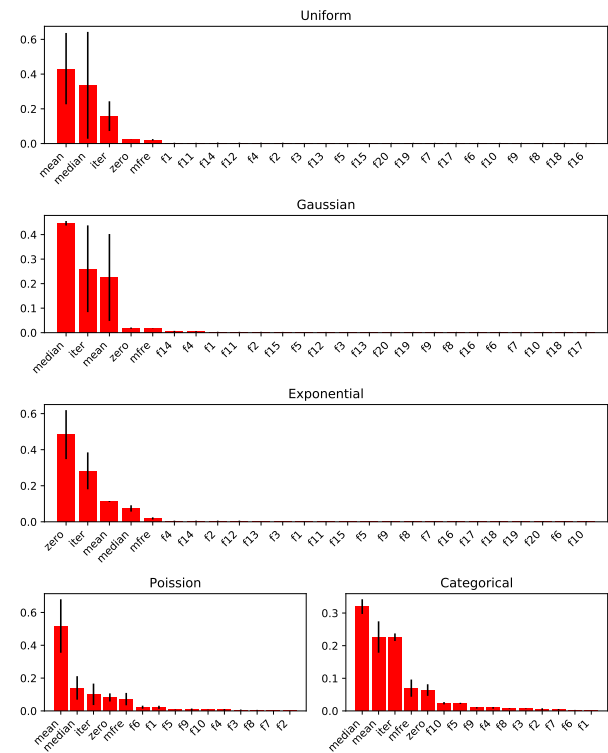


Fig. 7. Feature importance returned by gradient boosting decision tree, calculated based on total gain. For different data sets. The f_i s are the original features. For Uniform distribution, an observation X_1 has 10 features. To compute the paired distance, for example, the distance between X_1 , X_2 , we horizontally concatenated X_1 and X_2 , and append the distance calculated based on imputations. Therefore, the input feature space for L_{mask} are f_1, \dots, f_{20} and $mean, median, zero, mfre, iter$. For Poisson and Categorical data set, there were 5 features for the original observations, therefore, the concatenated features are f_1, \dots, f_{10} and $mean, median, zero, mfre, iter$.

for distance estimation with missing values is effectual.

B. Fuzzy-weighted Frequency vs. Crisp Frequency Drift Detection

Experiment 2 – Algorithm ablation test

In this experiment, we analyzed the contribution of each component's performance of MDL-FWF.

Data sets.

First, we needed to simulate drift, so we generated data sets using three different distributions and five different types of drift. We adopt the commonly used drift synthetic methods for drift detection [7], [36], [37], [39].

Configurations.

Config. 1 – No missing values (Complete). The first configuration is the MDL-FWF baseline, i.e., drift detection with no missing values in the data set. In this configuration, we created k-Means space partitions and counted the observations in each cluster as the frequency. The presence of a drift was determined from a Pearson's chi-square test.

Config. 2 – Impute missing values (MV Impute). In this setting, we duplicated the data set generated in Config. 1 and introduced MCAR missing values by randomly replacing some values with null. Those missing values were then imputed with

TABLE II

RESULTS OF THE MISSING VALUE ESTIMATION. THE MAE AND RMSE ARE CALCULATED BASED ON THE DIFFERENCE BETWEEN THE TRUE VALUES AND THE ESTIMATED VALUES.

		impute_zero	impute_mean	impute_medi	impute_mfre	impute_iter	MIDAS	GAIN	MDL
Uni	MAE	0.071±0.012	0.036±0.006	0.036±0.006	0.071±0.012	0.036±0.006	0.042±0.008	0.039±0.007	NA
	RMSE	0.217±0.022	0.109±0.011	0.012±0.041	0.216±0.022	0.109±0.011	0.133±0.016	0.125±0.016	NA
Gau	MAE	0.802±0.116	0.113±0.020	0.113±0.020	0.418±0.065	0.113±0.020	0.442±0.077	0.133±0.029	NA
	RMSE	2.189±0.176	0.372±0.048	0.141±0.636	1.177±0.113	0.373±0.048	1.294±0.142	0.440±0.073	NA
Exp	MAE	0.105±0.021	0.105±0.021	0.100±0.023	0.143±0.030	0.105±0.021	0.104±0.021	0.112±0.023	NA
	RMSE	0.370±0.069	0.371±0.069	0.156±1.243	0.527±0.088	0.371±0.069	0.370±0.069	0.410±0.070	NA
Poi	MAE	6.417±1.145	0.975±0.200	0.974±0.200	1.025±0.216	0.976±0.199	5.629±1.101	1.179±0.278	NA
	RMSE	16.002±1.749	2.524±0.404	6.566±27.245	2.672±0.450	2.529±0.404	14.989±1.710	3.081±0.667	NA
Cat	MAE	0.643±0.134	0.293±0.053	0.285±0.056	0.342±0.075	0.293±0.053	0.579±0.123	0.320±0.068	NA
	RMSE	1.713±0.248	0.694±0.092	0.544±1.884	0.925±0.149	0.695±0.092	1.566±0.242	0.823±0.140	NA

TABLE III

RESULTS OF THE DISTANCE ESTIMATION OF TWO OBSERVATIONS WITH MISSING VALUES. THE COMPARED METHODS ARE IMPUTATION BY ZERO, MEAN, MEDIAN, MOST FREQUENT, ITERATIVE IMPUTATION, MIDAS, GAIN AND MASKED DISTANCE LEARNING. THE BEST RESULTS APPEAR IN BOLD.

		impute_zero	impute_mean	impute_medi	impute_mfre	impute_iter	MIDAS	GAIN	MDL
Uni	MAE	0.146±0.003	0.087±0.003	0.087±0.003	0.145±0.003	0.087±0.003	0.087±0.003	0.087±0.003	0.077±0.002
	RMSE	0.202±0.003	0.127±0.004	0.127±0.004	0.201±0.004	0.127±0.004	0.127±0.003	0.127±0.004	0.104±0.002
Gau	MAE	3.731±0.031	0.294±0.015	0.294±0.015	1.247±0.122	0.3294±0.015	1.456±0.132	0.304±0.026	0.280±0.014
	RMSE	4.429±0.034	0.469±0.024	0.469±0.024	1.598±0.147	0.469±0.024	1.959±0.145	0.473±0.034	0.403±0.019
Exp	MAE	0.295±0.025	0.296±0.025	0.289±0.026	0.314±0.023	0.296±0.025	0.295±0.025	0.297±0.025	0.309±0.026
	RMSE	0.572±0.069	0.572±0.069	0.575±0.069	0.583±0.060	0.572±0.069	0.572±0.069	0.570±0.067	0.542±0.072
Poi	MAE	21.252±0.204	2.214±0.112	2.214±0.112	2.242±0.113	2.215±0.112	19.098±0.241	2.396±0.300	2.073±0.098
	RMSE	29.043±0.230	3.803±0.203	3.805±0.202	3.834±0.200	3.803±0.203	26.766±0.302	4.018±0.451	3.284±0.164
Cat	MAE	1.390±0.029	0.648±0.029	0.638±0.029	0.692±0.057	0.648±0.029	1.221±0.026	0.660±0.041	0.598±0.027
	RMSE	2.168±0.042	1.046±0.051	1.048±0.050	1.131±0.097	1.046±0.051	1.945±0.042	1.069±0.067	0.915±0.049

iterative imputation. We chose iterative imputation because after comparing the five imputation methods, the iterative approach gave the best performance except MDL. k-Means and Pearson's chi-square tests were used to detect drifts.

Config. 3 – Masked distance learning only (MDL). In this configuration, we used the masked distance learning algorithm to estimate the distance between the cluster centroids and the observations with missing values. Those estimated distances were then used to compute the contingency table. Again, Pearson's chi-square test was used to detect drifts.

Configs. 4.1 & 4.2 – MDL-FWF. Similar to Config. 3, we predicted the distance between the cluster centroids and the observations with missing values via MDL. However, as opposed to using the estimated distances to build the contingency table, we used the fuzzy-weighted frequency measurements via a Gaussian fuzzy membership function (Config. 4.1) and a triangular fuzzy membership function (Config. 4.2). Pearson's chi-square tests determined drift.

Configs. 5.1 & 5.2 – MDL-FWF with the top-Q clusters (MDL-FWF-TopQ). Based on Configs. 4.1 and 4.2, we measured the fuzzy-weighted frequency of the top-Q clusters, instead of considering the degree of the membership to all clusters. The same membership functions as in Configs. 4.1 and 4.2 apply to Configs. 5.1 and Config. 5.2.

Evaluation metric.

The conventional drift detection evaluation metrics are Type-I and Type-II errors. Type-I errors are rejections of a true null hypothesis (also known as “false positives”), and Type II errors are the false null hypothesis rates (i.e., “false negatives”). For concept drift detection, the null hypothesis

is that, there is no concept drift between the populations. Commonly, the Type-I and Type-II errors are shown separately and it is difficult to visualise them in one plot. Therefore, to better visualize the results of each drift detection method, we used the drift detection ratio as the metric [34], which is the number of rejected hypotheses divided by the total number of tests.

To quantify the performance of each configuration, we used Pearson's correlation coefficient as an indicator of the similarity between different drift detection configurations. Since the base model is kMeans, the drift detection ratios of Config. 1 (No MV) forms the baseline and Configs. 2 to 5 show the impact of each component. A higher Pearson's correlation coefficient to Config. 1 means that the component is less affected on the missing values.

Findings and discussion.

Fig. 8 summarizes the results of Experiment 2, and Table IV records the average drift detection ratio in terms of the drift severity level. Further, the Pearson's correlation coefficients for each configuration follows. Bold indicates the best configuration.

- Config. 1 vs. 2: 0.798
- Config. 1 vs. 3: 0.922
- Config. 1 vs. 4.1: 0.651
- Config. 1 vs. 4.2: 0.694
- **Config. 1 vs. 5.1: 0.924**
- **Config. 1 vs. 5.2: 0.944**

These coefficients reveal the impact of each configuration on the drift detection results. Noting that the closer the coefficient

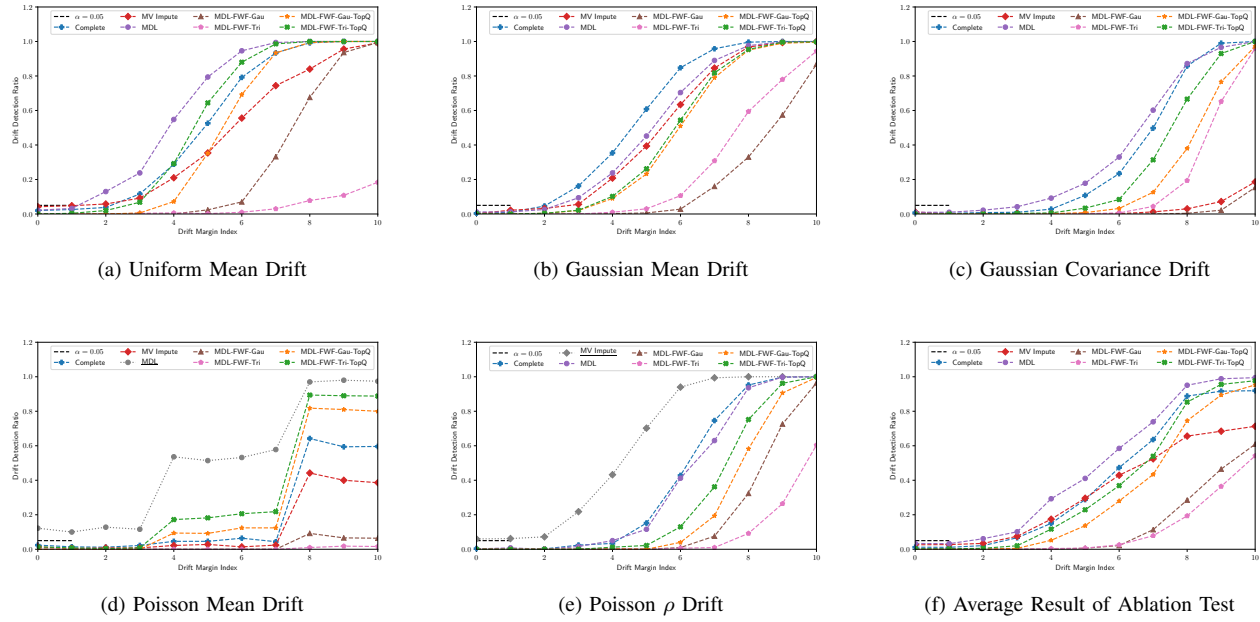


Fig. 8. The drift detection ratios for different components of MDL-FWF on synthetic data sets with different drift margins. The average drift detection ratio was calculated based on the drift margin index. The margin index indicate the severity of the drift. The higher the index, the more severe the drift. We used the average to demonstrate how performance changed with different severities of drift.

TABLE IV

THE DRIFT DETECTION RATIO OF ABLATION TEST. THE HIGHLIGHTED VALUES ARE GREATER THAN THE BASELINE, WHICH MEANS THE COMPONENT INCREASED THE SENSITIVITY OF DRIFT DETECTION.

Drift Severity Level	0	1	2	3	4	5	6	7	8	9	10
1. Complete	0.012	0.011	0.021	0.067	0.150	0.288	0.473	0.636	0.888	0.916	0.919
2. MV Impute	0.026	0.027	0.034	0.075	0.174	0.296	0.429	0.524	0.656	0.684	0.713
3. MDL	0.034	0.032	0.061	0.101	0.293	0.411	0.585	0.739	0.951	0.988	0.995
4.1 MDL-FWF-Gau	0.000	0.000	0.000	0.000	0.001	0.006	0.022	0.114	0.285	0.465	0.609
4.2 MDL-FWF-Tri	0.000	0.000	0.000	0.001	0.004	0.007	0.026	0.078	0.194	0.364	0.542
5.1 MDL-FWF-Gau-TopQ	0.000	0.000	0.001	0.005	0.052	0.137	0.280	0.434	0.745	0.894	0.953
5.2 MDL-FWF-Tri-TopQ	0.003	0.002	0.006	0.020	0.117	0.229	0.369	0.540	0.853	0.956	0.977

to 1, the better the missing value handling method is.

According to the coefficient, it is clear that simply applying data imputation then concept drift detection (Config. 2, MV Impute) did not produce very accurate results. Table IV shows that imputation might increase sensitivity for drifts of low severity but, with very severity drifts, it could have a opsite impact. Config. 3, where the distance between observations was estimated with masked distance learning, had a coefficient (0.922) very close to the baseline. However, the average drift detection ratios shown in Table IV tell us that MDL increased the drift sensitivity in most cases. This opens the possibility that MDL may have a high false alarm when there is no drift and, therefore, will not reach the required confidence level. By contrast, Config. 4.1 and 4.2 (MDL-FWF with Gaussian and triangular functions) performed very poorly. These configurations treat distance estimation errors as an uncertainty issue and apply a fuzzy membership function to model the weight of observations. However, in some cases, the distance estimation errors can be very large, which results in an overestimation of the observations' weights, in turn reducing the sensitivity of drift detection. Our solution was to choose

the top-Q clusters only to handle these overestimated weights. As Configs. 5.1 and 5.2 show, performance with this approach was almost as good as the baseline without increasing the false alarm rate, where the baseline was the drift detection on the complete data sets.

C. A Comparative Study Against Other Drift Detection Algorithms

Experiment 3 – A comparative study

These experiments were designed to evaluate MDL-FWF's performance in comparison to other concept drift detection algorithms. We chose four state-of-the-arts. The evaluation data sets and metric are the same as Experiment 2.

Configurations.

Multivariate Wald-Wolfowitz test (MWW test). This test was developed by Friedman and Rafsky [39] as a multivariate generalization of the Wald-Wolfowitz test. This is the baseline for this experiment. The implementation is publicly available online¹. No specific parameters are required to run this test.

¹<https://gist.github.com/vmonaco/e9ff0ac61fcb3b1b60ba>

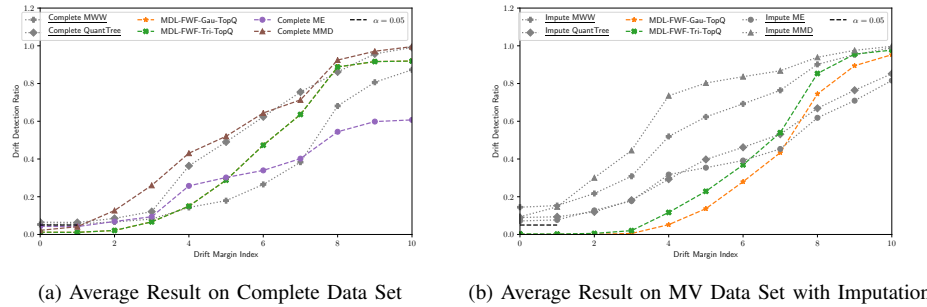


Fig. 9. The average drift detection ratio of different drift detection algorithms on synthetic data sets. The left subfigure shows the drift detection ratio on the complete data sets, that is, there are no missing values in the data sets. The right subfigure shows the drift detection ratio with the missing values imputed. The data sets used in the figures on the right-hand-side were generated by randomly removing values from the data sets used on the left-hand-side. (So the non-missing values in both data sets are the same.) These figures clearly show that the drift detection ratio of all compared algorithms has increased significantly. The drift detection ratios for the MWW, QuantTree, ME and MMD tests were around the desired threshold ($\alpha = 0.05$) at drift margin index 0, but the ratios far exceeded the threshold when the missing values appear.

QuantTree with Pearson statistics is a histogram-based change detection method for multivariate data streams [37]. Pearson's statistic is selected because it has the best reported performance. The implementation is available online². The partition size was set to the sample size divided by 50, which is the same as the default setting for MDL-FWF.

MDL-FWF-TopQ Our proposed algorithm, again, implemented with two different membership functions, i.e., Gaussian and Triangle. The implementation code is available online³.

ME Test is a generic normalized mean embedding (ME) test using a specified kernel. This is used in [43], [44]. The implementation of the ME test is available online⁴. The algorithm parameters were set to the defaults, except for J , which was set to 5 following the authors' recommendation.

Quadratic MMD Test [45], where the null distribution is computed by permutation. The implementation is available online⁵. The parameters were set following the authors' recommendation.

Findings and discussion.

The drift detection results are plotted in Fig. 9, and the average ratios on each of the data sets are shown in Tables V and VI. The Pearson's correlation coefficients of the 50 drift detection ratios were as follows:

- ME missing vs. complete 0.235
- MWW missing vs. complete 0.667
- QuantTree missing vs. complete 0.785
- MMD missing vs. complete 0.887
- **MDL-FWF-Gau-TopQ vs. complete: 0.924**
- **MDL-FWF-Tri-TopQ vs. complete: 0.944**

Missing values impacted the ME test the most, followed by the MWW test, QuantTree, MMD test and the MDL-FWF methods. These results confirm our assumption that missing values introduce uncertainty into distribution-based concept drift detection. Although it is difficult to examine their

general performance on all distributions, these results, at least, demonstrate the need to consider missing values when dealing with distribution changes. Another conclusion we draw from these results is that fuzzy theory has good potential for handle missing values in tasks that required accurate and sensitive concept drift detection.

D. Drift detection on real-world data sets with synthetic drifts and missing values

Experiment 4 – Evaluation with real-world data

Our final test was to compare MLD-FWF with state-of-the-art algorithms in real-world scenarios.

Data sets.

HIGGS Bosons and Background Data Set [35]. The objective of this data set is to distinguish signatures of the processes that produce Higgs boson particles from background processes that do not. We selected four low-level indicators of azimuthal angular momenta for four particle jets as features as the same as [35]. The jet momenta distributions of the background processes are denoted as Back (\mathcal{B}), and the processes that produce Higgs bosons are denoted as Higgs (\mathcal{A}). There were four types of data integration: Back-Back (Type-I Real I.a), where both sample sets were drawn from Back; Higgs-Higgs (Type-I Real I.b), where both sample sets were drawn from Higgs; Back-Higgs (Type-II Real I.a); and Higgs-Back (Type-II Real I.b)

Arabic Digit Mixture Data Set [38]. This data set contains audio features of 88 people (44 females and 44 males) pronouncing Arabic digits between 0 and 9. We applied the same configuration as Denis et al. [38]. The data set has 26 attributes which are a replacement mean and standard deviation for 13 time series. Mixture distribution \mathcal{A} contained randomly selected samples of both males and females, with female labels from 0 to 4 and male labels from 5 to 9. Mixture distribution \mathcal{B} reversed the labels, i.e., males from 0 to 4 and females from 5 to 9. The drift detection on $\mathcal{A} - \mathcal{A}$ is Type-I Real II.a; $\mathcal{B} - \mathcal{B}$ is Type-I Real II.b; $\mathcal{A} - \mathcal{B}$ is Type-II Real II.a; and $\mathcal{B} - \mathcal{A}$ is Type-II Real II.b.

Insects Mixture Data Set [38]. This data set contains 49 features from a laser sensor. The task is to distinguish

²<http://home.deib.polimi.it/boracchi/Projects/projects.html>

³<https://github.com/Anjin-Liu/TFS-MDL-FWF>

⁴<https://github.com/wittawatj/interpretable-test>

⁵<https://github.com/wittawatj/interpretable-test>

TABLE V

THE DRIFT DETECTION RATIO ON COMPLETE DATA SETS. THE UNDERLINE INDICATES THAT THE TYPE-I ERROR EXCEEDS THE DESIRED THRESHOLD $\alpha = 0.05$.

Drift Severity Level	0	1	2	3	4	5	6	7	8	9	10
Complete MWW	<u>0.053</u>	0.056	0.066	0.082	0.143	0.179	0.265	0.383	0.681	0.807	0.873
Complete QuantTree	<u>0.065</u>	0.064	0.085	0.121	0.364	0.49	0.624	0.755	0.862	0.957	0.991
Complete MDL	0.012	0.011	0.021	0.067	0.150	0.288	0.473	0.636	0.888	0.916	0.919
MDL-FWF-Gau-TopQ	0.012	0.011	0.021	0.067	0.150	0.288	0.473	0.636	0.888	0.916	0.919
MDL-FWF-Tri-TopQ	0.012	0.011	0.021	0.067	0.150	0.288	0.473	0.636	0.888	0.916	0.919
Complete ME	0.043	0.042	0.069	0.094	0.2576	0.302	0.340	0.402	0.544	0.598	0.607
Complete MMD	0.022	0.041	0.127	0.26	0.4308	0.520	0.644	0.713	0.924	0.971	0.996

TABLE VI

THE DRIFT DETECTION RATIO ON DATA SETS WITH MISSING VALUES. THE HIGHLIGHTED VALUES ARE THOSE WITH A HIGHER VALUE THAN THE COMPLETE DATA SET RESULTS.

Drift Severity Level	0	1	2	3	4	5	6	7	8	9	10
Impute MWW	<u>0.144</u>	0.153	0.217	0.308	<u>0.520</u>	<u>0.623</u>	<u>0.692</u>	<u>0.764</u>	<u>0.902</u>	<u>0.953</u>	<u>0.990</u>
Impute QuantTree	<u>0.090</u>	0.094	0.120	0.182	<u>0.293</u>	<u>0.398</u>	<u>0.462</u>	<u>0.531</u>	<u>0.669</u>	<u>0.764</u>	<u>0.852</u>
Impute MDL	<u>0.026</u>	<u>0.027</u>	<u>0.034</u>	<u>0.075</u>	<u>0.174</u>	<u>0.296</u>	<u>0.429</u>	<u>0.524</u>	<u>0.656</u>	<u>0.684</u>	<u>0.713</u>
MDL-FWF-Gau-TopQ	0.000	0.000	0.001	0.005	0.052	0.137	0.280	0.434	0.745	0.894	<u>0.953</u>
MDL-FWF-Tri-TopQ	0.003	0.002	0.006	0.020	0.117	0.229	0.369	0.540	0.853	<u>0.956</u>	<u>0.977</u>
Impute ME	<u>0.072</u>	<u>0.077</u>	<u>0.128</u>	<u>0.178</u>	<u>0.318</u>	<u>0.354</u>	<u>0.392</u>	<u>0.453</u>	<u>0.618</u>	<u>0.709</u>	<u>0.816</u>
Impute MMD	<u>0.094</u>	<u>0.146</u>	<u>0.300</u>	<u>0.445</u>	<u>0.735</u>	<u>0.803</u>	<u>0.836</u>	<u>0.868</u>	<u>0.939</u>	<u>0.976</u>	<u>0.997</u>

TABLE VII

DRIFT DETECTION TYPE-I ERRORS WITH THE REAL-WORLD DATA SETS. TtDiff IS THE SUM OF ABSOLUTE DIFFERENCE BETWEEN THE TYPE-I ERRORS WITH AND WITHOUT MISSING VALUES. IMPU STANDS FOR ITERATIVELY IMPUTED MISSING VALUES. COMP IS THE COMPLETE DATA SET WITH NO MISSING VALUES. RANK IS THE SORT ORDER IN TERMS OF AGDIFF.

	Real-I.a		Real-I.b		Real-II.a		Real-II.b		Real-III.a		Real-III.b		TtDiff	Rank
	Impu	Comp	Impu	Comp	Impu	Comp	Impu	Comp	Impu	Comp	Impu	Comp		
MWW	<u>0.508</u>	0.008	<u>0.874</u>	<u>0.076</u>	0.038	0.006	0.040	0.014	0.014	0.002	0.028	0.004	1.39	7
QuantTree	0.048	0.000	<u>0.170</u>	0.036	<u>0.068</u>	0.022	<u>0.200</u>	<u>0.092</u>	<u>0.278</u>	<u>0.062</u>	<u>0.482</u>	<u>0.090</u>	0.94	6
MDL	0.002	0.014	<u>0.016</u>	0.004	0.010	0.044	0.000	0.000	0.014	0.028	0.038	0.008	0.10	2
MDL-FWF-Gau-TopQ	0.000	0.014	0.012	0.004	0.010	0.044	0.000	0.000	0.014	0.028	0.000	0.008	0.08	1
MDL-FWF-Tri-TopQ	0.000	0.014	0.022	0.004	0.002	0.044	0.000	0.000	0.000	0.028	0.000	0.008	0.11	3
ME	<u>0.164</u>	0.020	0.040	<u>0.164</u>	<u>0.052</u>	<u>0.080</u>	0.004	0.000	<u>0.092</u>	<u>0.064</u>	<u>0.186</u>	0.010	0.50	5
MMD	<u>0.226</u>	0.020	<u>0.256</u>	<u>0.060</u>	0.008	0.008	0.002	0.000	0.000	0.000	0.012	0.010	0.41	4

TABLE VIII

DRIFT DETECTION TYPE-II ERRORS WITH THE REAL-WORLD DATA SETS.

	Real-I.a		Real-I.b		Real-II.a		Real-II.b		Real-III.a		Real-III.b		TtDiff	Rank
	Impu	Comp	Impu	Comp	Impu	Comp	Impu	Comp	Impu	Comp	Impu	Comp		
MWW	0.024	0.824	0.013	0.593	0.909	0.933	0.843	0.960	0.985	1.005	0.966	0.995	1.57	6
QuantTree	0.294	0.785	0.369	0.459	0.733	0.970	0.245	0.987	0.505	0.261	0.431	0.829	2.20	7
MDL	0.808	0.470	0.780	0.868	0.415	0.426	0.308	0.535	0.963	0.967	0.634	0.816	0.85	3
MDL-FWF-Gau-TopQ	0.500	0.469	0.996	0.862	0.483	0.423	0.757	0.532	0.960	0.969	0.981	0.817	0.62	1
MDL-FWF-Tri-TopQ	0.481	0.468	0.971	0.865	0.418	0.423	1.002	0.536	1.006	0.965	1.009	0.816	0.82	2
ME	0.729	0.881	0.918	0.786	0.796	0.932	0.970	0.927	0.972	0.703	0.467	0.777	1.04	5
MMD	0.168	0.544	0.133	0.323	0.142	0.188	0.226	0.367	1.008	1.009	0.500	0.626	0.88	4

between 5 possible specimens of flying insects that pass through a laser in a controlled environment (Flies, Aedes, Tarsalis, Quinx, and Fruit). A preliminary analysis showed no drift in the feature space. However, the class distributions gradually change over time. To simulate drift in multiple clusters, we selected the samples from different insects and grouped them together at different percentages; thus, the data distribution varies. The populations are generated baBased on the ratios that $\mathcal{A}=\{Flies : 0.2, Aedes : 0.2, Tarsalis : 0.2, Quinx : 0.2, Fruit : 0.2\}$ and $\mathcal{B}=\{Flies : 0.14, Aedes : 0.14, Tarsalis : 0.2, Quinx : 0.2, Fruit : 0.32\}$

Evaluation metric.

According to David [46], at least 25 samples are required to estimate Pearson, Kendall, and Spearman correlations. Therefore, the results were evaluated based on the absolute total difference between the complete and the imputed drift detection ratio in terms of Type-I and Type-II errors, i.e.,

$$TtDiff = \sum_i^{\#dr} |dr_i^{comp} - dr_i^{impu}|,$$

where dr_i^{comp} indicates the drift detection ratio on complete data sets, dr_i^{impu} is on missing values with imputation, and

$\#dr$ denotes the number of comparisons.

Findings and discussion.

The results of Experiment 4 are summarized in Tables VII and VIII. The results show a similar pattern to Experiment 3, that is, imputing missing values introduces uncertainty and decreases the sensitivity of drift detection. However, MDL-FWF is more robust to missing values than its comparators. From Tables VII and VIII, we can see that MMD was the most powerful drift detector on the complete data set, but, after introducing missing values, the Type-I errors increased dramatically and the Type-II errors dropped significantly. The total difference in Type-I errors for MMD on the six data sets was 0.41, and the difference in Type-II errors was 0.88. Thus the mean of all errors was $\frac{0.41+0.88}{6+6} = 0.1075$. In contrast, MDL-FWF had a mean of only 0.0583 for all errors. This is an improvement of nearly 46%. Such a change in the drift detection results is dangerous because it may increase the number of false alarms and result in unnecessary warnings. Based on the results of these four experiments, we conclude that missing values should be carefully addressed when comparing sample distributions and we find that using fuzzy set theory to model the uncertainty introduced by missing values has a positive effect on distribution discrepancy analysis.

V. CONCLUSION

In this paper, we investigated the influence of missing values on concept drift detection and imputation from a task perspective rather than a value perspective - that is, estimating the statistics we need, as opposed to estimating the missing values then calculating the statistics. The proposed algorithm is one integrated concept drift detection algorithm which can handle missing values, but not an imputation algorithm combined with a concept drift detection algorithm. The results show that applying data imputation then conducting drift detection impairs the performance of today's drift detection algorithms. We attribute this phenomenon to imputation errors and propose an observation fuzzification model to account for the errors. Our proposed drift detection algorithm consists of two major steps. The first is a masked distance learning (MDL) scheme that estimates the distance between observations with missing values and its surrounding cluster centroids. The second is the fuzzy-weighted frequency (FWF) method that considers MDL's estimation errors and uses fuzzy membership to normalize the weighted frequency of observations. Four experiments were conducted to show the efficacy of our method.

In future work, we need to explore a method that can address the disadvantages of complete case analysis, as well as a robust histogram construction method that can handle missing values in extreme cases. The dimensionality of the data is also worth mentioning. Since two observations are concatenated horizontally, which doubles the number of dimensions, computation costs can increase dramatically with high-dimensional data sets. We consider this as our next challenge to solve. Our intention is to begin by exploring whether any dimension reduction techniques might mitigate the problem. At the current stage, we have only applied a very shallow concept of fuzzy-distance with our fuzzy-distance estimation. However, there is

still potential for a distribution estimation based on uncertainty measures. This is a more theoretically-oriented strategy that we intend to explore in a future study

ACKNOWLEDGMENTS

The work presented in this paper was supported by the Australian Research Council (ARC) under Discovery Project DP190101733.

REFERENCES

- [1] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Transactions on Knowl. Data Eng.*, 2018.
- [2] M. Pratama, J. Lu, E. Lughofer, G. Zhang, and M. J. Er, "An incremental learning of concept drifts using evolving type-2 recurrent fuzzy neural networks," *IEEE Transactions on Fuzzy Syst.*, vol. 25, no. 5, pp. 1175–1192, 2016.
- [3] I. Žliobaitė, M. Pechenizkiy, and J. Gama, *An overview of concept drift applications*, book section Chapter 4, pp. 91–114. Springer, 2016.
- [4] J. Lu, A. Liu, Y. Song, and G. Zhang, "Data-driven decision support under concept drift in streamed big data," *Complex & Intell. Syst.*, vol. 6, no. 1, pp. 157–163, 2020.
- [5] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, pp. 1–37, 2014.
- [6] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Learning in nonstationary environments: A survey," *IEEE Comput. Intell. Mag.*, vol. 10, pp. 12–25, 2015.
- [7] N. Lu, G. Zhang, and J. Lu, "Concept drift detection via competence models," *Artif. Intell.*, vol. 209, pp. 11–28, 2014.
- [8] N. Lu, J. Lu, G. Zhang, and R. L. De Mantaras, "A concept drift-tolerant case-base editing technique," *Artif. Intell.*, vol. 230, pp. 108–133, 2016.
- [9] M. Pratama, J. Lu, and G. Zhang, "Evolving type-2 fuzzy classifier," *IEEE Transactions on Fuzzy Syst.*, vol. 24, no. 3, pp. 574–589, 2015.
- [10] M. Pratama, S. G. Anavatti, M. Joo, and E. D. Lughofer, "pclass: An effective classifier for streaming examples," *IEEE Transactions on Fuzzy Syst.*, vol. 23, no. 2, pp. 369–386, 2015.
- [11] M. Das, M. Pratama, S. Savitri, and Z. Jie, "Muse-rnn: a multilayer self-evolving recurrent neural network for data stream classification," in *Proceedings of the 19th IEEE International Conference on Data Mining*, 2019.
- [12] Y. Song, J. Lu, H. Lu, and G. Zhang, "Fuzzy clustering-based adaptive regression for drifting data streams," *IEEE Transactions on Fuzzy Syst.*, vol. 28, no. 3, pp. 544–557, 2020.
- [13] Z. Ning, J. Yu, Y. Pan, and H. Li, "Adaptive event-triggered fault detection for fuzzy stochastic systems with missing measurements," *IEEE Transactions on Fuzzy Syst.*, vol. 26, no. 4, pp. 2201–2212, 2017.
- [14] J. Song, Y. Niu, J. Lam, and H.-K. Lam, "Fuzzy remote tracking control for randomly varying local nonlinear models under fading and missing measurements," *IEEE Transactions on Fuzzy Syst.*, vol. 26, no. 3, pp. 1125–1137, 2017.
- [15] M. S. Santos, R. C. Pereira, A. F. Costa, J. P. Soares, J. Santos, and P. H. Abreu, "Generating synthetic missing data: A review by missing mechanism," *IEEE Access*, vol. 7, pp. 11651–11667, 2019.
- [16] R. J. Little and D. B. Rubin, *Statistical analysis with missing data*, vol. 793. Hoboken, New Jersey, United States: John Wiley & Sons, 2019.
- [17] J. Yoon, J. Jordon, and M. Schaar, "Gain: Missing data imputation using generative adversarial nets," in *Proceedings of the 2018 International Conference on Machine Learning*, pp. 5689–5698, 2018.
- [18] S. v. Buuren and K. Groothuis-Oudshoorn, "Mice: Multivariate imputation by chained equations in r," *J. statistical software*, pp. 1–68, 2010.
- [19] D. J. Stekhoven and P. Bühlmann, "Missforest - non-parametric missing value imputation for mixed-type data," *Bioinforma.*, vol. 28, no. 1, pp. 112–118, 2011.
- [20] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," *J. machine learning research*, vol. 11, no. Aug, pp. 2287–2322, 2010.
- [21] H.-F. Yu, N. Rao, and I. S. Dhillon, "Temporal regularized matrix factorization for high-dimensional time series prediction," in *Proceedings of the 2016 Advances in neural information processing systems*, pp. 847–855, 2016.

- [22] T. Schnabel, A. Swaminathan, A. Singh, N. Chandak, and T. Joachims, "Recommendations as treatments: Debiasing learning and evaluation," in *Proceedings of the 2016 International Conference on Machine Learning*, pp. 1670–1679, 2016.
- [23] P. J. García-Laencina, J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal, "Pattern classification with missing data: a review," *Neural Comput. Appl.*, vol. 19, no. 2, pp. 263–282, 2010.
- [24] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 2008 international conference on Machine learning*, pp. 1096–1103, 2008.
- [25] L. Gondara and K. Wang, "Mida: Multiple imputation using denoising autoencoders," in *Proceedings of the 2018 Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 260–272, Springer, 2018.
- [26] P. D. Allison, *Missing data*. Thousand Oaks, California, United States: Sage publications, 2010.
- [27] C. K. Enders, *Applied missing data analysis*. Guilford Press, 2010.
- [28] X. Wang, R. Wang, and C. Xu, "Discovering the relationship between generalization and uncertainty by incorporating complexity of classification," *IEEE Transactions on Cybern.*, vol. 48, no. 2, pp. 703–715, 2018.
- [29] X. Wang, H. Xing, Y. Li, Q. Hua, C. Dong, and W. Pedrycz, "A study on relationship between generalization abilities and fuzziness of base classifiers in ensemble learning," *IEEE Transactions on Fuzzy Syst.*, vol. 23, no. 5, pp. 1638–1654, 2015.
- [30] G. E. Box, W. G. Hunter, and J. S. Hunter, *Statistics for experimenters: an introduction to design, data analysis, and model building*, vol. 1. JSTOR, 1978.
- [31] A. Liu, G. Zhang, and J. Lu, "Concept drift detection via equal intensity k-means space partitioning," *IEEE Transactions on Cybern.*, vol. Early Access, pp. 1–14, 2020.
- [32] B. W. Silverman, *Density estimation for statistics and data analysis*. Routledge, 2018.
- [33] L. Bu, C. Alippi, and D. Zhao, "A pdf-free change detection test based on density difference estimation," *IEEE Transactions on Neural Networks Learn. Syst.*, vol. 29, no. 2, pp. 324–334, 2018.
- [34] A. Liu, J. Lu, F. Liu, and G. Zhang, "Accumulating regional density dissimilarity for concept drift detection in data streams," *Pattern Recognit.*, vol. 76, pp. 256–272, 2018.
- [35] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, no. Mar, pp. 723–773, 2012.
- [36] T. Dasu, S. Krishnan, S. Venkatasubramanian, and K. Yi, "An information-theoretic approach to detecting changes in multi-dimensional data streams," in *Proceedings of 2006 Symposium on the Interface of Statistics, Computing Science, and Applications*, pp. 1–24, Citeseer, 2006.
- [37] G. Boracchi, D. Carrera, C. Cervellera, and D. Maccio, "Quantree: Histograms for change detection in multivariate data streams," in *Proceedings of the 2018 International Conference on Machine Learning*, pp. 638–647, 2018.
- [38] D. M. dos Reis, P. Flach, S. Matwin, and G. Batista, "Fast unsupervised online drift detection using incremental kolmogorov-smirnov test," in *Proceedings of the 2016 International Conference on Knowledge Discovery and Data Mining*, pp. 1545–1554, ACM, 2016.
- [39] J. H. Friedman and L. C. Rafsky, "Multivariate generalizations of the wald-wolfowitz and smirnov two-sample tests," *The Annals Stat.*, pp. 697–717, 1979.
- [40] K. Honda and H. Ichihashi, "Linear fuzzy clustering techniques with missing values and their application to local principal component analysis," *IEEE Transactions on Fuzzy Syst.*, vol. 12, no. 2, pp. 183–193, 2004.
- [41] S. Van Buuren, *Flexible imputation of missing data*. Chapman and Hall/CRC, 2018.
- [42] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera, "A unifying view on dataset shift in classification," *Pattern Recognit.*, vol. 45, no. 1, pp. 521–530, 2012.
- [43] K. P. Chwialkowski, A. Ramdas, D. Sejdinovic, and A. Gretton, "Fast two-sample testing with analytic representations of probability measures," in *Advances in Neural Information Processing Systems 28* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 1981–1989, Curran Associates, Inc., 2015.
- [44] W. Jitkrittum, Z. Szabó, K. P. Chwialkowski, and A. Gretton, "Interpretable distribution features with maximum testing power," in *Proceedings of the 2016 Advances in Neural Information Processing Systems*, pp. 181–189, 2016.
- [45] W. Bounliphone, E. Belilovsky, M. Blaschko, I. Antonoglou, and A. Gretton, "A test of relative similarity for model selection in generative models," *Proc. 2016 Int. Conf. on Learn. Represent.*, 2016.
- [46] F. N. David, *Tables of the ordinates and probability integral of the distribution of the correlation coefficient in small samples*. Cambridge, United Kingdom: Cambridge University Press, 1938.



Anjin Liu is a Postdoctoral Research Associate in the A/DRsch Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia. He received the BIT degree (Honour) at the University of Sydney in 2012. His research interests include concept drift detection, adaptive data stream learning, multi-stream learning, machine learning and big data analytics.



Jie Lu (F'18) is an Australian Laureate Fellow, IEEE Fellow and IFSA Fellow. She is an internationally renowned scientist in the areas of computational intelligence, specifically in fuzzy transfer learning, concept drift, decision support systems, and recommender systems. She is the Director of the Centre for Artificial Intelligence (CAI), which has over 200 staff and students working on over 50 research projects, and the Associate Dean (Research Excellence) in the Faculty of Engineering and Information Technology at the University of Technology Sydney (UTS).

She has published six research books and over 450 papers in Artificial Intelligence, IEEE Transactions on Fuzzy Systems, IEEE Transactions on Neural Networks & Learning Systems, Decision Support Systems and other refereed journals and conference proceedings. She has won 10 ARC Discovery projects and led 15 industry projects. She has supervised 40 PhD students to completion. She serves as Editor-In-Chief for Knowledge-Based Systems (Elsevier) and Editor-In-Chief for International Journal on Computational Intelligence Systems (Atlantis). She is the Chair of IEEE CIS Emerging Technologies Technical Committee and the Chair of IEEE CIS Subcommittee for Senior Members. She has delivered 25 keynote speeches at IEEE and other international conferences and chaired 15 international conferences. She has received the UTS Medal for Research and Teaching Integration (2010), UTS Medal for research excellence (2019), finalist of the Australian Eureka Prize for Advanced Data Science (2017, 2018), the Computer Journal Wilkes Award (2018), the IEEE Transactions on Fuzzy Systems Outstanding Paper Award (2019), the Australian Most Innovative Engineer Award (2019) and other national and international awards.



Guangquan Zhang is an Associate Professor and Director of the Decision Systems and e-Service Intelligent (DeSI) Research Laboratory at the University of Technology Sydney, Australia. He received the Ph.D. degree in applied mathematics from Curtin University of Technology, Australia, in 2001.

His research interests include fuzzy machine learning, fuzzy optimization, and machine learning. He has authored five monographs, five textbooks, and 460 papers including 220 refereed international journal papers. Dr. Zhang has won seven Australian

Research Council (ARC) Discovery Projects grants and many other research grants. He was awarded an ARC QEII fellowship in 2005. He has served as a member of the editorial boards of several international journals, as a guest editor of eight special issues for IEEE transactions and other international journals, and co-chaired several international conferences and workshops in the area of fuzzy decision-making and knowledge engineering.