Handling concept drifts and limited label problems using semi-supervised combine-merge Gaussian mixture model

Ibnu Daqiqil Id[1], Pardomuan Robinson Sihombing[2], Supratman Zakir[3]
[1]Department of Computer Science, Universitas Riau, Indonesia
[2]BPS-Statistics Indonesia, Indonesia
[3]Faculty of Education and Teaching, Institut Agama Islam Negeri (IAIN) Bukittinggi, Indonesia

Article Info

Article history:

Received Aug 27, 2021
Revised Oct 28, 2021
Accepted Nov 1, 2021

Keywords:

Concept drift
Gaussian mixture model
Label propagation
Label scarcity
Semi-supervised learning

ABSTRACT

When predicting data streams, changes in data distribution may decrease model accuracy over time, thereby making the model obsolete. This phenomenon is known as concept drift. Detecting concept drifts and then adapting to them are critical operations to maintain model performance. However, model adaptation can only be made if labeled data is available. Labeling data is both costly and time-consuming because it has to be done by humans. Only part of the data can be labeled in the data stream because the data size is massive and appears at high speed. To solve these problems simultaneously, we apply a technique to update the model by employing both labeled and unlabeled instances to do so. The experiment results show that our proposed method can adapt to the concept drift with pseudo-labels and maintain its accuracy even though label availability is drastically reduced from 95% to 5%. The proposed method also has the highest overall accuracy and outperforms other methods in 5 of 10 datasets.

Corresponding Author:

Ibnu Daqiqil Id
Department of Computer Science, Universitas Riau
Jalan HR Soebrantas Km 12,5 Pekanbaru, Riau 28293, Indonesia
Email: ibnu.daqiqil@lecturer.unri.ac.id

1. INTRODUCTION

Today, we live in a digital age where practically anything can be done online. Consequently, the volume of data generated and produced is increasing at an exponential rate. According to the recent Statista report [1], the data generated in 2020 is estimated to be 64.2 zettabytes (trillion gigabytes). It is predicted to increase by over 19.2% in 2021-2025. Usually, some of this data is generated continuously and comes in infinite streams, such as IoT and social media data. This data type is known as data stream. Due to these characteristics, the distribution in the data stream might change over time. This change is caused by many factors like the change of human behavior, environment, and trends. The phenomenon where the distribution of data changes over time is called concept drift [2].

Concept drift caused model performance to decrease because the model trains and evaluates with data with different distributions. Many adaptive methods have been developed to address this problem [3], [4]. Most of them work by updating or retraining the model then redeploying them periodically or when the concept drift is detected. The critical component of updating or retraining the model is labeled data. It is impossible to label all the data in streaming data because of the size and the speed of data. Nonetheless, these solutions could be time-consuming and expensive because data labeling may not be available without human input.

Several methods have been proposed to solve the concept drift problem [2], [5]-[7], or label limitation problem [8]. However, when these two problems exist simultaneously, the retraining in a limited label situation may not be effective because it cannot capture the complete concept. In addition, several methods of detecting concept drift are also very dependent on its model output. It automatically limits the ability of the algorithm to detect concept drift because of the label limitation problem. Therefore, apart from adapting to concept drift, the ability to adapt to limited label conditions is an essential feature of the model to solve simultaneously.

In our previous work in [9]-[11], we proposed combine merge Gaussian mixture model (CMGMM) as an incremental algorithm to handle the concept drift problem based on the Gaussian mixture model (GMM). This algorithm can adapt concept drift by including a new component, updating, and removing it to accommodate the emerging concept drift. The main benefits of this algorithm are the ability to adapts and constantly learns from stream data using a local restoration technique to maintain previously learned information and avoid catastrophic forgetting. However, this solution only works when the data are fully labeled.

This paper proposes semi-supervised CMGMM (SCMGMM) to address the concept drift problem and limited label simultaneously. SCMGMM extends the CMGMM that allows us to adapt to new data that is partially labeled. The training and classification process are the same as the CMGMM, but the adaptation process is different. In SCMGMM, the model assigns a pseudo-label to fill the missing label before the adaptation process. This study aims to develop an incremental model and avoid performance degradation due to noisy pseudo-labels. This paper has been divided into the following parts: section 2 is the related work of this research; section 3 explains our proposed method; section 4 describes the experimental setup; section 5 discusses the experimental results; and finally, section 6 provides our conclusion.

2. RELATED WORK

2.1. Concept drift

In this paper, the term concept drift indicates arbitrary statistical change property of a target domain [7], [12]. Statistically, we define a concept as prior class probabilities $p(y)$ and class conditional probabilities $p(X|y)$ [13] where X is the input features and y is the corresponding label y in dataset $\mathfrak{D}$. $p(y)$ and $p(X|y)$ determine the joint distribution $p(X,y)$. In the future, the relationship of $p(y)$, $p(X|y)$, and $p(X,y)$ might change because the change of human or user behaviour and environtment. This situation is expressed as being as:

$$\exists X: p_{w_0}(X,y) \neq p_{w_n}(X,y) \quad (1)$$

In (1) shows a difference in the distribution between the joint probabilities at the two-time window w_0 and w_n. As a result, the model built based on w_0 is not suitable for use on w_n because the relationship between X and y might change, then increasing error and decreasing accuracy on w_n [14].

2.2. Combine-merge Gaussian mixture model

Combine-merge Gaussian mixture model (CMGMM) [9] is an incremental learning algorithm designed to solve the concept drift problem. CMGMM can include new components from newly coming data, update and delete existing components as a response to the change in the current data w_n. This algortihm consist of four main process, namely training best model, data prediction, detect concept drift and model adaptation. In the training process, we employ the expectation maximization (EM) [15] algorithm to train the model and the Bayesian information criterion (BIC) [16] to select the best model. To classify the data, we compute the log-likelihood for each class. Model adaptation intends to adapt the current model in response to newly received data, including new concepts or concept modifications. The result of this adaptation is an adapted weighted mixture component that respects the original mixture. Please refer to [9] for a detailed process and implementation of CMGMM.

3. PROPOSED METHOD

In this paper, we propose semi-supervised CMGMM (SCMGMM) as an extension of the combine-merge Gaussian mixture model (CMGMM) [9] to classify data in concept drifts and limited label situations. Therefore, SCMGMM and CMGMM have many similarities except in the incremental phase. The illustration of the process on the SCMGMM can be seen in Figure 1.

First, we build an optimal model $\mathrm{M}_{pretrain}$ from a fully labeled dataset using the EM [15] algorithm and select the best model using BIC [16]. Not all the data is labeled in the incremental process, so we employ label propagation to generate pseudo labels from unlabeled data. In the incremental process, when Kernel density drift detector (KD3) [7] detects the concept drift, the model triggers the concept drift adaptation process. The adaptation process employs both labeled and pseudo labels data. The detailed difference processes between SCMGMM and CMGMM are explained in the subsections that follow.

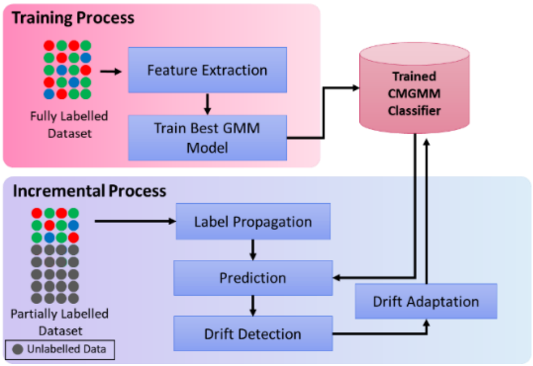

Figure 1. SCMGMM process pipeline

3.1. Pseudo label propagation

The pseudo label propagation process uses labeled data to predict labels for unlabeled data with a prior consistency assumption. The result of the prediction to label the unlabeled data is called a pseudo label. The model used to predict pseudo labels is trained based on current window data D_t using the label spreading method [17], assuming that nearby data instances and the similar structure share the same label. Finally, using the labeled and pseudo label data, we evaluate and adapt the concept drift.

In training, we build a Laplacian matrix of a directed graph utilizing the RBF Kernel from D_t. The edge of the graph is calculated using the RBF kernel in (2). Then we construct the Laplacian matrix S using $S = D^{-1/2}ED^{-1/2}$ where D is a diagonal matrix where the diagonal elements are the sum of the i-row of the E. Finally, we compute F as a class probability distribution to train the data, then apply the regularization function to smooth the label. F is $n \times C$ non-negative matrices that contain the classification result of D_t. A detailed equalization of the probability and regularization function can be seen in [17] at (1) and (4).

$$E_{ij} = \begin{cases} exp\left(-\frac{\|x_i - x_j\|^2}{2\theta^2}\right), i \neq j \\ 0, i = j \end{cases} \tag{2}$$

3.2. SCMGMM drift adaptation

SCMGMM drift adaptation is carried out to combine all existing components and then optimize by reducing the similar components. The main component of this method is a n-Gaussian mixture component denoted by $\{(w_1, \mu_1, P_1), (w_2, \mu_2, P_2), \dots (w_n, \mu_n, P_n)\}$ where w, μ, P are the prior probability or weight, the means, and the covariance. This weight must satisfy $w_1 + w_2 + \cdots + w_n = 1$, and has probability density function in (3).

$$f(X) = \sum_{i=1}^{n} w_i N(x; \mu_i, P_i) \tag{3}$$

In this study, the adaptation process is carried out by adding new components, reducing similar components, and remove components. The adaptation process begins by building a local model M$_{current}$ from the recent incoming data that represent the new concepts or concept updates in the data. Then, we combine the M$_{pretrain}$ and M$_{current}$ components to M$_{combined}$. This process aimed to append any new concepts from M$_{current}$ that may not exist in the M$_{pretrain}$ at the initial training.

When a new component in M$_{combined}$ is not similar to any other components, the process of addition is carried out. But if there are two or more similar components in M$_{combined}$, then the merge process is carried out. The merge process produces a new Gaussian component $(w_{ij}, \mu_{ij}, P_{ij})$ from two Gaussian component $\{(w_i, \mu_i, P_i), (w_j, \mu_j, P_j)\}$ where $w_i + w_j \leq 1$. $(w_{ij}, \mu_{ij}, P_{ij})$ preserve the zeroth-, first- and second-order moments of its original gaussian. The moment-preserving merge process is showed in (4)-(6).

$$w_{ij} = w_i + w_j \tag{4}$$

$$\mu_{ij} = \frac{w_i}{w_i + w_j}\mu_i + \frac{w_j}{w_i + w_j}\mu_j \tag{5}$$

$$P_{ij} = \frac{w_i}{w_i+w_j}P_i + \frac{w_j}{w_i+w_j}P_j + \frac{w_i w_j}{(w_i+w_j)^2}(\mu_i - \mu_j)(\mu_i - \mu_j)^T \tag{6}$$

To measure the similarity, we calculate the Kullback Leiber dissimilarity of two Gaussian component $f(x)$ and $g(x)$ in (7).

$$d_{KL}(\{(w_1,\mu_1,P_1)\},\{(w_2,\mu_2,P_2)\}) = \frac{1}{2}\left[tr(\,P_2^{-1}[P_1 - P_2 + (\mu_1-\mu_2)(\mu_1-\mu_2)^T]) + log\,\frac{\det(P_2)}{\det(P_1)}\right] \quad (7)$$

The detailed algorithm for model adaptation is shown in algorithm 1

```
Algorithm 1: Model adaptation
Input: Pre-train SCMGMM $M_{\mathrm{pretrain}}$, Labeled data $\mathcal{U}$, Unlabeled data $L$
Result: Adapted Model
Adaptation_data = $\mathcal{U}$+ labelPropagation(L)
$M_{\mathrm{current}}$ = findBestGMM (Adaptation_data)
$M_{\mathrm{combine}}$ = CombineGMMComponent ($M_{\mathrm{current}}$, $M_{\mathrm{pretrain}}$)
distanceMatrix= KL-Dissimalarity($M_{\mathrm{combine}}$)
for targetComponent to nComp
    $M_{\mathrm{merge}}$ = mergeComponent (target, distanceMatrix)
    $\mathrm{BIC}_{\mathrm{candidate}}$ = ComputeBIC ($M_{\mathrm{merge}}$, ds)
    if $\mathrm{BIC}_{\mathrm{candidate}}$ < $\mathrm{BIC}_{\mathrm{best}}$ then
      $M_{\mathrm{best}}$ = $M_{\mathrm{merge}}$
    end
end
$M_{\mathrm{best}}$=ComponentPrune ($M_{\mathrm{best}}$)
return $M_{\mathrm{best}}$
```

4. EXPERIMENT SETUP

This section describes the datasets, comparison methods, evaluation metrics, and hyperparameters used in this study to train and evaluate the proposed method.

4.1. Datasets

We used real-world and synthetic datasets to evaluate the proposed algorithm. These datasets contain concept drift with a certain level of label availability in the dataset, namely 90%, 75%, 50%, 25%, 5%, and 1%. The datasets used in experiments are given as shown in:

- Rotating hyperplane dataset [18] is an artificial dataset containing hyperplane data in d-dimensional space that continuously changes its position and orientation.
- SEA dataset [19] is an artificial dataset that consists of three attributes and 50,000 data instances. This dataset also contains abrupt concept drift simulated by four different concepts every 12,500 data points by changing the class decision boundary.
- CR4 [20] is an artificial dataset containing 144,400 samples using four classes rotating separately in 2-dimensional space.
- FG2C2D [20] is an artificial dataset that contains two bidimensional classes and two concept drift, namely gradual and incremental concept drift, every 200 data points. There are 200,000 samples and two classes in this dataset.
- GEAR2C2D [20] is artificial dataset that contain two rotating gears represented as two classes. There are 200,000 samples and two classes in this dataset.
- MG2C2D [21] is artificial dataset that contains two bidimensional multimodal Gaussian classes.
- NOAA is real-world weather dataset collected over 50-years at Bellevue, Nebraska. This dataset has 18,159 samples in two classes, namely rain, and no-rain.
- Electricity is one of the famous real-world dataset used to evaluate concept drift problems. This dataset collected by [22] contains electricity market records in New South Wales, Australia.
- Spam dataset is collected by [23] to separate malicious spam emails from legitimate ones.
- Phishing is collected by [24] containing data about malicious web pages.

4.2. Comparison methods

In our evaluation, we compare our proposed algorithm SCMGMM with three state-of-the-art algorithms from the related literature, namely:

- Self adjusting memory KNN (SAM-kNN) [25] is an incremental learning algorithm based on k-nearest neighbor (kNN). This algorithm is designed to able to deal with varied types of concept drift within streaming data.
- Learn++.NSE ensemble classifier [12] is an incremental ensemble classifier to solve a nonstationary environment (NSE). This algorithm uses weighting strategy and base classifiers association mechanism to track concept drift.

– Hoeffding tree [26] is an incremental decision tree learning algorithm for streaming data. It keeps track of the most recent streaming instances in a graph. To replace an old instance with a new one, a star-mesh transform technique is used.

These algorithms are designed to solve only the concept drift problem. Therefore, we apply the pseudo label propagation method to these algorithms to solve the limited label problem. The source code of the proposed and comparison method implementation and the dataset is available in our public repository[1].

4.3. Evaluation method and metric

We evaluate the proposed method performance using the interleaved test-then-train or prequential evaluation method. All the classifiers are trained using the same fully labeled training dataset. In the evaluation, newly incoming data is evaluated in specific windows. At this stage, the label of the data is removed according to the label availability setting. Furthermore, we perform pseudo labeling and test the incoming data stream, then updates or retrain the model if concept drifts are detected. The model is constantly tested on the data stream that has not been seen when evaluated in this order. The benefit of this approach does not require a holdout set for testing and allowing us to utilize the existing data efficiently.

5. RESULTS AND DISCUSSION

Table 1 presents the experimental results of SCMGMM, SAM-kNN, Learn++NSE ensemble classifier, and hoeffding tree in 10 datasets. SCMGMM shows better accuracy than other methods, especially in CR4, GEARS2C2D MG2C2D, and NOAA. In all models and datasets, the model's accuracy is in line with the accuracy of the label propagation. When the label propagation accuracy is low, then the model accuracy is also low and vice versa. For example, in SEA 90%, SCMGMM accuracy is 0.8258, and label propagation accuracy is 0.7534.

In the artificial dataset, CR4, FG2C2D, GEARS2C2D, MG2C2D, where the data is generated using a Gaussian distribution, the use of graph-based and SCMGMM gave better results. However, SCMGMM shows its worst performance in the hyperplane dataset. This dataset contains a set of data point that were created by the hyperplane $w_i x_i = w_0$ rotate with random direction so that the graph-based pseudo labeling method cannot work well.

Other algorithms like the Hoeffding Tree and SAM-KNN also show good accuracy. The Hoeffding tree shows its best accuracy in Hyperplane, SEA, and Electrical. SAM-KNN shows its best accuracy only in Phishing. However, the accuracy of Learn++NSE is very low on several datasets such as CR4, MG2C2D, FG2C2D, and GEARS2C2D, where this method was accurate less than 10% of the time. This algorithm is not suitable for the syntectic dataset generated by the Gaussian distribution.

Misclassification on the pseudo label can be said to be noise at the adaptation stage. This noise could decrease the model accuracy. The lower the label availability, the higher the noise on the pseudo label. In Figure 2, all classifiers that use the label propagation method can maintain their performance until label availability is 5%. Decreasing label availability from 90% to 5% only reduced label propagation performance by 5.2% (on average). The accuracy of label propagation reflected the model performance; the higher the label propagation accuracy, the better the model performance. However, in label availability 1%, there are significant performance decrements, especially in SEA, FG2C2D, and Phishing.

The statistical analysis of experimental results was conducted for four algorithms with 60 paired samples. The significant test level is alpha=0.050. For the populations SCMGMM (p=0.001), hoeffding tree (p=0.000), SAM-kNN (p=0.003), and learn++NSE (p=0.001), we rejected the null hypothesis that the population is normal. Thus, we assumed that not all populations are normal.

Furthermore, we utilize the non-parametric Friedman test as an omnibus test to check whether there are any significant differences between the median values of the populations because we have more than two populations, and some of them are not normally distributed. We also use the post-hoc Nemenyi test to infer which differences are significant. In this test, we report the median (MD), the median absolute deviation (MAD), and the mean rank (MR) among all populations over the samples. Differences between populations are significant if the mean rank difference is greater than the critical distance CD=0.606 of the Nemenyi test. Figure 3 shows the result of the Post-hoc Nemenyi test.

We reject the null hypothesis (p=0) of the Friedman test that there is no difference in the central tendency of the populations SCMGMM (MD=0.804±0.122, MAD=0.193, MR=1.950), Hoeffding Tree (MD=0.829±0.097, MAD=0.156, MR=1.958), SAM-KNN (MD=0.821±0.125, MAD=0.195, MR=2.508), and Learn++NSE (MD=0.557±0.346, MAD=0.411, MR=3.583). Therefore, we assume that there is a statistically significant difference between the median values of the populations.

Table 1. Comparison of model accuracy at certain label availability levels

Dataset	Label Availability	Label Propagation Accuracy	Model Accuracy			
			SAM-KNN	Learn++ NSE	Hoeffding Tree	SCMGMM*
Hyperplane	90%	0.5823	0.5903	0.5627	0.6551	0.6149
	75%	0.5803	0.5899	0.5568	0.6472	0.6122
	50%	0.5701	0.5830	0.5575	0.6352	0.6021
	25%	0.5665	0.5662	0.5442	0.6110	0.5809
	5%	0.5448	0.5279	0.5254	0.5682	0.5367
	1%	0.5273	0.5316	0.5205	0.5259	0.522
SEA	90%	0.7534	0.8215	0.7507	0.8258	0.8018
	75%	0.7584	0.8118	0.7496	0.8331	0.8069
	50%	0.7519	0.7915	0.7299	0.8216	0.7964
	25%	0.7423	0.7536	0.7073	0.7995	0.7672
	5%	0.7067	0.7039	0.6856	0.7641	0.708
	1%	0.6356	0.5823	0.5777	0.6362	0.5992
CR4	90%	0.9999	0.9993	0.0007	0.8154	0.9998
	75%	0.9999	0.9992	0.0006	0.8428	0.9998
	50%	0.9998	0.9989	0.0006	0.8433	0.9998
	25%	0.9998	0.9983	0.0007	0.8367	0.9997
	5%	0.9992	0.9981	0.0009	0.8642	0.9991
	1%	0.8485	0.9530	0.0040	0.9580	0.9580
FG2C2D	90%	0.9546	0.9620	0.0216	0.9591	0.9615
	75%	0.9531	0.9625	0.0214	0.9591	0.9631
	50%	0.9528	0.9603	0.0204	0.9617	0.9626
	25%	0.9496	0.9555	0.0244	0.9588	0.9590
	5%	0.9408	0.9371	0.0351	0.9467	0.9438
	1%	0.8234	0.7684	0.0518	0.8214	0.7827
GEARS2C2D	90%	0.9905	0.9919	0.0076	0.9617	0.9965
	75%	0.9907	0.9904	0.0092	0.9628	0.9950
	50%	0.9903	0.9879	0.0099	0.9654	0.9931
	25%	0.9890	0.9876	0.0098	0.9672	0.9903
	5%	0.9649	0.9526	0.0233	0.9562	0.9645
	1%	0.8774	0.8339	0.0715	0.8841	0.8678
MG2C2D	90%	0.9216	0.9264	0.0470	0.9083	0.9333
	75%	0.9174	0.9296	0.0409	0.9147	0.9358
	50%	0.9139	0.9275	0.0386	0.9155	0.9357
	25%	0.9053	0.9170	0.0442	0.9115	0.9296
	5%	0.8912	0.8863	0.0583	0.8877	0.9058
	1%	0.8164	0.7606	0.0765	0.8123	0.8095
ELEC	90%	0.6927	0.6549	0.7132	0.7472	0.7364
	75%	0.6720	0.6693	0.6913	0.7342	0.7140
	50%	0.6632	0.6737	0.6844	0.7131	0.6950
	25%	0.6483	0.6554	0.6482	0.6751	0.6554
	5%	0.5782	0.5124	0.5487	0.5543	0.5223
	1%	0.5362	0.5152	0.5378	0.5358	0.5123
NOAA	90%	0.7032	0.7335	0.6845	0.7338	0.7420
	75%	0.6933	0.7306	0.7068	0.7210	0.7398
	50%	0.6943	0.7285	0.7485	0.7138	0.7283
	25%	0.6955	0.7282	0.7365	0. 7009	0.7293
	5%	0.6895	0.7066	0.7029	0.7143	0.7178
	1%	0.6883	0.6515	0.6735	0.6780	0.6818
Spam	90%	0.8503	0.8210	0.8279	0.8715	0.9163
	75%	0.8293	0.8318	0.8122	0.8680	0.8638
	50%	0.8142	0.8609	0.8436	0.8713	0.7873
	25%	0.7703	0.8837	0.8513	0.8517	0.7664
	5%	0.7543	0.7017	0.7938	0.7165	0.6785
	1%	0.6941	0.6306	0.6109	0.6921	0.6147
Phising	90%	0.9227	0.9065	0.8912	0.9013	0.8965
	75%	0.9162	0.9060	0.8842	0.8992	0.8918
	50%	0.9091	0.8998	0.8588	0.8921	0.8722
	25%	0.8973	0.8833	0.8415	0.8667	0.8439
	5%	0.8386	0.7841	0.7765	0.7275	0.7651
	1%	0.6860	0.5756	0.6233	0.6630	0.6045
Average	90%	0.8371	0.8407	0.4507	0.8379	0.8599
	75%	0.8310	0.8421	0.4473	0.8382	0.8522
	50%	0.8259	0.8412	0.4492	0.8333	0.8372
	25%	0.8163	0.8328	0.4408	0.8309	0.8221
	5%	0.7908	0.7710	0.4150	0.7699	0.7741
	1%	0.7133	0.6802	0.3747	0.7206	0.6952

*Proposed algorithm

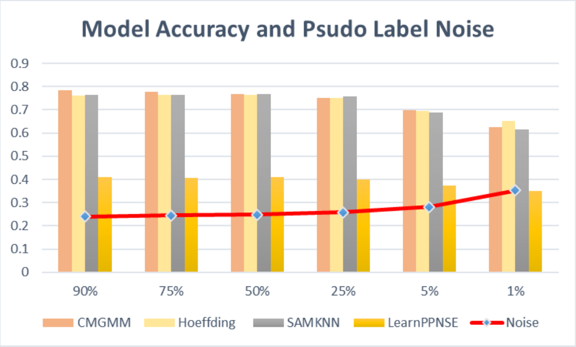

Figure 2. Comparison of model accuracy and noise in the pseudo label

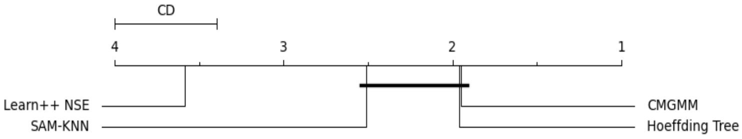

Figure 3. Post-hoc Nemenyi test result

6. CONCLUSION

This paper proposes an incremental algorithm that solves the concept drifts and label scarcity situations. The experiment result shows that our proposed algorithm outperforms 5 of 10 datasets and has the highest overall accuracy. The accuracy of the label propagation dramatically affects the accuracy of the model; the higher the accuracy, the higher the accuracy of the model. Our contributions are our proposed method to maintain model performance using pseudo label propagation even though label availability is drastically reduced from 95% to 5%, and the label propagation also can be applied to other incremental methods. As part of our future work, we plan to investigate the effectiveness of adapted data supplied by concept drift detectors in several types of concept drift. Furthermore, we plan to investigate this algorithm with complex feature datasets like images, sounds, and videos.

REFERENCES

[1] A. Holst, “Volume of data/information created, captured, copied and consumed worldwide from 2010 to 2025,” *Statista*, 2021. [Online]. Available: https://www.statista.com/statistics/871513/worldwide-data-created. [Accessed: 17-Jan-2021].

[2] J. Gama, R. Sebastião and P. P. Rodrigues, “On evaluating stream learning algorithms,” *Mach. Learn.*, vol. 90, no. 3, pp. 317–346, 2013.

[3] M. M. Masud *et al.*, “Facing the reality of data stream classification: coping with scarcity of labeled data,” *Knowl. Inf. Syst.*, vol. 33, no. 1, pp. 213–244, Oct. 2012.

[4] J. LIU, G. XU, S. ZHENG, D. XIAO and L. GU, “Data streams classification with ensemble model based on decision-feedback,” *J. China Univ. Posts Telecommun.*, vol. 21, no. 1, pp. 79–85, Feb. 2014.

[5] N. Wanas, A. Farouk, D. Said, N. Khodeir and M. Fayek, “Detection and Handling of Different Types of Concept Drift in News Recommendation Systems,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 11, no. 01, pp. 87–106, 2019.

[6] P. Zhao, L. W. Cai and Z. H. Zhou, “Handling concept drift via model reuse,” *arXiv*, pp. 533–568, 2018.

[7] J. P. Barddal, H. M. Gomes, F. Enembreck and B. Pfahringer, “A survey on feature drift adaptation: Definition, benchmark, challenges and future directions,” *J. Syst. Softw.*, vol. 127, pp. 278–294, 2017.

[8] A. E. Mehyadin and A. M. Abdulazeez, “CLASSIFICATION BASED ON SEMI-SUPERVISED LEARNING: A REVIEW,” *Iraqi J. Comput. Informatics*, vol. 47, no. 1, pp. 1–13, 2021.

[9] I. D. Id, M. Abe and S. Hara, “Concept Drift Adaptation for Acoustic Scene Classifier based on Gaussian Mixture Model,” in *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, 2020, vol. 2020-Novem, pp. 450–455.

[10] I. D. Id, M. Abe and S. Hara, “Evaluation of Concept Drift Adaptation for Acoustic Scene Classifier Based on Kernel Density Drift Detection and Combine Merge Gaussian Mixture Model,” in *2021 Spring meeting of the Acoustical Society of Japan*, 2021.

[11] I. D. Id, M. Abe and S. Hara, “Acoustic Scene Classifier Based on Gaussian Mixture Model in the Concept Drift Situation,” *Adv. Sci. Technol. Eng. Syst. J.*, vol. 6, no. 5, pp. 167–176, Sep. 2021.

[12] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning*. 2014.

[13] T. R. Hoens, R. Polikar and N. V. Chawla, "Learning from Streaming Data with Concept Drift and Imbalance: An Overview," *Prog. Artif. Intell.*, vol. 1, no. 1, pp. 89–101, 2012.

[14] A. Tsymbal, "The Problem of Concept Drift: Definitions and Related Work," 2004.

[15] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. R. Stat. Soc. Ser. B*, vol. 39, no. 1, pp. 1–38, Sep. 1977.

[16] A. D. R. McQuarrie and C.-L. Tsai, *Regression and Time Series Model Selection*, vol. 6, no. 2. WORLD SCIENTIFIC, 1998.

[17] D. Zhou, O. Bousquet, T. N. Lal, J. Weston and B. Sch, "Learning with Local and Global Consistency," vol. 1.

[18] A. Bifet, B. Pfahringer, J. Read and G. Holmes, "Efficient data stream classification via probabilistic adaptive windows," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing - SAC '13*, 2013, p. 801.

[19] W. N. Street and Y. Kim, "A streaming ensemble algorithm (SEA) for large-scale classification," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '01*, 2001, pp. 377–382.

[20] S. Ud Din, J. Shao, J. Kumar, W. Ali, J. Liu and Y. Ye, "Online reliable semi-supervised learning on evolving data streams," *Inf. Sci. (Ny).*, vol. 525, pp. 153–171, Jul. 2020.

[21] K. B. Dyer, R. Capo and R. Polikar, "COMPOSE: A Semisupervised Learning Framework for Initially Labeled Nonstationary Streaming Data," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 25, no. 1, pp. 12–26, Jan. 2014.

[22] M. Harries, "Splice-2 comparative evaluation: Electricity pricing," 1999.

[23] I. Katakis, G. Tsoumakas, E. Banos, N. Bassiliades and I. Vlahavas, "An adaptive personalized news dissemination system," *J. Intell. Inf. Syst.*, vol. 32, no. 2, pp. 191–212, Apr. 2009.

[24] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009, pp. 1–6.

[25] V. Losing, B. Hammer and H. Wersing, "KNN Classifier with Self Adjusting Memory for Heterogeneous Concept Drift," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 2016, pp. 291–300.

[26] C. Manapragada, G. I. Webb and M. Salehi, "Extremely fast decision tree," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, no. August, pp. 1953–1962, 2018.

BIOGRAPHIES OF AUTHORS

Ibnu Daqiqil Id the author obtained his BA in computer science at Universitas Brawijaya and Master's of information technology at Universitas Indonesia. Now he is working as a lecturer at the Universitas Riau. His research interests center around machine learning and software engineering. His current research is a classification under nonstationary environment and continual learning.

Pardomuan Robinson Sihombing the author works at the BPS-Statistics Indonesia. He completed his undergraduate education at the Sekolah Tinggi Ilmu Statistik (STIS) majoring in economic statistics, then went on to earn his Master's at Padjadjaran University in the department of applied statistics. Currently, the author is continuing his doctoral studies at IPB University, majoring in Statistics and Data Science. The author's research concerns are statistics, econometrics, social science, and machine learning.

Supratman Zakir the author has studied at the Educational Technology Masters Study Program, at Padang State University (UNP), and completed the Master Program in Computer Science at the Putra Indonesia University (UPI) Padang. He completed his Doctoral Program Education in the Technical and Vocational Education Training (TVET) Study Program at Padang State University (UNP). The author concentrates on research in Information Technology-based Learning Technology Development, such as IT-based learning media, Web Base Learning, Mobile Learning, and Hybrid Learning.