Deep Neural Network Ensembles for Time Series Classification

Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar and Pierre-Alain Muller
IRIMAS, Université Haute-Alsace, Mulhouse, France
Email: {first-name.last-name@uha.fr}

***Abstract*—Deep neural networks have revolutionized many fields such as computer vision and natural language processing. Inspired by this recent success, deep learning started to show promising results for Time Series Classification (TSC). However, neural networks are still behind the state-of-the-art TSC algorithms, that are currently composed of ensembles of 37 non deep learning based classifiers. We attribute this gap in performance due to the lack of neural network ensembles for TSC. Therefore in this paper, we show how an ensemble of 60 deep learning models can significantly improve upon the current state-of-the-art performance of neural networks for TSC, when evaluated over the UCR/UEA archive: the largest publicly available benchmark for time series analysis. Finally, we show how our proposed Neural Network Ensemble (NNE) is the first time series classifier to outperform COTE while reaching similar performance to the current state-of-the-art ensemble HIVE-COTE.**

I. Introduction

Time series data are omnipresent in many practical data science applications ranging from health care [1] and stock market predictions [2] to social media analysis [3] and human activity recognition [4]. Since 2006, time series analysis has been considered one of the most challenging problems in data mining [5], and in a more recent poll it has been shown that 48% of data expert had analyzed time series data during their career, ahead of text and images [6].

Time Series Classification (TSC) tasks differ from traditional classification tasks by the natural temporal ordering of their attributes [7]. To tackle this problem, a huge amount of research was dedicated into coupling and enhancing time series similarity measures with a Nearest Neighbor (NN) classifier [8], [9]. In [10], ten elastic distances were compared to the traditional Dynamic Time Warping (DTW) algorithm to find out that no single measure could outperform the classic NN coupled with DTW (NN-DTW) for TSC. These findings motivated the authors to construct a single Elastic Ensemble (EE) classifier that includes all eleven different similarity measures, and achieve a significant improvement compared to the individual classifiers [10]. Hence, recent contributions were focused on ensembling different discriminant classifiers such as decision trees (random forest) [11] and Support Vector Machines (SVMs) [12] on different data representation techniques such shapelet transform [12] or DTW features [13]. These ideas gave rise to the Collective Of Transformation-based Ensembles (COTE) [14] and its extended version HIVE-COTE [15] where 37 different classifiers were ensembled over multiple time series data transformation techniques in order to reach current state-of-the-art performance for TSC [7].

With the advent of deep neural networks into industrial and commercial applications such as self-driving cars [16] and speech recognition systems [17], time series data mining practitioners started investigating the application of deep learning to TSC problems [18]. In our recent empirical study [19], we showed how deep Convolutional Neural Networks (CNNs) are able to achieve results that are not significantly different than current state-of-the-art algorithms for TSC problems when evaluatedover the 85 time series datasets from the UCR/UEA archive [7], [20]. Outside the UCR/UEA benchmark, deep neural networks have seen some very successful applications such as evaluating surgical skills from multivariate time series [1] and recognizing human activities from wearable sensors data [4]. These results suggest that building upon deep learning based solutions for TSC could further improve the current state-of-the-art performance of deep neural networks.

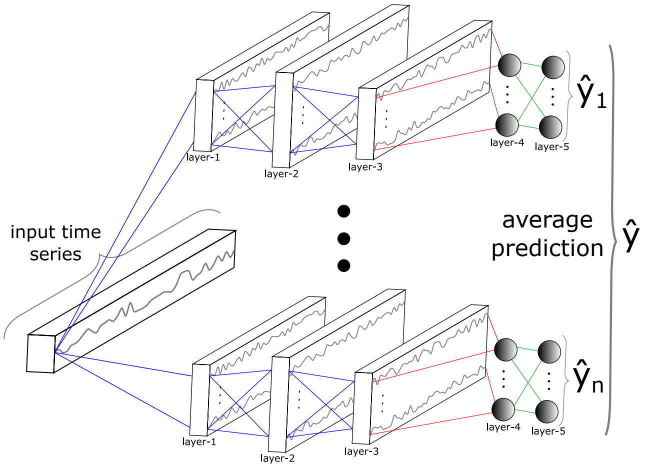

Fig. 1: Ensemble of deep convolutional neural networks for time series classification.

One way of improving neural network based classifiers is to build an ensemble of deep learning models. This idea seems very interesting for TSC tasks since the state-of-the-art is moving towards ensembled solutions [7], [10], [11], [15]. In addition, deep neural network ensembles seem to achieve very promising results in many supervised machine learning domains such as skin lesions detection [21], facial expression recognition [22] and automatic bucket filling [23].

Therefore, we propose to ensemble the current state-of-the-art deep learning models for TSC developed in [19], by constructing one model composed of 60 different deep neural networks: 6 different architectures [18], [24]–[26] each one with 10 different initial weight values. By evaluating on the 85 datasets from the UCR/UEA archive, we demonstrate a significant improvement over the individual classifiers while also reaching very similar performance to HIVE-COTE: the current state-of-the-art ensemble of 37 non deep learning based time series classifiers. Finally, inspired by the recent success of transfer learning for TSC [27], we replace ensembling randomly initialized networks with an ensemble constructed out of fine-tuned models from 84 different source datasets, which showed a significant improvement for TSC problems.

The paper is divided as follows, we first start by explaining the background material, before presenting our different techniques of ensembling deep neural networks. We then describe our results and discussions before drafting a final conclusion with our future directions.

II. Background

In this section we describe the current state of research in neural networks for TSC and then present some work related to ensembling neural network classifiers.

A. *Neural networks for time series classification*

Since AlexNet [28] won the ImageNet [29] competition in 2012 with a significant improvement in accuracy compared to previous state-of-the-art approaches, the computer vision field was revolutionized with many deep neural networks papers being published every year to solve image recognition and object localization problems [30]. In addition, sequential data mining tasks such as natural language processing and speech recognition are being tackled with deep convolutional, recurrent and generative adversarial neural networks [31], [32].

Inspired by this recent success of deep learning models, researchers started adopting these complex machine learning techniques to solve the underlying task of Time Series Classification [19], [33]. Specifically Wang *et al.* [18] showed very promising results, where a Fully Convolutional Neural network (FCN) and a Residual Network (ResNet) were designed to reach COTE's performance when evaluated on 44 datasets from the UCR/UEA archive [7], [20]. Moreover, in our recent empirical evaluation of deep learning models for TSC [19], we managed to reinforce these findings by testing FCN and ResNet on 85 datasets from the UCR/UEA archive. In fact, similar to two dimensional data (images), one dimensional convolutions when slid over an input time series, enable a non-linear transformation of the data. By applying backpropagation over a cascade of several convolutional layers with many filters, the network is able to learn this time invariant hierarchical representation of the input time series which is potentially useful for classification. For more detail about how these convolutions are being applied to one dimensional time series data, we refer the interested reader to our recent survey of deep learning for time series classification [19].

Different variants of CNNs were proposed for TSC and validated on the UCR/UEA archive. Multi-scale CNN (MCNN) [34] was among the first deep learning architectures to be evaluated for domain agnostic TSC. In [35] Time LeNet (t-LeNet) was proposed as an adaptation of the famous LeNet architecture which was originally proposed for document recognition [36]. Multi-Channels Deep Convolutional Neural Networks (MCDCNN) [24] and Time-CNN [25] were originally proposed for multivariate TSC, however in [19] we have shown how they can be easily extended for univariate TSC. One last CNN model called Encoder was proposed in [26] where FCN was extended to include the attention mechanism. Adding to the aforementioned neural network architectures, the classical Multi-Layer Perceptron (MLP) was considered as a baseline architecture in [18]. Finally, we should mention in addition to this pool of deep CNNs for TSC, a non-convolutional recurrent model called Time Warping Invariant Echo State Networks (TWIESN) [37], which showed promising results on different datasets in the archive [19].

In [19], we showed how ResNet, FCN and Encoder won on 43, 18 and 10 datasets respectively suggesting that indeed no single network would outperform all the others on the whole benchmark. This would motivate researchers to ensemble the decision of these deep learning classifiers, which is the main contribution of this paper: showing how an ensemble of different deep neural networks can outperform all single individual classifiers and reach new state-of-the-art performance for TSC.

B. *Neural networks ensemble*

Constructing an ensemble of many deep learning classifiers has been shown to achieve high performance in many different fields. In [21], an ensemble of two neural networks was adopted: (1) Inception-v4 and (2) Inception-ResNet-v2. Both of these classifiers are learned with a joint meta-learning approach in an end-to-end manner. A forest CNN was proposed in [38] for image classification, where similarly to random forest, the ensemble is constructed by replacing the individual nodes with a CNN and finally the classifier's decision is taken by performing a majority voting scheme over the different decisions of the individual trees in the forest. Another ensemble of CNNs for facial expression recognition was proposed in [22] where each individual classifier was trained independently to output a probability for each class and then the network's final decision was taken using a probability-based fusion method. In [23], an ensemble of neural networks was found to outperform other hybrid machine learning ensembles when solving an automatic bucket filling problem. Finally in [39], deep auto-encoders were ensembled in order to learn an unsupervised latent representation of the input data over multiple resolutions, thus improving the quality of the produced clusters.

Although in almost all use cases ensembling deep neural networks almost always yields to better decisions, we did not find any approach using a neural network ensemble for domain agnostic TSC. Perhaps the work in [40] is the closest to ours where a neural network based ensemble was used to perform biomedical TSC, where individual architectures were constructed with some domain knowledge specific to the classification problem at hand such as choosing the filter length with local and distorted views. In addition, our recent work on ensembling two deep learning models (with or without

Approach	Rank	Wins
ResNet [18]	1.88	41
FCN [18]	2.49	18
Encoder [26]	3.34	10
MLP [18]	4.08	4
Time-CNN [25]	4.38	4
MCDCNN [24]	4.83	3

TABLE I: Average rank of the six classifiers constituting the Neural Network Ensemble for time series classification over the 85 datasets from the UCR/UEA archive.

data augmentation) showed how the ensemble classifier was able to outperform significantly the individual model [41]. Therefore, we decided to further explore ensembling deep neural networks for TSC, by combining multiple deep learning models in different settings.

III. Methods

In this section, we start by presenting the six different architectures composing our ensembles of neural networks. For completeness, we describe the random initialization technique adopted for all models. Finally, we present a transfer learning based alternative to randomly initializing the weights of the networks.

A. Architectures

The average rank of the six chosen deep learning classifiers, over the 85 datasets from the UCR/UEA archive [7], [20] is listed in Table I. All of these architectures were implemented in a common framework during our empirical study [19], containing originally 9 different deep learning approaches for TSC. However only 6 out of these 9 approaches were probabilistic classifiers whereas the three other classifiers performed a hard prediction: meaning an input time series is assigned a specific class rather than a probability distribution over all the classes in a dataset. Therefore, we chose to only ensemble the 6 probabilistic models, thus allowing us to combine the networks by averaging the a posteriori probability for each class over the individual classifiers' output. Finally, we present a brief description of these 6 different architectures and refer the interested reader to a more thorough explanation in the corresponding papers. All hyperparameters can be found in [19].

1) Multi-Layer Perceptron: (MLP) is the simplest form of deep neural networks and was proposed in [18] as a baseline architecture for TSC. The architecture contains three hidden layers, with each one fully connected to the output of its previous layer. The main characteristic of this architecture is the use of a Dropout layer [42] to reduce overfitting. One disadvantage is that since the input time series is fully connected to the first hidden layer, the temporal information in a time series is lost [19].

2) Fully Convolutional Neural Network: (FCN), originally proposed in [18], is considered a competitive architecture yielding the second best results when evaluated on the UCR/UEA archive (see Table I). This network is comprised of three convolutional layers, each one performing a non-linear transformation of the input time series. A global average pooling operation is used before the final softmax classifier, thus reducing drastically the number of parameters in a network and allowing an architecture that is invariant to the length of the input time series. The latter characteristic motivated us to perform a transfer learning technique in [27], and ensembling the resulting neural networks which is later discussed in Section III-C.

3) Residual Network: (ResNet) was originally proposed in [18] and showed similar performance to FCN when evaluated on 44 datasets from the archive. However, when evaluated over the 85 datasets, ResNet significantly outperformed FCN (see Table I). The main characteristic of ResNet is the addition of residual connections which enables a direct flow of the gradient [18].

4) Encoder: (Encoder) was originally proposed in [26] as a hybrid CNN that modifies the FCN architecture [18] by mainly adding a Dropout layer [42] and an attention mechanism. The latter operation enables Encoder to learn to localize which regions of the input time series are useful for a certain class identification.

5) Multi-Channels Deep Convolutional Neural Networks: (MCDCNN) was originally proposed in [24] for multivariate TSC and adapted to univariate data in [19]. It consists of a traditional CNN, where each convolutional layer is followed by a max pooling operation, then a traditional fully connected layer is used before the final softmax classifier.

6) Time Convolutional Neural Network: (Time-CNN) was originally proposed for univariate as well as multivariate TSC [25]. Similarly to MCDCNN, this network is a traditional CNN with one major exception: the use of the mean squared error instead of the traditional categorical cross-entropy loss function, which has been used by all the deep learning approaches we have mentioned so far. Therefore for Time-CNN, the sum of the output class probabilities is not guaranteed to be equal to one.

B. Ensembling models with random initial weights

We have described in the previous subsection, the architecture of six different classifiers. The weights for each network are initialized randomly using Glorot's uniform initialization method [43]. This technique ensures a uniform distribution of the initial weight values. However due to non-convexity, networks with the same architecture but different initial weights could yield different validation accuracy. In [44], the authors showed that deeper networks are much more stable with respect to the randomness. This would suggest that ensembling relatively non deep architectures would yield to a much better improvement in accuracy than ensembling deeper architectures. Fortunately, for low dimensional time series data, current state-of-the-art architectures are much less deeper than their counterpart networks for high dimensional images. Therefore, we believe that we can leverage this instability of neural networks for time series data by ensembling the decision taken by the same network but with different random initializations, using the following equation:

$$\hat{y}_{i,c} = \frac{1}{n}\sum_{j=1}^{n}\sigma_c(x_i,\theta_j) \ \mid \ \forall c \in [1, C] \tag{1}$$

with $\hat{y}_{i,c}$ denoting the ensemble's output probability of having the input time series x_i belonging to class c, which is equal to the logistic output σ_c averaged over the n randomly initialized models. We should note that training an ensemble of the same architecture with different initial weight values has been shown to improve neural network's performance on many computer vision problems [22], however, we did not encounter any previous work that combines such classifiers for TSC.

C. Transfer learning

An alternative to training a deep classifier from scratch is to fine-tune a model that has been already pre-trained on a un/related task [27]. This process is called transfer learning, where the network is first trained on a source dataset, then the final layer is removed and replaced with a new randomly initialized softmax layer whose number of neurons is equal to the number of classes in the target dataset. The pre-trained model is then fine-tuned or re-trained on the target dataset's training set. With 85 datasets in the archive, each target dataset will have 84 potential source datasets, which motivated us to ensemble the decision of these 84 FCN models.

IV. Results

In this section we present the results of different ensembling schemes when evaluated on the 85 datasets from the UCR/UEA archive [7], [20], which is currently the largest publicly available benchmark for time series analysis. In order to compare multiple classifiers over several datasets, following the recommendations in [45], we perform the Friedman test to first reject the null hypothesis. For the post-hoc analysis, following the recent recommendations in [46], we abandoned the average rank comparison in favor of a pairwise statistical comparison: the Wilcoxon signed-rank test with Holm's alpha correction ($\alpha = 5\%$). Finally, we used a critical difference diagram [45] to visualize the results of these statistical tests projected onto the average rank axis, with a thick horizontal line showing a clique of classifiers that are not significantly different (see Figure 2 for an example of such diagram). All experiments were conducted on a hybrid cluster of more than 60 NVIDIA GPUs comprised of GTX 1080 Ti, Tesla K20, K40 and K80. Note that the code, the raw results and all the pre-trained models are publicly available on the paper's companion repository[1].

A. Ensembling randomly initialized models

By ensembling randomly initialized networks, we are able to achieve a significant improvement in accuracy. Figure 2 shows a critical difference diagram where ten different random initializations of ResNet did not yield to significantly different results. However, by ensembling these different networks, we were able to demonstrate a significant improvement in the average rank over the 85 datasets. We should note that the latter phenomenon was also observed for the five other neural networks described in Section III. Finally, we should emphasize that an ensembling technique will improve the stability of ResNet in terms of accuracy, in other words reducing the bias due to the initial weight values as well as the randomness induced by gradient descent based optimization.

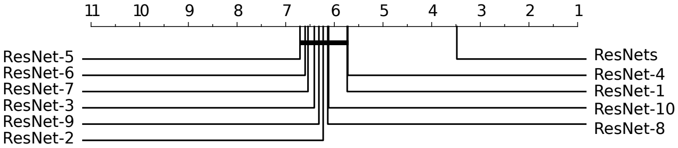

Fig. 2: Critical difference diagram showing the pairwise statistical comparison of ten ResNets with random initializations as well as one ResNet ensemble composed of these ten individual neural networks.

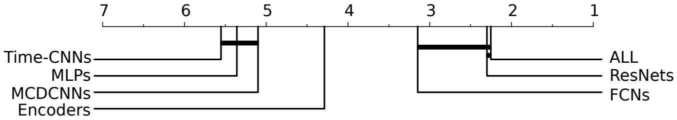

Fig. 3: Critical difference diagram showing the pairwise statistical comparison of six architectures ensembled with ten different random initializations each, as well as one ensemble containing the six models.

B. Ensembling all neural networks

After demonstrating that using an ensemble of neural networks is always better than a single classifier, we sought to answer the following question: *Could an ensemble of hybrid randomly initialized networks achieve even better performance?* Figure 3 shows a critical difference diagram containing six ensembles of homogenized networks as well as the hybrid ensemble of *all* available networks. The latter classifier contains sixty different networks: each architecture (six in total) is initialized with ten different random weight values. The results show that ensembling all networks was able to outperform all classifiers. However the statistical test failed to find any significant difference between the full ensemble and individual ResNet/FCN ensembles. This would suggest that the ensemble is highly affected by the poor performance of Time-CNN, MLP and MCDCNN. The latter classifiers showed the worst average rank without any significant difference, thus suggesting that removing them would yield even better performance.

C. Neural Network Ensemble

The results in the previous section, suggest that choosing carefully the classifiers in the pool would yield to a better ensemble. Therefore, we construct a Neural Network Ensemble (NNE) comprised solely of ResNet, FCN and Encoder. These three architectures were the only ones to yield significantly

[1] https://github.com/hfawaz/ijcnn19ensemble

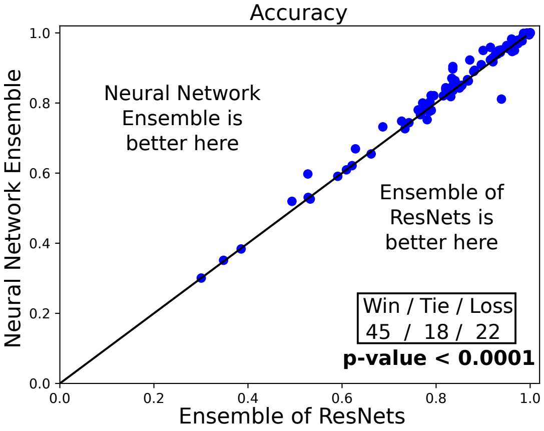

Fig. 4: The Neural Network Ensemble (NNE) composed of ResNet, FCN and Encoder is significantly better than an ensemble of pure ResNets.

different results when a homogenized ensemble was adopted (Figure 3). Further investigations suggested that FCN performs better than ResNet on electrocardiography datasets [19], which would motivate researchers to combine these two classifiers in order to have a robust algorithm that improves the accuracy over the whole datasets. However, for small datasets such as DiatomSizeReduction, both FCN and ResNet overfitted the dataset very easily with 30% test accuracy [41], whereas Encoder managed to achieve very good performance with a 92% accuracy, therefore implying a combination of ResNet, FCN and Encoder would yield to better accuracy on a various range of TSC datasets. Figure 4 shows how NNE is able to outperform an ensemble of pure ResNets with 45 wins and 18 ties on 85 datasets from the archive. We believe that the combination of an FCN with ResNet and Encoder, enables the classifier to benefit respectively from the residual linear connections and the attention mechanism.

To further understand how NNE is performing with respect to current state-of-the-art TSC algorithms, we illustrate in Figure 5 a critical difference diagram containing NNE and seven other non deep learning based classiifers: (1) NN-DTW corresponds to the nearest neighbor coupled with the Dynamic Time Warping distance; (2) EE is an ensemble of nearest neighbor classifiers with eleven elastic distances; (3) BOSS corresponds to the ensemble Bag-of-SFA-Symbols; (4) ST is another ensemble of off-the-shelf classifiers computed over the Shapelet Transform data domain; (5) PF or Proximity Forest is an ensemble of decision trees coupled with eleven elastic distances; finally (6) COTE and (7) HIVE-COTE are two ensembles of respectively 35 and 37 classifiers using multiple data transformation techniques. The results for these classifiers were taken from [7] except for PF whose results were taken from the original paper [47]. Figure 5 clearly shows how our NNE is able to reach state-of-the-art performance for TSC, suggesting that CNNs are able to extract one dimensional discriminant features useful for classification in an end-to-end manner, as opposed to other hand-engineered features used by HIVE-COTE such as the Discrete Fourier Transform, DTW features and the Shapelet Transform.

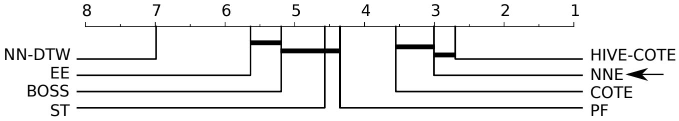

Fig. 5: Critical difference diagram showing the pairwise statistical comparison of current state-of-the-art algorithms with the Neural Network Ensemble (NNE) added to the pool.

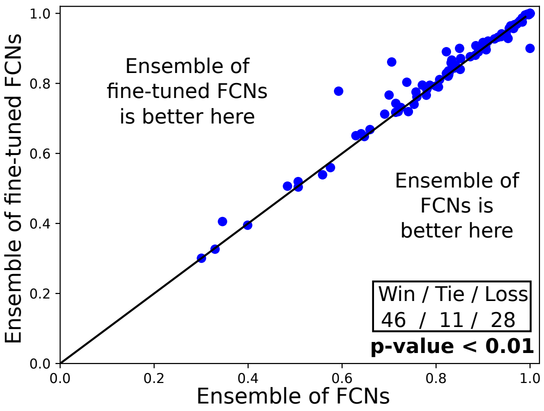

Fig. 6: Ensembling fine-tuned models is significantly better than ensembling randomly initialized FCN models that are trained from scratch.

D. *Ensembling fine-tuned models*

Figure 6 shows that ensembling fine-tuned FCNs is significantly better than ensembling randomly initialized FCN models that are trained from scratch. However, this transfer learning based ensemble did not manage to outperform ResNets' ensemble nor NNE. These results show that the choice of architecture is very crucial and suggest that an ensemble of transferred ResNets would demonstrate even better performance than an ensemble of pure ResNets or NNE.

V. Conclusion

In this paper, we showed how ensembling deep neural networks can achieve state-of-the-art performance for time series classification. We showed that it would be almost always beneficial to ensemble randomly initialized models rather than choosing one trained neural network out of the ensemble. Finally, we investigated an ensemble of transferred deep CNNs to demonstrate even better performance than ensembling randomly initialized networks. In the future, we would like to consider a meta-learning approach where the output logistics of individual deep learning models are fed to a meta-network that learns to map these inputs to the correct prediction.

ACKNOWLEDGMENT

The authors would like to thank the providers of the UCR/UEA benchmark datasets, as well as NVIDIA Corporation for the GPU Grant and the Mésocentre of Strasbourg for providing access to the GPU cluster.

References

[1] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Evaluating surgical skills from kinematic data using convolutional neural networks," in *International Conference On Medical Image Computing and Computer Assisted Intervention*, 2018, pp. 214–221.

[2] L. Anghinoni, L. Zhao, Q. Zheng, and J. Zhang, "Time series trend detection and forecasting using complex network topology analysis," in *International Joint Conference on Neural Networks*, 2018, pp. 1–7.

[3] N. Xu, G. Chen, and W. Mao, "MNRD: A merged neural model for rumor detection in social media," in *International Joint Conference on Neural Networks*, 2018, pp. 1–7.

[4] R. Xi, M. Hou, M. Fu, H. Qu, and D. Liu, "Deep dilated convolution on multimodality time series for human activity recognition," in *International Joint Conference on Neural Networks*, 2018, pp. 1–8.

[5] Q. Yang and X. Wu, "10 challenging problems in data mining research," *Information Technology & Decision Making*, vol. 05, no. 04, pp. 597–604, 2006.

[6] G. Piatetsky, "Data types/sources analyzed," https://www.kdnuggets.com/2014/05/poll-results-data-types-sources-analyzed.html.

[7] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Mining and Knowledge Discovery*, vol. 31, no. 3, pp. 606–660, 2017.

[8] H. A. Dau, D. F. Silva, F. Petitjean, G. Forestier, A. Bagnall, and E. Keogh, "Judicious setting of dynamic time warping's window width allows more accurate classification of time series," in *IEEE International Conference on Big Data*, 2017, pp. 917–922.

[9] S. Gharghabi, S. Imani, A. Bagnall, A. Darvishzadeh, and E. Keogh, "An ultra-fast time series distance measure to allow data mining in more complex real-world deployments," in *IEEE International Conference on Data Mining*, 2018, pp. 17–20.

[10] J. Lines and A. Bagnall, "Time series classification with ensembles of elastic distance measures," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 565–592, 2015.

[11] M. G. Baydogan, G. Runger, and E. Tuv, "A bag-of-features framework to classify time series," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2796–2802, 2013.

[12] A. Bostrom and A. Bagnall, "Binary shapelet transform for multiclass time series classification," in *Big Data Analytics and Knowledge Discovery*, 2015, pp. 257–269.

[13] R. J. Kate, "Using dynamic time warping distances as features for improved time series classification," *Data Mining and Knowledge Discovery*, vol. 30, no. 2, pp. 283–312, 2016.

[14] A. Bagnall, J. Lines, J. Hills, and A. Bostrom, "Time-series classification with COTE: The collective of transformation-based ensembles," in *International Conference on Data Engineering*, 2016, pp. 1548–1549.

[15] J. Lines, S. Taylor, and A. Bagnall, "Time series classification with HIVE-COTE: The hierarchical vote collective of transformation-based ensembles," *ACM Transactions on Knowledge Discovery from Data*, vol. 12, no. 5, pp. 52:1–52:35, 2018.

[16] Z. Qiu and X. Gu, "Multi ROI and multi map networks for accurate and efficient pedestrian detection," in *International Joint Conference on Neural Networks*, 2018, pp. 1–7.

[17] B. Liul, S. Nie, S. Liang, Z. Yang, and W. Liu, "Stochastic multiple choice learning for acoustic modeling," in *International Joint Conference on Neural Networks*, 2018, pp. 1–6.

[18] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *International Joint Conference on Neural Networks*, 2017, pp. 1578–1585.

[19] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery*, 2019.

[20] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, "The UCR Time Series Classification Archive," July 2015, www.cs.ucr.edu/~eamonn/time_series_data/.

[21] M. Goyal and J. C. Rajapakse, "Deep neural network ensemble by data augmentation and bagging for skin lesion classification," *ArXiv*, 2018.

[22] G. Wen, Z. Hou, H. Li, D. Li, L. Jiang, and E. Xun, "Ensemble of deep neural networks with probability-based fusion for facial expression recognition," *Cognitive Computation*, vol. 9, no. 5, pp. 597–610, 2017.

[23] S. Dadhich, F. Sandin, and U. Bodin, "Predicting bucket-filling control actions of a wheel-loader operator using a neural network ensemble," in *International Joint Conference on Neural Networks*, 2018, pp. 1–6.

[24] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Time series classification using multi-channels deep convolutional neural networks," in *Web-Age Information Management*, 2014, pp. 298–310.

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

[26] J. Serrà, S. Pascual, and A. Karatzoglou, "Towards a universal neural network encoder for time series," *ArXiv*, 2018.

[27] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Transfer learning for time series classification," in *IEEE International Conference on Big Data*, 2018, pp. 1367–1376.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1097–1105.

[29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[30] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[31] C. Li, Y. Su, and W. Liu, "Text-to-text generative adversarial networks," in *International Joint Conference on Neural Networks*, 2018, pp. 1–7.

[32] X. Wang, C. Li, and B. Xu, "Hierarchical tree long short-term memory for sentence representations," in *International Joint Conference on Neural Networks*, 2018, pp. 1–6.

[33] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Adversarial attacks on deep neural networks for time series classification," in *IEEE International Joint Conference on Neural Networks*, 2019.

[34] Z. Cui, W. Chen, and Y. Chen, "Multi-Scale Convolutional Neural Networks for Time Series Classification," *ArXiv*, 2016.

[35] A. Le Guennec, S. Malinowski, and R. Tavenard, "Data Augmentation for Time Series Classification using Convolutional Neural Networks," in *International Workshop on Advanced Analytics and Learning on Temporal Data, ECML PKDD*, 2016.

[36] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[37] P. Tanisaro and G. Heidemann, "Time series classification using time warping invariant echo state networks," in *IEEE International Conference on Machine Learning and Applications*, 2016, pp. 831–836.

[38] J. Lee, M. Kang, and J. Kang, "Ensemble of binary tree structured deep convolutional network for image classification," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2017, pp. 1448–1451.

[39] D. Ienco and R. G. Pensa, "Semi-supervised clustering with multiresolution autoencoders," in *International Joint Conference on Neural Networks*, 2018, pp. 1–8.

[40] L.-p. Jin and J. Dong, "Ensemble deep learning for biomedical time series classification," *Computational Intelligence and Neuroscience*, vol. 2016, pp. 13–, 2016.

[41] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Data augmentation using synthetic data for time series classification with deep residual networks," in *International Workshop on Advanced Analytics and Learning on Temporal Data, ECML PKDD*, 2018.

[42] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[43] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International Conference on Artificial Intelligence and Statistics*, vol. 9, 2010, pp. 249–256.

[44] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun, "The loss surfaces of multilayer networks," in *International Conference on Artificial Intelligence and Statistics*, 2015, pp. 192–204.

[45] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[46] A. Benavoli, G. Corani, and F. Mangili, "Should we really use post-hoc tests based on mean-ranks?" *Machine Learning Research*, vol. 17, no. 1, pp. 152–161, 2016.

[47] B. Lucas, A. Shifaz, C. Pelletier, L. O'Neill, N. Zaidi, B. Goethals, F. Petitjean, and G. I. Webb, "Proximity forest: an effective and scalable distance-based classifier for time series," *Data Mining and Knowledge Discovery*, 2019.