2021 International Conference on Computer Communication and Artificial Intelligence (CCAI) | 978-1-7281-9401-1/20/$31.00 ©2021 IEEE | DOI: 10.1109/CCAI50917.2021.9447507

A Time Series Data Augmentation Method Based on Dynamic Time Warping

Xinyu Yang, Zhenguo Zhang*, Xu Cui, Rongyi Cui
Department of Computer Science and Technology, Yanbian University
977 Gongyuan Road, Yanji, P.R.China 133002
{2020050049, zgzhang, xcui, cuirongyi}@ybu.edu.cn

***Abstract*—Deep learning has become a hot research topic in the field of time series analysis and data mining. Training models often requires balanced and large data sets, but on the one hand, the number of different types of data in time series data sets is often extremely imbalanced, on the other hand, some time series data are often difficult to collect. Therefore, it is necessary to augment the training data before training the deep learning model. SMOTE is a data augmentation method widely used in preprocessing imbalanced data sets, but the classical SMOTE method does not satisfy the characteristics of time series when processing time series, so it is not effective when applied to time series data sets. To address this point, we propose an oversampling data augmentation method based on dynamic time warping—DTW-SMOTE. For the possible phase shifts of time series with the same characteristics, this method uses dynamic time warping to obtain more reasonable similarity between different time series, which improves the classical SMOTE method and achieves data augmentation for time series data. We conducted comparison experiments using two models with different architectures: ResNet and LSTM. The experimental results show that the performance of the above models is significantly improved after training with the DTW-SMOTE method pre-processed dataset.**

Keywords—data augmentation, time series, dynamic time warping, deep learning

I. INTRODUCTION

Whether it is human activity or nature, time series are generated at any time and any place, such as weather data, financial records, physiological signals and industrial observations [1]. Nowadays, time series analysis is an important and challenging problem in the field of data mining [2]. With the rise of deep learning, scholars have proposed several models for time series analysis. Common deep learning frameworks for time series analysis include LSTM (Long Short-Term Memory) [3], GRU (Gate Recurrent Unit) [4], CNN (Convolutional Neural Networks) [5], ResNet (Deep Residual Networks) [6]. However, due to the large differences in the size of each class in most time series data sets [7], the above model is unlikely to be trained very well.

Data augmentation is an effective way to solve the problem of excessive differences in sample size between classes in time series datasets [8]. SMOTE is a method to solve the imbalance problem of data sets [9], but this method uses Euclidean distance to match the similarity between different data, which is not completely consistent with the characteristics of time series data [10]. Therefore, data augment of time series using classical SMOTE has some disadvantages, the adoption of a new time series similarity measurement method for time series data augmentation is particularly important [11].

To solve the above problems, we propose a Synthetic Minority Oversampling Data Augmentation Method Based on Dynamic Time Warping (DTW-SMOTE) that can effectively augment the time series dataset, and use the data set pre-processed by our proposed method to train the deep learning model can get better performance. We used the DTW-SMOTE method to augment the training data of 17 selected time series data sets, and then trained ResNet and LSTM respectively, and compared the effects of classic SMOTE on the training results. Experimental results show that the model performance is significantly improved after pre-processing the training set with DTW-SMOTE.

II. RELATED TECHNIQUES

A. SMOTE

Synthetic Minority Oversampling Technique (SMOTE) [12] is an effective method to solve the imbalance problem of the data set.

In reality, the amount of data in each class is often extremely imbalanced, and the data set is usually composed of a large number of “normal” samples and a small number of “abnormal” samples. For classification tasks, In the training process of the classifier, the cost obtained by misclassifying a very small number of samples in the training set into the class with a large sample size is often lower than the cost of misclassifying samples in the class with a large sample size, and thus the classifier tends to discard the cost associated with misclassifying a small number of samples, and eventually, it is extremely easy to obtain a classifier that classifies all samples into the class with a large sample size.

SMOTE technique tends to generate new samples from classes with small sample size, so that the sample size of different classes is relatively balanced, the classifier can correctly classify classes with small sample size and improve the classification accuracy of the classifier. As shown in Fig. 1, the method randomly selects some "centers" from the class with a small sample size, uses the KNN [13] algorithm based on Euclidean distance to obtain K neighbors of each “central” sample. Next, the method uses random linear interpolation to mix all the “central” samples with their corresponding K

*corresponding author.

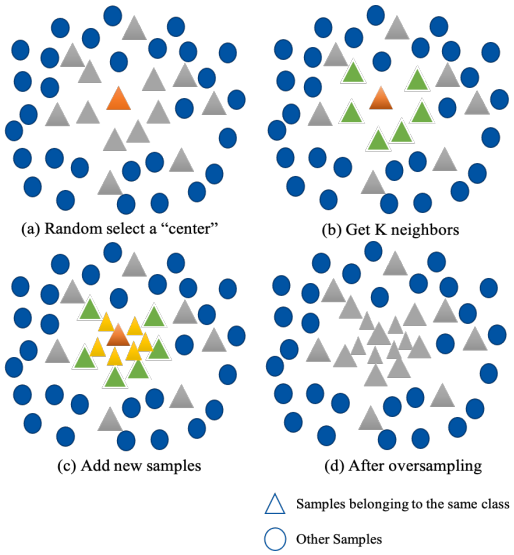

Figure 1. The SMOTE oversampling process.

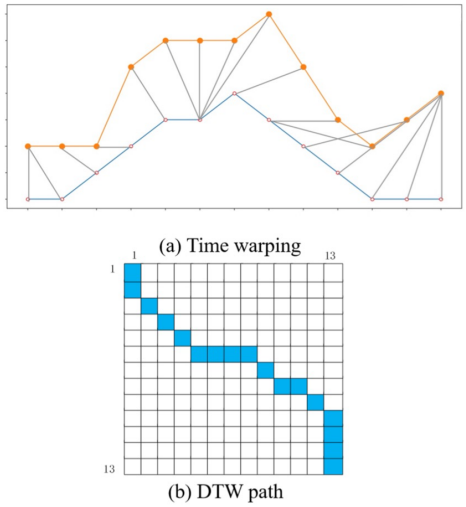

Figure 2. An example of DTW, both time series have dimension 13.

neighbors to obtain some new samples and thus a new training set. Using the new training set augmented by the SMOTE technique to train deep learning model will generally get better results. However, this method cannot meet the special properties of time series data when dealing with time series data, so it cannot achieve data augmentation effectly when pre-processing time series data.

B. DTW

Dynamic Time Warping (DTW) [14] technique makes the comparison of the similarity of time series more reasonable. To accommodate phase shifts and amplitude changes between time series, dynamic time warping uses a dynamic programming algorithm to improve the alignment of time series data points from Euclidean distance "one-to-one" to "one-to-many".

As shown in Fig. 2, the DTW technique constructs a matrix of $n \times m$ when two time series samples $Q = (q_1, q_2, \dots, q_i, \dots, q_n)^T$ and $C = (c_1, c_2, \dots, c_j, \dots, c_m)^T$ are given and lengths are n and m respectively. The matrix element (i, j) is the Euclidean distance $D(q_i, c_j)$ between q_i and c_j. On the premise of satisfying continuity and monotonicity constraints, the path from matrix element $(1,1)$ to matrix element (n, m) is called a *warping path*, the sum of corresponding elements on the path is its cost. The *warping path* with the least cost is the DTW path, and the corresponding cost is the DTW distance between Q and C. Compared with Euclidean distance, this method can more accurately represent the degree of difference between different time series due to the time series being stretched or compressed on the time axis. However, the calculation of DTW is more tedious and time-consuming compared to the calculation of Euclidean distance.

III. OUR APPROACH

The main idea of our proposed DTW-SMOTE method is to use the DTW distance to measure the similarity of different time series, thus improving the classical SMOTE method and improving the performance of deep learning models to analyze time series.

The classical SMOTE method uses the KNN algorithm based on Euclidean distance to obtain the k nearest neighbors of each randomly selected "central" sample, performs a random linear interpolation between the "central" samples and their neighbors to generate new sample points, and adds the new sample points to the training set. After these operations, we get a new training set, which is oversampled by the SMOTE technique, and we can use the new training set to train the deep learning model. However, since the Euclidean distance does not represent the similarity between time series well, we can use DTW for similarity comparison and use the KNN algorithm based on DTW to select the k nearest neighbors of "central" samples.

If the total amount of data in a data set is N and the total amount of classes is Sum, then after we select the overall expansion factor T of the data set, the data amount of the data set after oversampling is NT.

To ensure the balance of the amount of data of different classes, if the amount of data for a class is x, the final amount of data that needs to be expanded for this class is shown below:

$$Num = \left\lfloor \frac{NT}{Sum} \right\rfloor - x \tag{1}$$

Next, we select a hyperparameter k of the KNN algorithm, then the randomly selected "central" sample size in this class is determined as:

$$S = \left\lceil \frac{Num}{k} \right\rceil \tag{2}$$

For each "central" sample Q, we calculate the DTW similarity between Q and the other samples in the same class as Q. If the length of all samples is n, $Q = (q_1, q_2, \dots, q_i, \dots, q_n)^T$. We assume that a sample of the same class as Q is C, $C = (c_1, c_2, \dots, c_j, \dots, c_n)^T$. Then we construct

a $n \times n$ matrix M, the matrix element m_{ij} is the Euclidean distance between q_i and c_j.

$$m_{ij} = \sqrt{(q_i - c_j)^2} \quad (3)$$

We define the *warping path* as follows:

$$W = w_1, w_2, \dots, w_k, \dots, w_K \ (n \le k < 2n-1) \quad (4)$$

Each element of warping path corresponds to a position of matrix M, and w_1 must correspond to m_{11}, and w_K must correspond to m_{nn}. If the warping path element w_k corresponds to the matrix element m_{ij}, w_{k-1} must correspond to one of the matrix elements $m_{i-1,j}$, $m_{i-1,j-1}$, and $m_{i,j-1}$ ($k > 2$ and m_{ij} is not on the boundary of the matrix M).

There may be multiple *warping paths* that meet the above constraints. The path with the smallest sum obtained by accumulating the corresponding matrix elements along the warping path is called DTW path, and the corresponding accumulated value is the DTW distance.

The DTW distance can be easily obtained by dynamic programming approach. We define $dp(i,j)$ as the smallest cumulative sum of all warping paths when moving to the matrix element m_{ij}. We can use the state transition equation to calculate the distance of DTW by dynamic programming algorithm:

$$dp(i,j) = m_{ij} + min\{dp(i-1,j), dp(i-1,j-1), dp(i,j-1)\} \quad (5)$$

The DTW distance between "central" Q and sample C is:

$$DTW(Q,C) = min\left(\sqrt{\sum_{k=1}^{K} w_k}\right) \quad (6)$$

According to the DTW distance, we can get the k nearest neighbors of "center" Q. For each nearest neighbor, we perform a extrapolation with the "center" Q, for a total of k times. To generate more rare and underrepresented samples, we replace the interpolation with extrapolation [15]. We assume that the nearest neighbor of Q is nn, then the new sample New obtained after extrapolation is:

$$New = Q + rand(0,1) * (Q - nn) \quad (7)$$

After extrapolation of all "centers" in each class and their k nearest neighbors, a training data set after data augment with an oversampling multiple of T can be obtained.

It should be noted that when calculating the Euclidean distance of two time series, if the length of the two data is both n, the time complexity is $O(n)$. When using the dynamic programming approach to calculate the DTW distance of two time series, if the length of the two data is also n, the time complexity is $O(n^2)$, Calculating DTW distance is more time-consuming than calculating Euclidean distance. But usually, when we train a deep learning model, every time we update the neural network parameters, we often only need a small batch of data. We can split the training data by class before training the model. At the beginning of training, the model can be trained with training data that has not been augmented. Once the DTW-SMOTE preprocessing of a certain class of data is completed, the previous training data can be replaced by the corresponding part of the new data that has been augmented with DTW to continue training the model, thereby reducing the time spent.

IV. EXPERIMENTS

A. Datasets

We selected 17 time series data sets to verify our proposed data augmentation method. These data sets have been shown to possess the characteristics of time series phase shifts and amplitude changes[1]. All data sets have been divided into training set and testing set by default. Each row of the data set represents a sample, and the first number in each row represents the number of the class to which the sample belongs. The left half of Table I shows the details of these data sets. Column 3 of Table I shows the size of the smallest class in the training set, and column 4 shows the size of the largest class in the training set. As shown in Table I, the training data of most data sets are not balanced, and the sample size of the training sets are small.

B. Experiment Settings

We choose two models with different architectures: ResNet [16][17] and LSTM neural network [18], to perform classification tasks on the above 15 data sets. The structure of the above models is shown in Fig. 3 We use the original training data, the data set that has been oversampled 10 times by SMOTE (based on extrapolation), and the data set that has been oversampled by DTW-SMOTE 10 times to train the model respectively. After training, we use the accuracy of the model on the testing set to evaluate the performance of the model.

To ensure the consistency of the comparison experiment variables, we set all epoch to 1, all learning rate to 0.001, all *batch_size* to 32, all optimizer to *Adam*, and all criterion to *cross_entropy* [19][20][21]. In addition, due to the different number of samples in different classes, for the convenience of calculation, we set the hyperparameter of KNN to 1.

In order to evaluate the performance of different data augmentation methods more objectively and eliminate the influence of chance factors in the training process, we select the average accuracy of the training model on the testing set after removing the maximum and minimum values in 10 experiments as the metric.

C. Results and Analysis

The experimental results of ResNet and LSTM with different data augmentation methods on all datasets are shown in the middle and right half of Table I.

[1]http://timeseriesclassification.com.

TABLE I. DATASETS DETAILS AND AVERAGE ACCURACY (%) OF RESNET AND LSTM

Datasets information				ResNet			LSTM		
Datasets	**# training set**	**Size of Min class**	**Size of Max class**	**No Augmentation**	**SMOTE**	**DTW-SMOTE**	**No Augmentation**	**SMOTE**	**DTW-SMOTE**
50Words	450	1	52	66.8	71.8	**72.5**	70	69.2	**70.4**
ChlorineC.	467	91	262	71.1	82	**84.3**	70.9	84.7	**87.5**
DiatomS.	16	1	6	93.1	93.1	**97.6**	92.8	92.3	**93.8**
DistalP.	276	115	161	72.5	77.2	**80.6**	77.2	76.8	**78.1**
Fish	175	21	28	96.5	97.6	**98**	85.4	85.8	**86.3**
GunPoint	50	24	26	99.2	99.2	**99.6**	90.3	93.1	**93.7**
ItalyP.	67	33	34	95.3	96	**96.1**	96.5	95.8	**96.7**
MiddleP.	291	125	166	75.8	**76.2**	76	71.4	**71.8**	71
OSU Leaf	200	15	53	90	97.6	**97.9**	51.2	54.5	**55.9**
Phoneme	214	1	24	34.6	36.7	**37**	9.8	10.2	**10.3**
ShapeletS.	20	10	10	70.1	89.6	**96**	49.5	49.9	**49.9**
Symbols	25	3	8	94.9	**97.8**	96.9	86.4	86.7	**88.1**
ToeSeg.	40	20	20	85.5	93.5	**96.7**	58.2	58.8	**58.9**
Wine	57	27	30	54.4	60.9	**74.3**	54.9	61.6	**64.4**
WordsS.	267	2	60	57.9	57.7	**59.2**	57.6	59.6	**60.7**
Worms	77	8	33	59.1	61.4	**62.8**	32.9	33.7	**34.9**
WormsT.	77	33	44	70.9	71.3	**72.2**	58.6	56.4	**59.9**

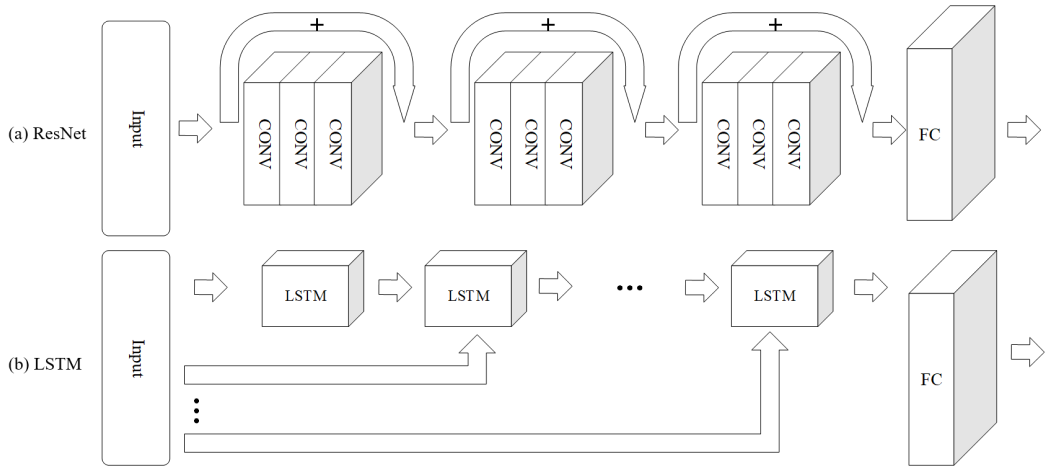

Figure 3. The architecture of models used in our experiments.

The experimental results show that the final performance of the training model on different datasets, whether using the ResNet model or the LSTM model, is improved by training the model with the training data augmented by DTW-SMOTE. In particular, when the sample size of different classes in the training set is imbalanced, the highest average accuracy of the model is improved by 13.43% when using the DTW-SMOTE technique compared to the classical SMOTE.

When we augmented the training data using the classical SMOTE method on the datasets *50words*, *DiatomSizeReduction*, *DistalPhalanxOutlineCorrect* and *ItalyPowerDemand*, the average accuracy of the LSTM was even lower than the case without data augmentation. We obtained the same results for training ResNet after augmenting the training data on the dataset *WordsSynonyms* using the classical SMOTE method. However, after augmenting the above dataset using DTW-SMOTE, the accuracy of the model is still better than the case without using the data augmentation.

Since there may be some datasets that do not have the typical characteristics of time series phase shift, in a few datasets our proposed method is slightly inferior to the classical SMOTE. However, our proposed method still significantly outperforms the classical SMOTE method on most of the data sets.

In addition, we found that using DTW-SMOTE can further improve the convergence speed of the loss function after the training starts. To verify this inference, we recorded the change curves of the Loss function when the two models were trained with different methods on the *DiatomSizeReduction* dataset. As shown in Fig. 4, both SMOTE and DTW-SMOTE can improve the convergence speed of the Loss function after the start of training, and the convergence speed using the DTW-SMOTE method is further improved than the SMOTE method. The loss values using the DTW-SMOTE method are lower than the other two cases in most of the training processes. When we use the augmented training set, there may be some errors that lead

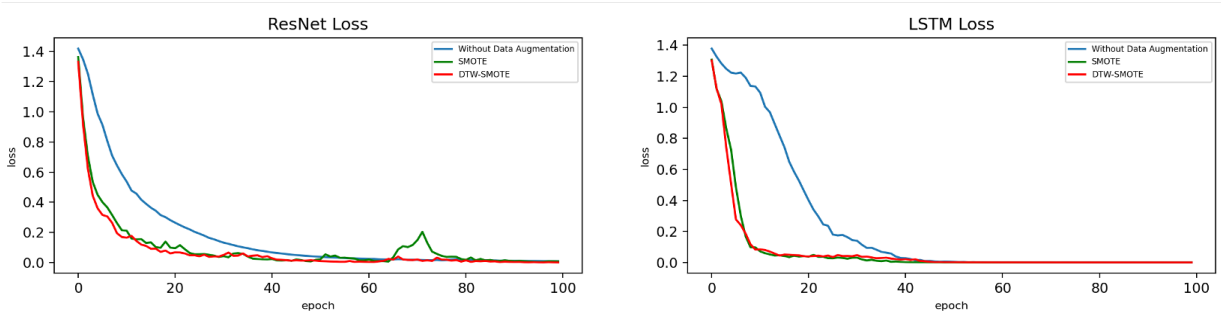

Figure 4. Variation of Loss function when training models.

to sudden fluctuations in the loss during training. Therefore, it is important to use cross-validation or other effective methods when training models with the training data augmented by DTW-SMOTE.

V. Conclusion

In this paper, we propose a new time series data augmentation method DTW-SMOTE, which combines DTW and SMOTE to make the oversampling of time series data more reasonable. The method randomly selects some sample points from each class, uses KNN based on DTW to find neighbors, and uses randomly selected sample points to perform extrapolation with their neighbors, to achieve the purpose of data augmentation for time series training data. The experimental results show that compared with the classic SMOTE, the accuracy of the model and the convergence speed of the loss function is improved after using the method we proposed. In future work, we will explore to overcome the errors caused by randomly selected sample points and extrapolation.

Acknowledgment

This work is supported by the school-enterprise cooperation project of Yanbian University [2020-16], Doctor Starting Grants of Yanbian University [2020-16] and the project of school-enterprise cooperation "The cultivation of innovative talents for computer discipline under the background of emerging engineering education".

References

[1] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, "Deep learning for time series classification: a review," Data Mining and Knowledge Discovery, vol. 33, no. 4, pp. 917-963, 2019.

[2] J. C. B. Gamboa, "Deep learning for time-series analysis," arXiv preprint arXiv:1701.01887, 2017.

[3] F. A. Gers, D. Eck, and J. Schmidhuber, "Applying LSTM to time series predictable through time-window approaches," in Neural Nets WIRN Vietri-01, Springer, London, 2002, pp. 193-200.

[4] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," Scientific reports, vol. 8, no. 1, pp. 1-12, 2018.

[5] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, "Convolutional neural networks for time series classification," Journal of Systems Engineering and Electronics, vol. 28, no. 1, pp. 162-169, 2017.

[6] H. Choi, S. Ryu, and H. Kim, "Short-term load forecasting based on ResNet and LSTM," in 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), 2018, pp. 1-6.

[7] H. Cao, X. L. Li, D. Y. K. Woon, and S. K. Ng, "Integrated oversampling for imbalanced time series classification," IEEE Transactions on Knowledge and Data Engineering, vol. 25, no.12, pp. 2809~2822, 2013.

[8] C. Shorten, and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," Journal of Big Data, vol. 6, no. 1, pp. 60, 2019.

[9] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, "SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary," Journal of artificial intelligence research, vol. 61, pp. 863-905, 2018.

[10] E. Keogh, and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," Knowledge and information systems, vol. 7, no. 3, pp. 358-386, 2005.

[11] M. Cuturi, and M. Blondel, "Soft-DTW: a differentiable loss function for time-series," arXiv preprint arXiv:1703.01541, 2017.

[12] N. V. Chawla, K.W. Bowyer, L. O. Hall, and W.P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," Journal of artificial intelligence research, vol. 16, pp. 321-357, 2002.

[13] T. Cover, and P. Hart, "Nearest neighbor pattern classification," IEEE transactions on information theory, vol. 13, no. 1, pp. 21-27, 1967.

[14] D. J. Berndt, and J. Clifford, "Using dynamic time warping to find patterns in time series," inn KDD workshop, 1994, Vol. 10, No. 16, pp. 359-370.

[15] T. DeVries, and G. W. Taylor, "Dataset augmentation in feature space," arXiv preprint arXiv: 1702.05538, 2017.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Communications of the ACM, vol. 60, no. 6, pp. 84-90, 2017.

[17] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in 2017 International joint conference on neural networks (IJCNN) , 2017, pp. 1578-1585.

[18] S. Hochreiter, and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[19] M. Li, T. Zhang, Y. Chen, and A. J. Smola, "Efficient mini-batch training for stochastic optimization," in Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 2014, pp. 661-670.

[20] D. P. Kingma, and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[21] P. T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," Annals of operations research, vol.134, no. 1, pp. 19-67, 2005.