Journal of Big Data

RESEARCH

Open Access

Distance variable improvement of time-series big data stream evaluation

Ari Wibisono*, Petrus Mursanto, Jihan Adibah, Wendy D. W. T. Bayu, May Iffah Rizki, Lintang Matahari Hasani and Valian Fil Ahli

*Correspondence:
ari.w@cs.ui.ac.id
Faculty of Computer Science, Universitas Indonesia, Indonesia, Kampus, UI Depok, Depok 16424, Indonesia

Abstract

Real-time information mining of a big dataset consisting of time series data is a very challenging task. For this purpose, we propose using the mean distance and the standard deviation to enhance the accuracy of the existing fast incremental model tree with the drift detection (FIMT-DD) algorithm. The standard FIMT-DD algorithm uses the Hoeffding bound as its splitting criterion. We propose the further use of the mean distance and standard deviation, which are used to split a tree more accurately than the standard method. We verify our proposed method using the large Traffic Demand Dataset, which consists of 4,000,000 instances; Tennet's big wind power plant dataset, which consists of 435,268 instances; and a road weather dataset, which consists of 30,000,000 instances. The results show that our proposed FIMT-DD algorithm improves the accuracy compared to the standard method and Chernoff bound approach. The measured errors demonstrate that our approach results in a lower Mean Absolute Percentage Error (MAPE) in every stage of learning by approximately 2.49% compared with the Chernoff Bound method and 19.65% compared with the standard method.

Keywords: Intelligent Systems, Data stream, Distance improvement, Big data regression

Introduction

Real-time information mining for regression problems involving a huge time series dataset is becoming an increasingly challenging task in the data mining community. This condition has motivated some researchers to develop an incremental algorithm that executes fast and is able to accurately adapt to such problems. Adak and Akpinar [1] applied a hybrid approach combining the artificial honey bee algorithm and multiple linear regression for processing time-series datasets. In addition, Aghaborzogi and Wah [2] employed a multi-step clustering approach to discover valuable pieces of information from big time-series datasets. Such an approach was taken since the classic data mining method proved to be ineffective at processing big time-series datasets that generally have large dimensionality, high feature correlation, and vast amounts of noise [2].

Researchers have proposed a number of algorithms to address these issues. An incremental stream mining algorithm that can predict and form model trees was introduced by Ikonomovska et al. [3]. This algorithm uses the Standard Deviation Reduction (SDR) as a method to determine the splitting criterion and uses the Hoeffding bound to evaluate and determine the mechanism of the tree splitting process [4]. It uses the Binary Search Tree (E-BST) as its tree structure and calculates a linear model for the leaves using the linear model perceptron. Moreover, the online change detection of this algorithm is measured using Page–Hinckley (Ph) change detection. We have improved the accuracy of the fast-incremental model tree with drift detection (FIMT-DD) algorithm [5], which was developed by Ikonomovska et al. [3]. The authors suggested using tanh as its activation function rather than using a linear activation function.

Zhang presented Bennet-type generalization bounds for a learning process with independent and identically distributed (i.i.d.) samples [6]. The authors provide two types of Bennet-deviations: the first provides a generalization bound using uniform entropy numbers, and the second uses the Rademacher complexity. The results showed that an alternative expression that is developed results in a faster rate of convergence than traditional results. Beygelzimer et al. proposed an online algorithm to develop a logarithmic depth tree to predict the conditional probability of a label [7]. The natural reduction of the problem is examined to make a set of binary regressions in the form of a tree, and then it determines a regret bound that changes based on the depth of the tree. A new algorithm framework for non-parametric testing was developed in [8]. The authors presented sequential non-parametric testing with the law of the iterated algorithm. The novel approach presented in this paper conducts on-the-fly testing computations, which take linear time and constant space.

Several researchers use various techniques to assess large datasets. The discovery of a connection between the traffic flow and weather parameters is presented in [9]. This study constructs a deep belief network architecture to predict the weather and traffic flows. The results of this research showed that the weather data affected the traffic flow prediction, and a data fusion technique could increase the accuracy of traffic flow prediction.

A technique to predict the traffic flow using deep learning and Dempster–Shafer theory is introduced by Soua et al. [10]. In this research, the authors divided data into two categories: event-based data and a data stream. The authors applied deep belief networks to predict the traffic flow and Tennet's wind power plant dataset using the data stream and event-based data, and Dempster–Shafer theory was used to renew the belief and integrate the results.

We have created a framework to visualize and predict very large traffic flows by using the FIMT-DD algorithm. Detailed visualization of the traffic flow and Tennet's wind power plant dataset is developed from the prediction system that has been trained using the datasets. The results of the research showed that the accuracy (measurement error) of the FIMT-DD algorithm follows a decreasing trend in the stream evaluation process [11]. In our previous work, we have also proposed an intelligent system architecture based on a verified police department account [12]. The authors described the system architecture and algorithm that could be used to classify the street status into the low traffic flow, medium traffic flow, or high traffic flow. The authors used a standard neural network approach, which is called Learning Vector Quantization, to train the dataset and predict the traffic flow for 30–60 min ahead of the current time.

Another study proposed a mechanism using the first deep architecture model, which included stacked autoencoders, to learn the generic traffic flow features to be used in a prediction system [13]. The results of the research showed that this method could represent latent traffic flow. A greedy layer-wise unsupervised learning algorithm was used to train the deep network, and the model parameters were tuned to improve the prediction performance. The authors' proposed method is superior to the BP NN, RW, SVM, and RBF NN models.

Xia et al. studied traffic flow prediction using a Hadoop architecture [14]. A parallel K-Nearest neighbors approach was implemented using the MapReduce mechanism, which was used to predict traffic flows. Correlation analysis was also conducted using the MapReduce platform. A real-time prediction system was developed using two key modules: offline distributed training and online parallel prediction. The result of this research showed that the measurement error of the traffic flow prediction using correlation analysis was significantly improved compared to the ARIMA, Naive Bayes, and Multilayer Perceptron. Additionally, this method provided a solution that could be scaled up because it is implemented in the Hadoop platform.

Hou and Li presented a repeatability and similarity method to predict big traffic data in China [15]. By using the repeatability and similarity of the traffic flow, the authors were able to combine the predictions of short- and long-term traffic flow forecasting. The results showed that the repeatability and similarity approach could effectively observe and predict the traffic flow of big data in China.

The review of the online evaluation of big data stream are compared to identify the models by A. Bifet [16]. The utilization of the prequential method to evaluate the result is used in this paper. The distributive regression task has been developed and tested to get a speedup of 4.7 × execution time compared to the sequential version [17]. The development of big data stream architecture for a certain area has been designed [18]. Some tools have been recommended to enable big data to be processed, such as Kafka, nimbus, zookeeper, Hadoop, and storm.

In this paper, we propose an improved method for big data stream problems. We decreased the Mean Absolute Percentage Error (MAPE) by 3% compared to our previous improvement [19], which used the Chernoff bound approach. Additionally, we decreased the MAPE by 12% compared to the standard method [3].

FIMT-DD algorithm

Currently, datasets are increasing in size. An incremental algorithm to process vast data is needed because it is impossible to store and process the whole datasets at once. The FIMT-DD algorithm works iteratively based on the instance's arrival. This algorithm decides the best split for all its attributes. It will split attributes if the splitting criterion is met. Then, the adaptation strategy will be performed if the local concept drift occurs.

The attribute selection, which is used to determine the best attribute for samples, is conducted using the Hoeffding Bound and SDR. In particular, dataset S with the size of N is introduced. Attribute A will split the data into two categories S_L and S_R with the size of NL and NR, respectively, where $S = S_L$ U S_R and $N = NL + NR$. The SDR (hA) is calculated by Eq. (1).

$$SDR(hA) = sd(S) - \frac{NL}{N}sd(S_L) - \frac{NR}{N}sd(S_R) \tag{1}$$

It can be observed in Eq. (2) that the FIMT-DD algorithm preserves the values of attributes y and y^2. We can see that the real random variable r is the ratio of the SDR values for hA and hB; and its value varies between 0 and 1, depending on if (hA) is the best split of attribute A and (hB) is the best split of attribute B.

$$\begin{gathered} sd(S) = \sqrt{\frac{1}{N}\left(\sum_{i=1}^{N}\left(yi - y\right)^2\right)} \\ sd(S) = \sqrt{\frac{1}{N}\left(\sum_{i=1}^{N}yi^2 - \frac{1}{N}\left(\sum_{i=1}^{N}yi\right)^2\right)} \end{gathered} \tag{2}$$

Then, the evaluation ratio can be obtained by Eq. (3). Each r of each stream can be represented by real numbers r_1, r_2,..., r_n. To obtain a high confidence interval of the mean random variables, the FIMT-DD uses the Hoeffding bound probability. It enables us to use 1 – δ, where the value of δ is 5%. This is the average of a random sample of N i.i.d. Variables with range R within a distance ε of the true mean.

$$r = \frac{SDR(h_B)}{SDR(h_A)} \tag{3}$$

Equation (4) can be used to calculate the value of ε.

$$\varepsilon = \sqrt{\frac{R^2 lnln\left(\frac{1}{\delta}\right)}{2N}}. \tag{4}$$

When values are observed, the value of ε continues to decrease. The sample mean will approach the true mean. In this process, the Hoeffding bound contributes to decreasing the sum of a random variable's deviation from its expected value. The FIMT-DD algorithm calculates the lower and upper bounds of the estimated sample with Eq. (5).

$$r^+ = r + \varepsilon and r^- = r - \varepsilon and r^- \le r_{true} \le r^+ \tag{5}$$

The gradient descent method is used to calculate the weight update for every instance in the stream. It uses the linear perceptron to weight the relations among the parameters. The weights are updated regularly for every arrival of new instances. It does not use the whole dataset at once to calculate the weights. To be able to obtain the output value, every weight is updated using the difference of the normalized attributes (x_i), the real value (y), the learning rate (η), and the output (o). The formula for the weight is given in Eq. (6).

$$\omega_i = \omega_i + \eta\left(o - y\right)x_i, i \ne 0 \tag{6}$$

Before the learning process, the variables are categorized and changed into binary (numerical) variables. The normalization is conducted for all of the attributes. Therefore, all of the attributes will have the same effect in the filtering process. The normalization process is conducted incrementally.

Proposed method

We propose modifying the standard forms of the FIMT-DD. We add the distance and standard deviation to the Equation. We consider the standard forms of the Hoeffding without any loss of generality [20] [21].

Hoeffding bound

The Hoeffding Bound is used in the standard FIMT-DD algorithm. It is defined as the following Equation. Let X_i, where $i = 1, 2, 3, \ldots, N$, be an independent random variable such that $PrPr(X_i \in [a_i, b_i]) = 1$. Then, for $X = \sum_{i=1}^{N} X_i$ for all $\varepsilon > 0$, we have the inequality in Eq. (7).

$$PrPr(X - E[X] \geq N\varepsilon) \leq exp\left(\frac{-2N^2\varepsilon^2}{\sum_{i=1}^{N}(b_i - a_i)^2}\right)$$

$$\leq exp\left(\frac{-2N^2\varepsilon^2}{\sum_{i=1}^{N} R^2}\right) \tag{7}$$

Assume $\sum_{i=1}^{N}(b_i - a_i)^2 = R^2$. Then, we obtain

$$PrPr(X - E[X] \geq N\varepsilon) \leq exp\left(\frac{-2N\varepsilon^2}{R^2}\right) \tag{8}$$

because $PrPr(X - E[X] \geq N\varepsilon) \leq \delta$. Thus, we can simplify Eq. (8) into

$$\delta = exp\left(\frac{-2N\varepsilon^2}{R^2}\right)$$

$$\delta = \frac{1}{exp\left(\frac{2N\varepsilon^2}{R^2}\right)}.$$

After solving for ε by taking the logarithm of both sides, we obtain the Hoeffding Bound as shown in Eq. (9).

$$\begin{aligned} lnexp\left(\frac{2N\varepsilon^2}{R^2}\right) &= ln\frac{1}{\delta} \\ \left(\frac{2N\varepsilon^2}{R^2}\right) &= ln\frac{1}{\delta}. \\ \varepsilon &= \sqrt{\left(\frac{R^2 ln\left(\frac{1}{\delta}\right)}{2N}\right)} \end{aligned} \tag{9}$$

We propose adding the values of k and m as the modified standard deviation and mean distance, respectively. Equation (14) depicts k, which is the actual value of the standard deviation d divided by the sum of the actual values $y \, for \, n$ instances.

$$k = \frac{d}{\sum_{i=1}^{N} y_i} \tag{14}$$

$$S = \sum_{i=1}^{n} x_i \tag{15}$$

$$f = \frac{s}{i} \tag{16}$$

$$m = \frac{abs\left(f - s\right)}{f}. \tag{17}$$

Based on Eq. (15), x_i is the feature of the dataset and S is the sum of the value of the feature in one instance. In Eq. (16), we obtained f from the sum of features value S and divide it by i, where i is the number of features. m is calculated based on the absolute value difference between f and S divided by f.

Therefore, we modify Eq. (5) as Eq. (19). It is modified with variable k from Eq. (14) and m from Eq. (17).

$$r^+ = r + mk\varepsilon andr^- = r - mk\varepsilon andr^- \leq r_{true} \leq r^+ \tag{19}$$

Where ε is the value of the Hoeffding bound.

Results and discussions

In this research, we assessed our approach using three datasets that contain large numbers of instances. The first dataset is a traffic demand dataset, the second dataset is power system data, and the third dataset is water absorption in Chicago. We evaluate and compare our approach with the standard FIMT-DD algorithm [3], our previous improvement of the FIMT-DD Chernoff [19], and the current approach (Distance Improvement). According to the evaluation metrics (MAE, RMSE, and MAPE), our approach gives consistently lower errors compared to previous methods.

The traffic demand data were obtained from the Grab challenge. The goal of this challenge was to predict the order demand at a specific time and in a specific area. The features that are used to predict the traffic demand are the location, which is in the form of geocoding (location); day; hour; and minute. All locations have been masked to protect user privacy, and the traffic demand values have been normalized from 0 to 1 [22]. The number of instances in this dataset is 4,206,332.

The second dataset that we used is the power system dataset provided by the Open Power System Data (OPSD) [23]. The dataset used in this research are Tennet's wind power plant dataset from Germany, which consisted of wind power generation data. These data were gathered from 2005 until 2018. The size of the dataset is 21.45 MB. The dataset contains 435,268 instances. The dataset has 9 attributes, which describe onshore and offshore wind plant's actual power generation, and also its forecasted figures over 15 min time intervals.

The third dataset is obtained from sensors that were mounted by the Chicago government to measure water absorption from roads and sidewalks [24]. These data can be used to measure the development of the green infrastructure against flooding in the city of Chicago. The sensors also include sensors to obtain weather information. Each row of data represents the measurement results of the sensors for each time, location, and type of measurement. The number of instances or rows in these data is 31,642,635 rows. The size of this data set is approximately 3.7 GB. We convert timestamps to the day of the week format, and we use the period, which is the same as the road weather dataset and Traffic Demand Dataset.

We measured the errors from those datasets for every 100,000 instances for the traffic demand data, every 5000 instances for the power system data, and every 100,000 instances for the infrastructure monitoring data. These measurement points are based on the number of instances for each dataset to obtain the best graph visualization. The evaluation metrics for the measurement error that we used are the Mean Absolute Error (MAE), the Root Mean Square Error (RMSE), and the Mean Absolute Percentage Error (MAPE), as described by Eqs. (16), (17), and (18), respectively.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|f_i - y_i\right| \tag{16}$$

Mean Absolute Error (MAE)

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left|f_i - y_i\right|^2} \tag{17}$$

Root Mean Squared Error (RMSE)

$$MAPE = \frac{1}{N}x100\%\sum_{i=1}^{N}\left|\frac{f_i - y_i}{y_i}\right| \tag{18}$$

Mean Absolute Percentage Error (MAPE)

Here, f_i is the predicted values, y_i is the real values, and N is the amount of data in each stream.

The specifications of the computer that we used for the simulations are an Intel(R) Core(TM) i7-6800 K CPU @ 3.4 GHz, 32 GB of RAM, and a 2 TB hard disk drive. We modified the code of the FIMT-DD algorithm from the Massive Online Analysis (MOA) application. The simulation is conducted on top of the MOA application [25]. The information that we measured from the simulation is the measurement errors (MAE, RMSE, and MAPE).

In this research, the measured MAEs for those three datasets are described in Fig. 1 (Traffic Demand Dataset), Fig. 2 (Power System Dataset) and Fig. 3 (Road Weather Dataset). The STD is used as the identifier for the results that are obtained by using the standard FIMT-DD algorithm, IM is used as the identifier for the Chernoff-Bound approach [19], and ED is the identifier for our new approach (distance value).

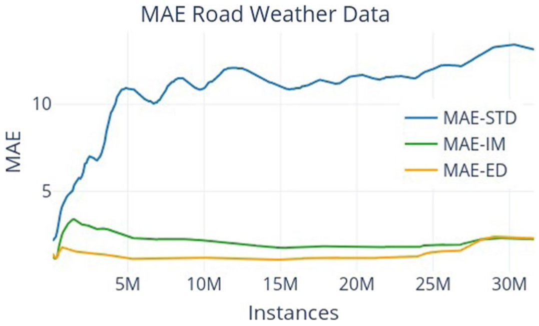

Fig. 1 MAEs of the road weather data

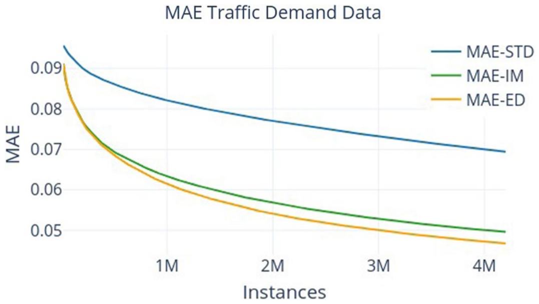

Fig. 2 MAE comparison of the traffic demand data

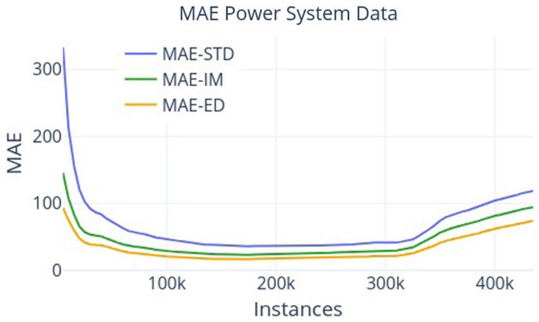

Fig. 3 MAE comparison of the power system data

Based on the results of our experiment, our approach can increase the accuracy and lower the error of the FIMT-DD algorithm. In the first simulation using the Traffic Demand Dataset, the Hoeffding bound MAE-STD is 0.069 at the 4,200,000th instance, whereas MAE-IM is 0.049 and MAE-ED is 0.046. Similarly, in the second simulation using Tennet's wind power plant dataset, the maximum MAE-STD is 118.88 at the 435,000th instance, whereas the MAE-IM and MAE-ED are 94.51 and 74.18, respectively. Moreover, for the simulation using the road weather dataset, MAE-STD is 10.90 at the 5,000,000th instance, whereas MAE-IM and MAE-ED are 2.38 and 1.12, respectively. The MAE-ED for the three datasets is lower than the MAE-STD and MAE-IM in every stream evaluation.

In this research, the measured RMSEs for those three datasets are described in Fig. 4 (Traffic Demand Dataset), Fig. 5 (Power System Dataset) and Fig. 6. (Road Weather Data). In the Traffic Demand Dataset simulation, RMSE-STD is 0.12 at the 4,200,000th instance, whereas RMSE-IM is 0.088 and RMSE-ED is 0.085. The Tennet's wind power plant dataset simulation results in an RMSE-STD of 380.51 at the 435,000th instance, whereas RMSE-IM is 237.03 and RMSE-ED is 165.91. Moreover, for the simulation by using the Road weather dataset, RMSE-STD is 23.13 at the 31,600,000th instance, whereas RMSE-IM is 14.41 and RMSE-IM is 9.42. Our approach produces a lower RMSE-ED compared to RMSE-STD and RMSE-IM in every stream evaluation.

The measured MAPEs for the three datasets described in Fig. 7 (Traffic Demand Dataset), Fig. 8 (Power System Dataset) and Fig. 9 (Road Weather Dataset). In the Traffic Demand Dataset experiment, MAPE-STD is 1332% at the 4,200,000th instance,

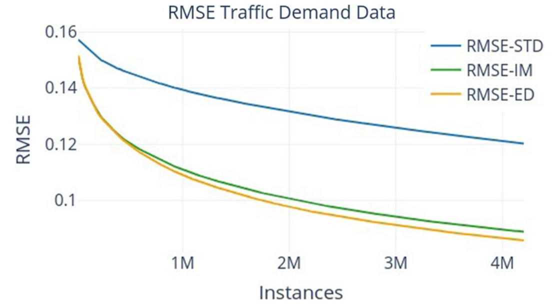

Fig. 4 RMSE comparison for the traffic demand data

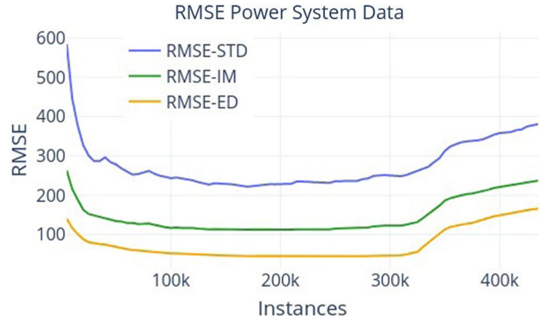

Fig. 5 RMSE comparison for the power system data

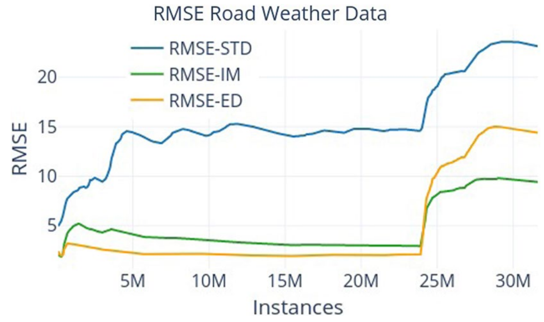

Fig. 6 RMSEs of road weather data

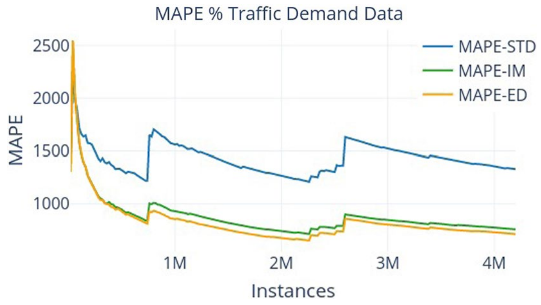

Fig. 7 MAPE comparison of the traffic demanddData

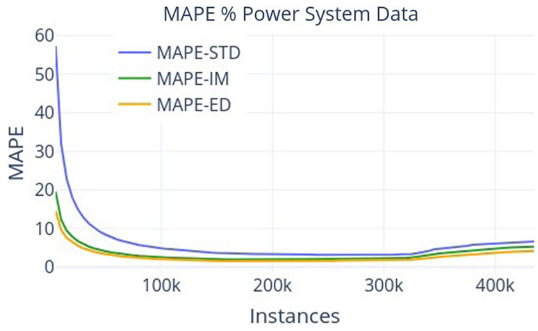

Fig. 8 MAPE comparison of the power system data

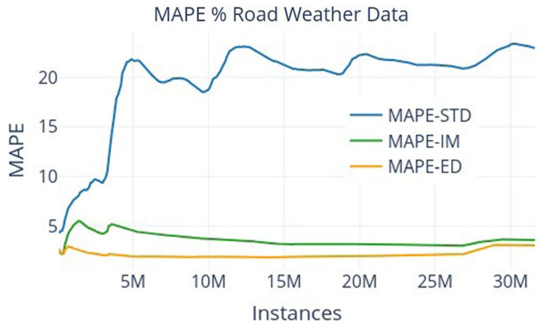

Fig. 9 MAPEs of the road weather data

whereas MAPE-IM is 761%, and MAPE-ED is 714%. Additionally, for the Tennet's wind power plant dataset simulation, MAPE-STD is 6.62% at the 435,000th instance, whereas MAPE-IM is 5.29%, and MAPE-ED is 4.13%. In addition, for the simulation using the Road Weather Dataset, MAPE-STD is 22.97% at the 31,600,00th, for instance, whereas MAPE-IM is 3.57% and MAPE-ED is 3.03%. MAPE-ED results in a lower MAPE than the MAPE-STD and MAPE-IM in every stream evaluation. Based on the MAPE result, our proposed MAPE-ED gives a lower MAPE compared to MAPE-STD and MAPE-IM.

The differences in MAPE-ED compared to MAPE-IM are 47% for the traffic demand data, 1.16% for the power system data, and 0.54% for the road weather data.

Comparing our approach with the standard method, the differences in MAPE-ED compared to MAPE-STD are 618% for the traffic demand data, 2.49% for the power system data, and 19.65% for the road weather data. We use the MAPE to measure the error performance in the form of a percentage error. As seen in Figs. 7, 8, and 9, the overall MAPE measurements from those three datasets show that our proposed method can reduce the error percentage in every stream's evaluation process.

The comparison summary of MAEs, RMSEs, and MAPEs value of the standard method (STD), Chernoff-bound approach, and our improvement (distance variable) are described in Table 1. Based on Table 1 result, ED approach gives smaller errors compared to IM and STD method. Also, the evaluation of the real value, predicted value and the measurement error show that the distance means approach can accelerate the learning rate because it causes the tree to split more often in the early stage of the learning process.

Conclusion

The FIMT-DD algorithm is a data mining method that enables us to perform data stream evaluations. The standard FIMT-DD algorithm uses the Hoeffding bound method for its split criterion process. In this study, we evaluate and analyze the Distance Mean and Standard Deviation approach for the FIMT-DD algorithm. We evaluate using three big time-series datasets, namely, the Traffic Demand Dataset, Tennet's wind plant power generation dataset, and the Road Weather Dataset. In all simulations, our proposed approach of the FIMT-DD algorithm can consistently lower the error in every step of the learning process compared to the standard method and Chernoff method approach. Based on the experiments that we have conducted and the measurement errors that are produced, all measurement errors (MAE, RMSE, and MAPE) show that our approach has lower measurement errors compared to the previous approaches (Chernoff Bound) and the standard method. Our approach (distance mean) contributes by lowering the MAPE in every stage of learning by approximately 2.49% compared to the Chernoff Bound Method and 19.65% compared to the standard method. In the future, we plan

Table 1 Summary of MAE, RMSE, and MAPE result

No	Dataset	STD	IM	ED
MAE				
1	Road weather data	13.15	2.23	2.29
2	Traffic demand data	0.06	0.04	0.04
3	Power system data	118.88	94.51	74.18
RMSE				
1	Road weather data	23.13	9.42	14.41
2	Traffic demand data	0.12	0.08	0.08
3	Power system data	380.51	237.03	165.91
MAPE				
1	Road weather data	22.97%	3.57%	3.03%
2	Traffic demand data	1329.7%	759.30%	714.43%
3	Power system data	6.62%	5.29%	4.13%

to optimize and determine which bound is appropriate to be used for certain streams of data.

Abbreviation

FIMT-DD: Fast Incremental Model Tree with Drift Detection; MAE: Mean Absolute Error; MAPE: Mean Absolute Percentage Error; RMSE: Root Mean Square Error; OPSD: Open Power System Data; SDR: Standard Deviation Reduction.

Acknowledgements

We would like to express our gratitude for the grant received from Universitas Indonesia.

Authors' contributions

AW: Contributed the idea of the distance variable improvement, implemented the coding, created the simulation scenarios, measured the dataset simulations, revised the introduction and methods, added datasets, and revised the results & discussions. PM: Verified the experiment process, the complied data and the consistency of derived formula application; and revised the results, analysis and discussion sections. JA: Evaluated the model and verified the algorithm. WDWTB: Prepared and cleansed the dataset. MIR, LMH, VFA: Wrote the introduction, related work, and results of this paper. All authors read and approved the final manuscript.

Authors' information

Ari Wibisono was born in Jakarta, December 27, 1988. Now, he works as a lecturer with the Faculty of Computer Science at Universitas Indonesia. He received his Master of Computer Science degree in 2012 and Bachelor's Degree in 2010 from the Faculty of Computer Science, Universitas Indonesia. The author's specific fields of interest are System Programming, Intelligent Systems, and High-Performance Computing.

Petrus Mursanto was born in Surakarta, June 25, 1967. He has been working as a senior lecturer with the Faculty of Computer Science at Universitas Indonesia since 1992. He received his Doctoral degree from the Faculty of Computer Science at Universitas Indonesia. He obtained his Master's degree in Computer Science from the University of Auckland in 1999 and his Bachelor's degree in Electrical Engineering from Universitas Indonesia in 1992. The author's fields of expertise are software engineering, reconfigurable computing, and digital technique design.

Jihan Adibah, Wendy D. W. T. Bayu, May Iffah Rizki, Lintang Matahari Hasani, and Valian Fil Ahli are students with the Faculty of Computer Science at Universitas Indonesia.

Funding

Universitas Indonesia (2020)

Availability of data and materials

https://drive.google.com/drive/folders/16Uz4qOA5LDAX2794sgD5xGQh9Dsy_Lz4?usp=sharing.

Competing interests

The authors declare that they have no competing interests.

Received: 28 April 2020 Accepted: 14 September 2020
Published online: 13 October 2020

References

1. Adak, M. Fatih, and Mustafa Akpinar. 2018. A hybrid artificial bee colony algorithm using multiple linear regression on time-series datasets.
2. Aghabozorgi S, Wah TY. Clustering of Large Time Series Datasets. Intelligent Data Analysis. 2014;18(5):793–817. https://doi.org/10.3233/ida-140669.
3. Ikonomovska E, Gama J, Džeroski S. Learning model trees from evolving data streams. Data Min Knowl Disc. 2011;23(1):128–68.
4. Hoeffding W. Probability Inequalities for Sums of Bounded Random Variables. Journal of the American Statistical Association. 1963;58(301):13–30.
5. Wibisono, A., Wisesa, H.A., Jatmiko, W., Mursanto, P., Sarwinda, D. 2016. Perceptron rule improvement on FIMT-DD for large traffic data stream. In: Proceedings of the International Joint Conference on Neural Networks. 2016; 5161–7.
6. Zhang, C., Bennett-type generalization bounds: Large-deviation case and faster rate of convergence. 2013. In: Uncertainty in Artificial Intelligence - Proceedings of the 29th Conference UAI 2013. 2013; 714–22.
7. Beygelzimer, A., Langford, J., Lifshits, Y., Sorkin, G., Strehl, A. 2009. Conditional probability tree estimation analysis and algorithms. In: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence UAI. 2009; 51–8.
8. Balsubramani, A., Ramdas, A. 2016. Sequential nonparametric testing with the law of the iterated logarithm. 32nd Conference on Uncertainty in Artificial Intelligence 2016 UAI 2016. 42-51.
9. Koesdwiady A, Soua R, Karray F. Improving traffic flow prediction with weather information in connected cars: a deep learning approach. IEEE Trans Veh Technol. 2016;65(12):9508–17.
10. Soua, R., Koesdwiady, A., Karray, F. 2016. Big-data-generated traffic flow prediction using deep learning and dempster-shafer theory. In: Proceedings of the International Joint Conference on Neural Networks. 2016; 3195–202.
11. Wibisono A, Jatmiko W, Wisesa HA, Hardjono B, Mursanto P. Traffic big data prediction and visualization using Fast Incremental Model Trees-Drift Detection (FIMT-DD). Knowl-Based Syst. 2016;93:33–46.

12. Wibisono, A., Sina, I., Ihsannuddin, M.A., Hafizh, A., Hardjono, B., Nurhadiyatna, A., Jatmiko, W., Mursanto,.P. 2012. Traffic intelligent system architecture based on social media information, International Conference on Advanced Computer Science and Information Systems, ICACSIS. 2012; 25–30.
13. Y. Lv, Y. Duan, W. Kang, Z. Li and F. Y. Wang. 2015. Traffic Flow Prediction with Big Data: A Deep Learning Approach. In: IEEE Transactions on Intelligent Transportation Systems. vol. 16, p. 865–73.
14. Xia D, Li H, Wang B, Li Y, Zhang Z. A map reduce-based nearest neighbor approach for big-data-driven traffic flow prediction. IEEE Access. 2016;2016:2920–34.
15. Hou Z, Li X. Repeatability and Similarity of Freeway Traffic Flow and Long-Term Prediction Under Big Data. In: IEEE Transactions on Intelligent Transportation Systems. 2016; 1786–96.
16. Bifet A, et al. Efficient online evaluation of big data stream classifiers. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. 2015.
17. Vu AT, et al. Distributed adaptive model rules for mining big data streams. In: 2014 IEEE International Conference on Big Data (Big Data). IEEE, 2014.
18. Ta V-D, Chuan-Ming L, Goodwill WN. Big data stream computing in healthcare real-time analytics. In: 2016 IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA). IEEE, 2016.
19. Wibisono A, Sarwinda D, Mursanto P. Tree stream mining algorithm with Chernoff-bound and standard deviation approach for big data stream. Journal of Big Data. 2019;6:1.
20. Phillips, J.M. 2012. Chernoff-hoeffding inequality and applications. arXiv preprint arXiv:1209.6396. 2012 Sep 27.
21. Y. Lv, Y. Duan, W. Kang, Z. Li and F. Y. Wang. 2015. Traffic Flow Prediction with Big Data: A Deep Learning Approach. In: IEEE Transactions on Intelligent Transportation Systems. 16, 2 (April 2015), 865–73.
22. Grab, Traffic Management| Grab AI, https://www.aiforsea.com/traffic-management.
23. Open Power System Data (OPSD). (2018). Data Platform: Renewable Power Plants. https://data.open-power-system-data.org/renewable_power_plants/.
24. Smart Green Infrastructure Monitoring Sensors - Historical, https://data.cityofchicago.org/Environment-Sustainable-Development/Smart-Green-Infrastructure-Monitoring-Sensors-Hist/ggws-77ih, US-Department of Transportation-Seattle, Accessed 5 Apr 2019.
25. Bifet A, Holmes G, Kirkby R, Pfahringer B. MOA: massive Online Analysis. J Mach Learn Res. 2010;11:1601–4.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.