Understanding Elapsed-time Sampling Delayed Feedback

Hanlang Zhao[1,a], Yiyun Quan[2,b*], Hao Yu[3,c], Yongqi Wu[4,d], Zhengyuan Liu[5,e]

[a] arthurhanlang@gmail.com, [b] yiyunq@uci.edu, [c] 625566612@qq.com, [d] 1294332658@qq.com, [e] Jackonchina@gmail.com

[1]Dulwich College Suzhou, Jiangsu, 215000, China

[2]University of California Irvine, Irvine, 92612, United States

[3]Lancaster University, Lancaster, LA1 4YL, UK

[4]Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, 710048, China

[5]Rancho Cucamonga High School, Rancho Cucamonga, 91701, US

Abstract. ES-DFM is the model proposed recently to improve algorithmic performance in conversion rate in an E-commerce recommendation system. The model addresses the delayed feedback issue, which is a cutting-edge issue in terms of two core evaluation indices within a good recommendation system – user likeability and user behavior. Adding in data of false negatives, the model better weighed between the waiting time before updating the model training data and the freshness of the training data. In our research, we replicated the baselines – DFM, FNC, FNW, FSIW, ESDFM by trying on Google Co-lab and rented server. Results show that the codes run successfully. Then we experimented on the parameters of the ES-DFM model, in hope of optimizing the results even more. However, the change in parameters returned equally good performance but longer processing time.

Keywords: delayed feedback, ES-DFM model

1. Introduction

ES-DFM is the model proposed recently to improve algorithmic performance in conversion rate in E-commerce recommendation system. The model addresses the delayed feedback issue, which is a cutting-edge issue in terms of two core evaluation indices within a good recommendation system – user likeability and user behavior. Adding in data of false negatives, the model better weighed between the waiting time before updating the model training data and the freshness of the training data. In our research, we replicated the baselines – DFM, FNC, FNW, FSIW, ESDFM by trying on Google Co-lab and rented server. Results show that the codes run successfully.

ICIDC 2023, June 02-04, Nanchang, People's Republic of China

DOI 10.4108/eai.2-6-2023.2334607

2. Background

A recommendation system, according to Leskovec, Rajaraman, and Ullman, is a facility, "an extensive class of Web applications that involve predicting user responses to options" (319) [1]. A good recommendation system has five indices: user data, funnel effect, consumer satisfaction, off-line evaluation and on-line evaluation. Our research is cutting-edge in terms of user likeability in user data construction and user behavior in on-line evaluation.

To designate user likeability and user behavior in on-line retailing, the commonly referred indicator is the conversion rate (CVR). Conversion is the act or process of changing something from one form, use or system to another. In our research, conversion is the change from the act of clicking on an item in an on-line retailer to the act of payment or buying. Conversion rate (CVR) is the ration of conversions divided by clicks, speaking of the same item. Due to the fact that conversion can bring profits, the research on conversion rate is now paid a lot of attention both academically and industrially. More and more people are involved in research in this area. Many machine jobs are slowly starting to use it.

Since the data changes very quickly, in order to capture the dynamic changes of user needs, it is necessary for the business system to update the latest data in a timely manner. The learned model is then updated with the latest data to ensure correct results. However, in on-line shopping, conversions usually do not happen immediately after user clicks, and that further complicates the conversion prediction problem. Hence the delayed feedback issue of whether it is better to wait for a long enough time before updating the training data for the model or to maintain short waiting time to keep the training data freshness.

3. Motivation

In order to solve the problem of delayed feedback, Chapelle proposed a delayed feedback model (DFM) to optimize the conversion rate as a joint probability to predict the conversion rate and delay time distribution [2]. The joint probability is estimated over the observed time interval and may deviate from the true transformation distribution; Biased conversion rates can be even more inaccurate due to delayed feedback issues in online learning Settings.

Ktena et al. proposed False Negative Weighting (FNW) methods where each arriving instance is first marked as negative and then corrected at a later time transition [3]. The model updates the model in near real-time, but each fake negative instance can have side effects on the learned model until it is corrected. This side effect is magnified if the data distribution changes frequently. For example, there may be a sharp increase in user clicks at the start of a promotional campaign, with most conversions occurring after a certain amount of time. Furthermore, this overwhelming false negative could harm predictive models.

Yasui et al. proposed a Feedback Shift Importance Weighting (FSIW) algorithm, where the model waits long enough for a true switch, but it does not allow data correction even if a switch event occurs afterward [4]. The authors argue that positive examples are important for delayed feedback prediction because positive examples are always scarcer than negative examples, but FSIW may lack model freshness due to long waits.

This time focusing on the Elapsed-Time Sampling Delayed Feedback Model, which is a slightly improved model for streaming conversion rate prediction. In the original paper, the authors proposed some rigorous streaming training and testing experimental protocols to better align with real industrial applications. They optimized the expectation of the true conversion distribution by importance sampling under the elapsed time sampling distribution (Yang et al., 2020).

4. Methods

4.1 DFM

When the topic of delayed feedback was started to be studied, the first proposed model was the Delayed Feedback Model (DFM). This is the first model to introduce the problem of delayed feedback into recommender systems. This provides a theoretical framework for future research and lays a good foundation for future research. DFM will use x as the characteristics of the sample. C represents whether the sample will eventually transform (C = 0 means the sample will not transform and C = 1 means the sample will transform). Y represents whether the sample will transform within the observed time. D represents conversion delay time. E represents the observation time. On this basis, the following formula is derived.

$$P(Y = 1|x) = P(C = 1|x) \; (1)$$

$$P(Y = 0|x) = P(C = 0|x) + P(C = 1|E < D) \; (2)$$

Equation 1 means the probability the sample will transform within the observed time equals to the probability the sample will transform eventually. Equation 2 means the probability the sample will not transform within the observed time equals to the probability the sample will not transform eventually add the probability the sample will transform but the transformation time is longer than the observed time.

4.2 NoDeF

On the basis of DFM, NoDeF was proposed. It presents the distribution of delayed time by defining the mode of weighted kernel function without any parametric assumptions, so that the model can adapt to more complex scenarios. However, this model also has its own problem, which is the positive samples may be lost.

4.3 FSIW

Feedback shift importance weighting method(FSIW) is a method based on importance sampling (IS). IS is to estimate the ideal loss of each sample by constructing the mathematical relationship between the observed distribution and the real distribution, which is the following formula:

$$\begin{aligned} L_{ideal} &= E_{(x,y)\sim p(x,y)} l(y, f_\theta(x)) \\ &= \int p(x)dx \int p(y|x) l(y, f_\theta(x))dy \\ &= \int p(x)dx \int q(y|x)\frac{p(y|x)}{q(y|x)} l(y, f_\theta(x))dy \end{aligned}$$

$$\approx \sum_{(x_i,y_i \in D)} [y_i \frac{p(y_i = 1|x)}{q(y_i = 1|x)} \log(f_\theta(x_i)) + (1 - y_i) \frac{p(y_i = 0|x)}{q(y_i = 0|x)} \log(1 - f_\theta(x_i))]$$

Where l is the loss function, y is the observed sample, p is the real sample distribution, and q is the observed sample distribution.

Through importance sampling, FSIW defines the variable S. If S=1, the observed sample Y is consistent with the real sample C. And the ratio of the real sample distribution to the predicted distribution has the following relationship:

$$\frac{P(C = 1|X = x)}{P(Y = 1|X = x)} = \frac{1}{P(S = 1|C = 1, X = x)} \quad (3)$$

$$\frac{P(C = 0|X = x)}{P(Y = 0|X = x)} = 1 - \frac{P(S = 0, C = 1|X = x)}{P(Y = 0|X = x)} \quad (4)$$

Equation 3 represents the probability that both the real sample and the observed sample are positive. Equation 4 expresses the probability that both the real sample and the observed sample are negative.

Based on the importance sampling, this algorithm has a positive performance in effect. However, this algorithm also has its disadvantages, which is not considering the reflow of delayed feedback samples and reuse them again, which will result in the loss of some positive samples.

4.4 FNW

Fake negative weighted method (FNW) was proposed in 2019. This is also based on importance sampling, but has been improved for the shortcomings of FSIW. FNW takes sample reflow into account with delayed feedback. In this method, all samples are initially negative, and after positive samples are observed, these samples become positive. The relationship between the observed distribution b and the real distribution p will have the following equations:

$$b(y = 1|x) = \frac{p(y = 1|x)}{1 + p(y = 1|x)}$$

$$b(y = 0|x) = \frac{1}{1 + p(y = 1|x)}$$

Since all samples are regarded as negative samples at the beginning, and changing them to positive when positive samples are observed will cause positive samples to be reused once. It can also be seen in the equations that the total sample size becomes $1+p(y=1|x)$. This is the display of the positive sample being repeated. Although this method optimizes the shortcomings of FSIW, it will also introduce many false-negatives to the model.

4.5 ES-DFM

This model is further optimized on the basis of FNW. As mentioned above, FNW repeated all positive samples, which led to introducing many false negative samples. ES-DFM only reuses the positive samples of delayed feedback on its basis. The samples that are positive during the observation time will not be initialized as negative at the beginning, which can reduce the introduction false negative samples.

In this model, firstly, assuming the waiting time as e and e is an indeterminate value. x is a characteristic of the item, such as the price of the item and so on. For instance, users need to consider more time when purchasing high-priced items, so long waiting times are required. Therefore, e should depend on x. In order to realize flexible control on the waiting time, the elapsed time is drawn from an elapsed time distribution p(e|x). For p(e|x), there are three different cases: real negative, fake negative and positive. For real negatives, no matter how long we wait, there is no feedback. And fake negative means that there is a conversion after the observation time, because it exceeds the observation time, so in fact, this is also classified as negative in another situation. For the positive case, feedback can be received within the observation time. In these two cases, through the conversion distribution q(y|x), the following formula can be written:

$$q(y=0)=\frac{p(y=0)+p(y=1)p(h>e|y=1)}{1+p(y=1)p(h>e|y=1)} \quad (5)$$

$$q(y=1)=\frac{p(y=1)}{1+p(y=1)p(h>e|y=1)} \quad (6)$$

Here h represents the time receives the feedback and e represents the time waits for. Equation 5 means the conversion rate of y equals to 0 which is the real negative case and the fake negative case. Equation 6 means the conversion rate of y equals to 1 which is the positive case. To obtain unbiased CVR estimation in delayed feedback problem, optimizing the expectation of p(y|x) via important samplings is needed. Since the delayed positive part is very small, it's reasonable to assume p(x) is approximately equal to q(x).

$$\int p(x)dx \int q(y|x)\frac{p(y|x)}{q(y|x)} l(y, f_\theta(x))dy \quad (7)$$

According to the equation 7, the ideal target can be optimized, and the appropriate weight w (x, y) = p (y | x)/q (y | x). w(x, y) can be decomposed into two parts: pdp(x) and prn(x). pdp(x) is the delay positive probability, representing the probability that a sample is repeatedly positive; prn(x) is the true negative probability, representing the probability that the observed negative probability will not be converted. The importance weighted CVR loss function is the following formula.

$$L_{iw}^{n}=-\sum_{(x_i,y_i)\in D}^{n} y_i[1+p_{dp}(x_i)]\log(f_\theta(x_i))+(1-y_i)[1+p_{dp}(x_i)]p_{rn}(x_i)\log(1-f_\theta(x_i))$$

The importance weighted loss function is unbiased because they are using ideal value pdp and prn. But in actual situations, bias will always exist, so they need to use estimated fdp and frn instead of ideal value pdp and prn. In this case, the predicted probability f(x) converges to this formula at the bottom of the slide.

Since the bias always exists, p(e|x) can be used to control the bias. If e is very large, then p(h>e) will become very small. This means there are few fake negative cases. Therefore, controlling the waiting time distribution p(e|x) can reduce bias. And this is also the missing part of the existing method.

4.6 Cox Model

The Cox model assumes that the hazard function consists of two non-negative functions: a baseline hazard function and a risk score, defined as the effect of an individual's observed covariate on baseline risk. They represent h(x) as a function of long run risk. Assume that the risk function has the form in the following formula.

$$\lambda(t|x) = \lambda_0(t) * e^{h(x)}$$

5. Datasets

The dataset used is an offline dataset called the Criteo dataset. It is used in Chapelle (2014). This dataset is formed from 60 days of real-time traffic data corresponding to post-click conversions. It includes timestamps for clicks and conversions. And each sample is described by a set of hashed categorical features and some continuous features.

5.1 Data Pre-processing

In the previous paper, it is indicated that the datasets are evenly divided into two parts: pretraining part and testing part.

Firstly, as shown in figure 1, taking ES-DFM as an example. During the pretrain action, the computer is actually doing both training and testing action are executing. This first part, from 0 to 30, is used to train the data and get a trained model. While the second part, from 30 to 60, is used to test the trained model.

```
"""es-dfm"""
cut_hour = parse_float_arg(name, "cut_hour")
cut_sec = int(SECONDS_AN_HOUR*cut_hour)
train_data = data.sub_days(0, 30).shuffle()
train_label_tn = np.reshape(train_data.pay_ts < 0, (-1, 1))
train_label_dp = np.reshape(
    train_data.pay_ts - train_data.click_ts > cut_sec, (-1, 1))
train_label = np.reshape(train_data.pay_ts > 0, (-1, 1))
train_data.labels = np.concatenate(
    [train_label_tn, train_label_dp, train_label], axis=1)
"""true negative+delayed positive label, train label(0,1)"""
"""multitasks:true negative, delayed positive, true label?"""
test_data = data.sub_days(30, 60)
test_label_tn = np.reshape(test_data.pay_ts < 0, (-1, 1))
test_label_dp = np.reshape(
    test_data.pay_ts - test_data.click_ts > cut_sec, (-1, 1))
test_label = np.reshape(test_data.pay_ts > 0, (-1, 1))
test_data.labels = np.concatenate(
    [test_label_tn, test_label_dp, test_label], axis=1)
```

Figure 1. The code of training the ES-DFM model by splitting the datasets into two parts

5.2 Evaluation Metrics

To test whether ES-DFM helps to increase the efficiency while dealing with data stream, focusing on three main indexes is needed: the auc(area under ROC), prauc(area under the precision-recall curve) and nll(negative log likelihood).

Firstly, AUC is the area of ROC, the value of area will not excess of 1, in which the more area it close to 1, the more real data is. PRAUC is quite similar with AUC which is also used to determine the reality of the data. NLL, negative log likelihood, is a probability distribution that used to predict the reality of the data.

5.3 Choice of p(e|x)

In the reference paper, p(e|x) is defined based on expert knowledge and the bias analysis described above. When buying an expensive product, the customer will usually think for more time, and thus cost for a longer waiting time. However, the information about the price is unknown since the data is anonymized. To balance the model, they make p(e=c|x)=1(c is a constant), which made a Dirac distribution that could balance the price and improve the efficiency. However, it could be a vital mistake since every product have different price, it is not very precise to just use one constant to represent a price with a huge distribution. The research is looking forward to replace this with cox model which will improve the efficiency of the model.

5.4 Replication of code run through

We could get access to only one of the two datasets used in Yang et al for replicating the process of ES-DFM model code run through, Fortunately, as shown in figure 2, we had the model run through [5].

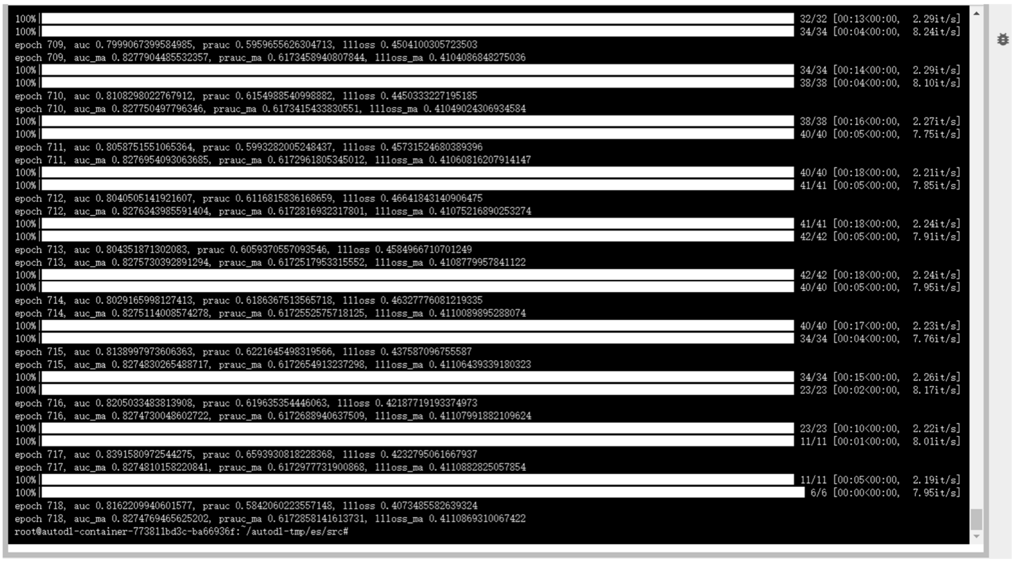


Figure 2. Open source code run through

6. Experiment

Our experiment tried to optimize the ES-DFM model by adding in the mathematical thinking of the Cox model. The idea is to suppose ourselves as shop owners. Whenever a customer walks in a shop, details of his/her dress, manner, accent, tone of speaking, or whether s/he has a company, help the shop owners to judge on whether and when the customer will make a purchase, just like the way the Cox model describes the process of doctors making a judgement on whether a patient of cancer will survive a particular date or year – multiple factors to be considered in.

The experiment deepened the ES-DFM neuro-network layers by one, in order to simulate that human brain takes one more factor into consideration before making a judgement, so the new ES-DFM model has four-layer neuro-network. Then experimenting on setting the number of nodes in each layer from 256, 256, 128 for three layers to 256, 256, 128, 128 for four layers and 128, 128, 64, 64 for four layers. For the sake of comprehensive comparison, we also experimented on 128, 128, 64 nodes for three layers.

A sever with 15 vCPU Intel(R) Xeon(R) Platinum 8358P CPU, 2.6 GHz, GPU NVIDIA GeForce RTX 3080(10GB), environment TensorFlow 2.9.0, Python 3.8(ubuntu 20.04), Cuda 11.2, memory 80G, hard disk 20G, solid state disk 50G is rented.

Judging from the tables in appendices 1, 2 and 3, comparisons of the ES-DFM model and every one of the three other models showed little difference. Each pair overlapped almost perfectly except obvious difference in the beginning epochs, indicating that, for the customers to make a purchase, whether they take into consideration three factors or four factors does not matter much so long as the ES-DFM model is applied. However, the running time of the four models vary greatly. Running time of the ES-DFM model, with 256, 256, 128 nodes, is about 50 minutes. Running time of our models with four layers of 256, 256, 128, 128 nodes is about 62 minutes, 128, 128, 64, 64 about 61 minutes, and 128, 128, 64 nodes 71 minutes.

7. Conclusion

We have run the ES-DFM model successfully, and we made further exploration by adding in a layer of neural network in the codes for ES-DFM model. Results showed that adding in one layer did not bring much improvement to the ES-DFM model performance, but provided evidence that the ES-DFM model is superior because of its shorter running time.

References

[1] Leskovec, Jure, Anand Rajaraman, and Jeffrey David Ullman. Mining of Massive Database. 3rd ed. Cambridge UP, 2020.

[2] Chapelle, O., Modeling delayed feedback in display advertising. In KDD, 1097–1105. ACM. 2014.

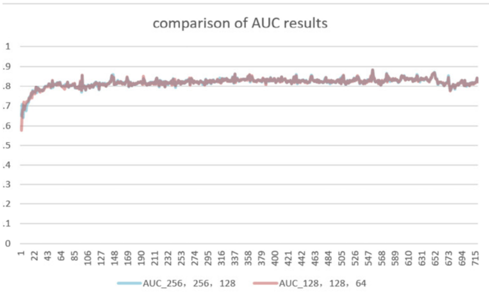
[3] Ktena, S. I., et al., Addressing delayed feedback for continuous training with neural networks in CTR prediction. In RecSys, 187–195. ACM. 2019.

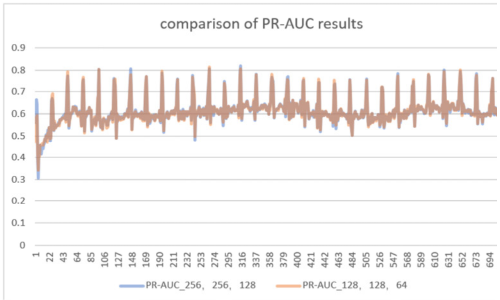
[4] Yasui, S. et al., A Feedback Shift Correction in Predicting Conversion Rates under Delayed Feedback. In WWW '20: The Web Conference 2020, 2740–2746. ACM / IW3C2. 2020.

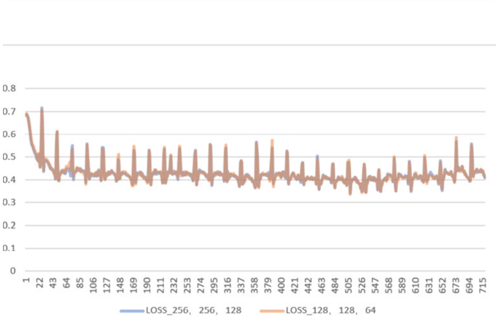
[5] Yang, Jiaqi et al., "Capturing Delayed Feedback in Conversion Rate Prediction Via Elapsed-Time Sampling", arXiv.org, 6 Dec. 2020, arxiv.org/abs/2012.03245v4.

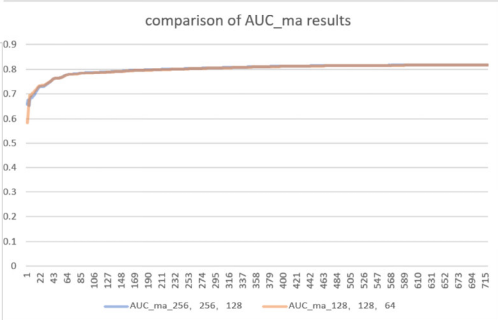
Appendix

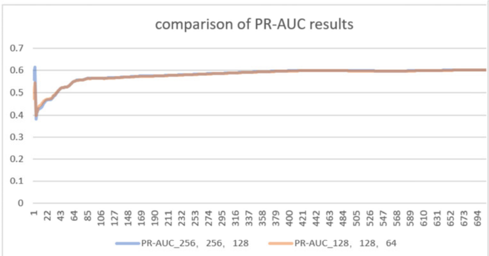
Appendix-1: comparison of results by the ES-DFM model (256, 256, 128) and a new model (128, 128, 64)

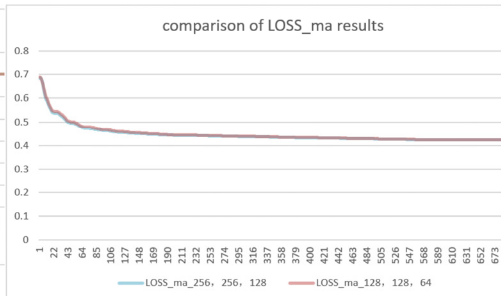




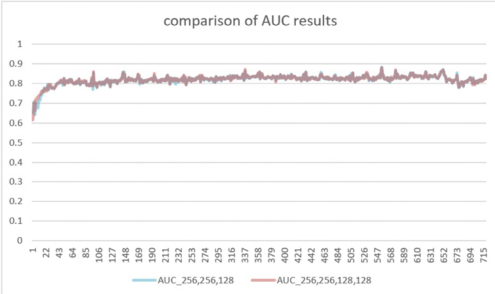


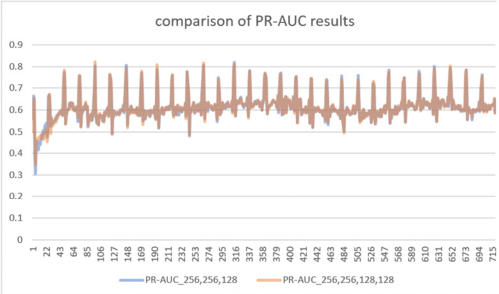


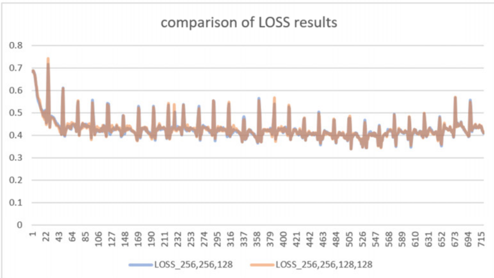


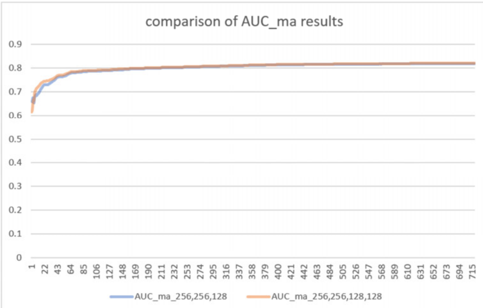


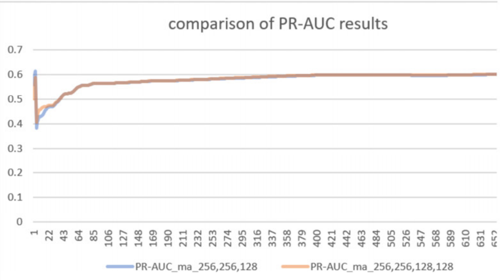
Appendix-2: comparison of results by the ES-DFM model (256, 256, 128) and a new model (256, 256, 128, 128)

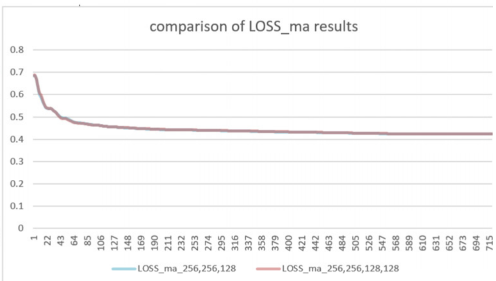












Appendix-3: comparison of results by the ES-DFM model (256, 256, 128) and a new model (128, 128, 64, 64)