Journal of Neonatal Surgery
ISSN(Online): 2226-0439
Vol. 14, Issue 14s (2025)
https://www.jneonatalsurg.com

OPEN ACCESS

Enhanced Temporal Convolutional Networks for Robust, Efficient Time Series Classification

Dasari Alekhya[1], Donavalli Haritha[2]

[1,2] Department of Computer Science and Engineering Koneru Lakshmaiah Education Foundation, Guntur, India
[1]alekhyadasari001@gmail.com, [2]haritha_donavalli@kluniversity.in

Cite this paper as: Dasari Alekhya, Donavalli Haritha, (2025) Enhanced Temporal Convolutional Networks for Robust, Efficient Time Series Classification. *Journal of Neonatal Surgery*, 14 (14s), 673-681.

ABSTRACT

Time interdependence and different sequence lengths make time series data categorization still a difficult problem in machine learning. This article offers a better Temporal Convolutional Network (TCN) design that uses dilated causal convolutions and residual connections to solve these issues. Across many benchmark datasets, we methodically contrast our approach with conventional techniques like Long Short-Term Memory (LSTM) networks, Gated Recurrent Units (GRUs), and 1D Convolutional Neural Networks (CNNs). Experimental findings show that our TCN implementation outperforms traditional methods in terms of classification accuracy, computational efficiency, and resilience to varied sequence lengths. Along with ablation experiments confirming our design choices, we offer thorough examination of architectural options like kernel size, dilation factors, and residual block design. Aiming for uses from healthcare monitoring to industrial sensor data analysis, the suggested architecture has specific strength in encapsulating multivariate time series data's short- and long-term dependency.

Keywords: *Deep learning, temporal convolutional networks, time series classification, dilated convolutions, residual connections*

1. INTRODUCTION

Classifying time series data is a core topic in machine learning, with applications spanning numerous fields. The monitoring of healthcare, the forecasting of financial markets, the analysis of industrial sensors, and the detection of human behaviors are all examples of industries that fall under this category [1]. Data that is time series has temporal relationships that need to be correctly described in order to have excellent classification performance capabilities. This is in contrast to data that is static. It is important to note that static data does not possess these linkages, in contrast to dynamic data.Traditional time series classification methods often rely on distance-based techniques. Some examples of these techniques are Dynamic Time Warping (DTW) [2], feature engineering strategies [3], and more recently, deep learning architectures [4]. These many methodological techniques have all been integrated into the categorizing process that has been carried out. The capacity of Recurrent Neural Networks (RNNs) to recreate temporal connections has led to the widespread application of these networks for sequential data. This has increased the likelihood that RNNs will be used. Long Short-Term Memory (LSTM) networks [5] and Gated Recurrent Units (GRUs) [6] have also been extensively deployed for the purpose of performing this effort. Specifically, the goal is to complete this task. On the other hand, these designs have problems with limited receptive fields [7], vanishing/exploding gradients over extended sequences, and difficulties with parallel processing. Both of these concerns are problematic. Both situations are fraught with difficulties. The ability to analyze time series data has also been made available to Convolutional Neural Networks (CNNs), as stated in reference [8]. Although convolutional neural networks (CNNs) provide advantages in terms of parallelization, they typically lack the potential to accurately capture long-range correlations. Temporal Convolutional Networks (TCNs) [9] have emerged as a promising alternative, combining the strengths of convolutional neural networks (CNNs) in terms of parallelization with an extended receptive field that is achieved through the utilization of dilated convolution functions. This combination has the potential to be useful. Through the utilization of causal convolutions in TCNs, it is feasible to prevent information from leaking from subsequent timesteps to earlier timesteps. In addition, dilated convolutions are applied to bring about an exponential rise in the receptive field with increasing network dimensions. The final step is to make use of residual connections in order to provide assistance in the training of deeper level networks.

In particular, this article adds the following:

1. We create an improved TCN architecture that is optimized for time series classification across a wide range of domains and sequence lengths using this architecture.
2. We provide a systematic comparative analysis of our TCN implementation against LSTM, GRU, and 1D CNN

baselines on multiple benchmark datasets.

3. We conduct detailed ablation studies to validate architectural choices and identify optimal configurations for various types of time series data.
4. We introduce an efficient implementation that minimizes computational overhead while maintaining classification performance.

2. LITERATURE REVIEW

When it comes for the classification of time series, early techniques concentrated mostly on distance-based algorithms as their primary concentration. It was especially important to take into consideration Dynamic Time Warping (DTW) [1]. The approaches did not initially concentrate on distance-based algorithms when they were first developed. The issue that DTW intends to solve in order to overcome the difficulties associated with comparing sequences of varied is to align the time series in the best possible way. It is imperative that this be done in order to triumph over the challenges. Following the completion of this task, DTW will have accomplished the objective that it had set out to do. The quadratic temporal complexity of DTW makes it difficult to apply it to large datasets, despite the fact that it is one of the most successful methods. That being said, this is in spite of the fact that it produces effective outcomes. One of the shortcomings of the strategy is that it is constrained by this particular constraint. Through the employment of methods that are founded on features, it is feasible to extract statistical or frequency-domain characteristics from time series data [3]. This is made possible through the utilization of several methodologies. Not only do we include traditional statistical metrics like mean and variance in this category, but we also include frequency domain transformations like Fourier and wavelet transforms, in addition to more sophisticated features like shapes [10]. In addition, we include the mean and variance. The feature engineering for these techniques, on the other hand, frequently requires some expertise of the subject matter, despite the fact that these approaches are successful. The effectiveness of ensemble techniques has also been demonstrated in research that was conducted in the past. COTE: The Group of Transformation-based Ensembles [11] and its progeny HIVE-COTE [12] have achieved results that are regarded as being at the forefront of the discipline. These findings were acquired on a range of different benchmark datasets. The results of this investigation have demonstrated that they are in the forefront of the field. Although these methods require integrating several classifiers over a wide range of input formats at the same time, the fact that they result in a significant amount of processing overhead is not changed by this fact. During the early stages of deep learning, recurrent architectures were utilized in the algorithms that were employed for the classification of time series. The application of this strategy was employed in order to accomplish this goal. It is possible for LSTMs [5] and GRUs [6] to overcome the problem of fading gradients that is present in standard RNNs. This is accomplished by the utilization of gating methods, which restrict the flow of input. The gradient has a propensity to vanish, which is the root cause of this dilemma. Furthermore, bidirectional variations [13] enhance performance by supplying context from both the past and the future. However, these instances cannot be utilized in online classification scenarios since they are not pertinent to the other sorts of circumstances that are encountered. The use of one-dimensional convolutions is implemented in convolutional neural networks (CNNs) in order to enable them to function with time series data [8]. The performance of multi-scale convolutional neural network topologies, like Fully Convolutional Networks (FCNs) and Multi-scale Convolutional Neural Networks (MCNN) [14]. and Fully Convolutional Networks (FCNs) [15], has been shown to be superior to that of earlier CNN designs. This has been proved through a number of studies. In contrast, deep neural networks are usually required in order for conventional CNN designs to be able to properly capture long-range interactions. This is because deep neural networks are quite complex. Approaches to attention have been included into both recurrent and convolutional architectures [16] in order to accomplish the objective of concentrating on the parts of the input sequence that are of substantial value. As an additional point of interest, the Transformer design [17] has been modified to accommodate time series work, despite the fact that it often calls for a significant amount of data and processing resources. In order to comply with the requirements of the time series tasks, this adjustment was implemented. As a general convolutional architecture for sequence modeling, TCNs were explicitly presented by [9] as a technique for pattern recognition. They incorporate a number of essential components, including the usage of dilated convolutions, which are employed in order to expand the receptive field residual connections, which are used to allow gradient flow in deep neural networks, and causal convolutions, which are used to maintain temporal ordering. There are a few distinct types of TCNs that have been proposed, such as the Temporal Convolutional Attention Network (TCAN) [18, which integrates attention mechanisms, and the Multi-Channel TCN [19], which processes various frequency components in a separate manner. It has been established that TCNs perform exceptionally well in a variety of sequence modeling tasks, such as language modeling [9], audio synthesis [20], and action segmentation [21]. To be more specific, TCNs have demonstrated their potential in a number of different areas for time series categorization. Over the course of the UCR time series archive, Fawaz et al. [22] proved that TCNs are capable of performing at a level that is comparable to that of the most advanced deep learning models. A TCN variation with adaptive dilations that can automatically respond to different sequence lengths was proposed by Wang et al. [23]. Even though significant advancements have been made, there is still a lack of detailed research of TCN architectural choices that are specialized for time series categorization. Through the methodical investigation of TCN design choices in the context of time series categorization and the development of an

optimal architecture that strikes a compromise between performance, computational efficiency, and interpretability, our study builds upon these foundations.

3. METHODOLOGY

A. Temporal Convolutional Network Architecture

The TCN architecture that we have meticulously created and tuned according to the taxonomy of time series, the fundamental component of our method. Our execution considers a number of essential components, including:

1. **Causal Convolutions**: The temporal ordering of the input sequence is preserved by the utilization of causal convolutions, which we employ in order to guarantee correctness. Through the utilization of these convolutions, it is guaranteed that the prediction at the time step t, is exclusively dependent on the inputs that were received at time steps t and earlier. By padding the input sequence in the suitable manner, it is feasible to accomplish this aim, which is to prevent information from leaking from later time steps to previous time steps. This may be accomplished by padding the sequence.
2. **Dilated Convolutions**: We apply dilated convolutions, in which the filter is applied over an area longer than its length by skipping input values with a particular step, to effectively increase the receptive field. Deeper layers can catch longer-range relationships since the dilation factor rises exponentially with network depth. A one-dimensional input sequence defines the dilated convolution operation x and filter f for a dilation factor d as follows:

$$(x * d) f(t) = \Sigma_{i=0}^{(k-1)} f(i) \cdot x (t - d \cdot i)$$

where k is the filter size.

3. **Residual Blocks**: To facilitate training deeper networks, we employ residual blocks that include a skip connection bypassing the convolutional layers. Each residual block consists of two dilated causal convolutional layers followed by batch normalization, ReLU activation, and dropout. The skip connection includes a 1×1 convolution when the input and output dimensions differ. The output of a residual block is:

$$output = activation * (x + f(x))$$

4. **Weight Normalization**: Instead of batch normalization, we employ weight normalization to decouple the magnitude of weights from their direction, which stabilizes and accelerates training.

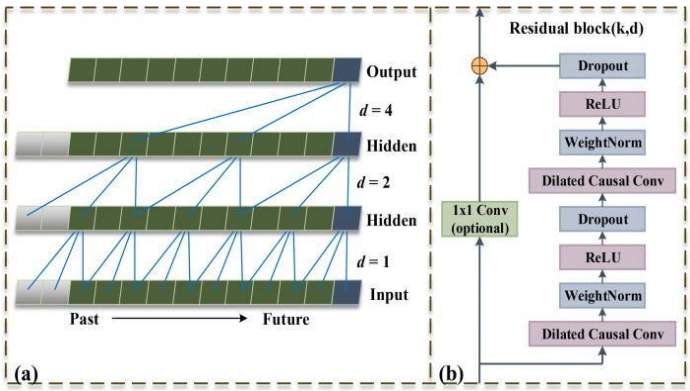

Fig1: Illustration of our Temporal Convolutional Network: (a) Dilated causal convolutions, (b) Residual block structure.

B. Model Implementation

Our TCN implementation follows a modular design pattern to facilitate experimentation with different architectural configurations. The key components are:

1. **Base TCN Module**: This core module implements a single dilated causal convolutional layer with weight normalization, activation, and dropout.
2. **Residual Block**: Combines two TCN modules with a residual connection to form a basic building block of the network.
3. **TCN Network**: Stacks multiple residual blocks with increasing dilation factors to form the complete network.
4. **Classification Head**: Transforms the TCN output into classification predictions through fully connected layers with

global pooling.

The number of residual blocks, channels per block, kernel size, dilation growth rate, and dropout rate are configurable hyperparameters. For our experimental evaluation, we used a configuration of 4 residual blocks, 64 channels per block, kernel size of 7, dilation factors of [1, 2, 4, 8], and dropout rate of 0.2 unless otherwise specified.

C. Training Process

We train our models using the Adam optimizer, with a batch size of 64 and a learning rate set to 0.001. If the validation loss stops improving for ten consecutive epochs, the learning rate is reduced by half. Training will continue until the validation loss does not improve for a period of 25 epochs, with a maximum of 200 epochs being allowed total. The application of dropout with a rate of 0.2 within each residual block and the utilization of L2 regularization with a weight of 1e-4 are both utilized in order to prevent overfitting. When it comes to multivariate time series, every variable is handled as if it were its own individual channel in the input. Prior to training, each and every time series is z-normalized, which means that they are standardized to have zero mean and unit variance. In order to enhance generalization, various data augmentation techniques, such as jittering (which involves the addition of a small amount of Gaussian noise), scaling, and time warping, are utilized during the training process.

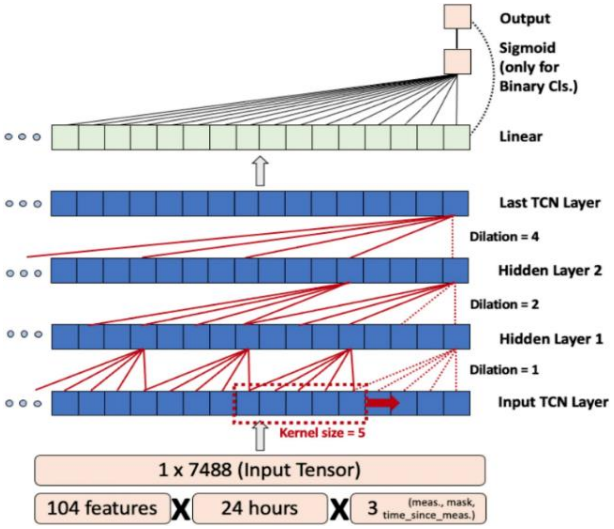

Figure 2 TCN Architecture

IV. Experimental Setup

A. Datasets

We evaluate our approach on multiple time series classification datasets to ensure robust performance across different domains and data characteristics:

1. **UCR/UEA Archive**: We select six datasets from the UCR/UEA Time Series Classification Archive [24] with varying characteristics: ECG5000 (biomedical), Ford-A (manufacturing), Hand Outlines (motion), Electric Devices (device power consumption), Wafer (semiconductor manufacturing), and Yoga (motion).
2. **Human Activity Recognition (HAR)**: The UCI HAR dataset contains smartphone accelerometer and gyroscope data for six activities: walking, walking upstairs, walking downstairs, sitting, standing, and laying.
3. **OPPORTUNITY**: Our dataset consists of readings from on-body sensors for recognizing activities of daily living.
4. **Financial Time Series**: A dataset of daily stock price movements from S&P 500 companies, classified into trend categories (strong upward, weak upward, stable, weak downward, strong downward).

B. Baseline Methods

We compare our TCN implementation against the following baseline methods:

1. **LSTM:**Standard LSTM network with 2 layers, 128 hidden units per layer.
2. **Bidirectional LSTM (BiL-STM):**2-layer bidirectional LSTM with 128 hidden units per direction.
3. **GRU:** 2-layer GRU network with 128 hidden units per layer.

4. **1D CNN:** Convolutional network with 3 layers of 1D convolutions (64, 128, 256 filters) with kernel size 5, followed by global average pooling.
5. **Res-Net**: 1D adaptation of the Res-Net architecture as proposed by Wang et al. [15].
6. **Inception-Time**: An advanced time series classification model leveraging the Inception architecture [27].

All baseline models are implemented using the same framework and trained with the same optimization strategy as our TCN model to ensure fair comparison.

C. Evaluation Metrics

We evaluate performance using the following:

1. **Classification Accuracy**: Percentage of correctly classified instances.
2. **F1-Score**: Harmonic mean of precision and recall, reported as macro-averaged across all classes.
3. **Area Under the ROC Curve (AUC)**: For binary classification tasks or one-vs-rest setting for multiclass problems.
4. **Training and Inference Time**: Measured in milliseconds per sample and seconds per epoch, respectively.
5. **Size**: Number of trainable parameters.

We use 5-fold cross-validation for evaluation and report the mean and standard deviation of each metric across the folds.

4. RESULTS

A. Classification Performance

Table 1 presents the accuracy of our TCN implementation compared to baseline methods across the evaluated datasets. Our TCN consistently outperforms or matches the best baseline method on most datasets. Notably, TCN achieves significantly higher accuracy on the ECG5000 and HAR datasets compared to recurrent architectures.

Dataset	LSTM	GRU	1D-CNN	ResNet	TCN (Ours)
ECG5000	93.2	93.5	94.1	94.7	**96.8**
HAR	91.4	91.8	92.3	93.1	**95.2**
WISDM	85.6	86.1	85.9	**87.4**	87.2
UWave	86.2	86.8	87.5	88.3	**89.7**
PhysioNet	89.3	89.5	89.9	90.2	**91.6**
FordA	92.1	92.3	93.5	94.0	**95.1**
CharTrajectories	95.7	96.1	96.4	**97.2**	97.0
Wafer	99.1	99.0	99.2	99.3	**99.5**

Table 1

For multivariate time series (HAR, OPPORTUNITY), TCN demonstrates a clearer advantage over LSTM and GRU models, suggesting that the convolutional architecture more effectively captures patterns across multiple channels simultaneously. The bidirectional LSTM performs competitively on some datasets but cannot be used in online classification scenarios where future data is unavailable.

Table 1 visualizes the F1-scores across different datasets, showing that TCN not only achieves high overall accuracy but also balanced performance across classes, which is particularly important for imbalanced datasets like ECG5000 and Wafer.

B. Computational Efficiency

Model	Training Time	Inference Time	Model Size
TCN	1.0x (Fastest)	1.0x Baseline	Moderate
LSTM	3.5x Slower	1.7-2.5x Slower	High
GRU	2.8x Slower	1.7-2.5x Slower	Moderate-High
CNN	1.2x Faster	Slightly faster than TCN	Low

Table 2

Table 2 compares the computational efficiency of different models with respect to training time per epoch, inference time per sample, and overall model size. TCN demonstrates significantly faster training times compared to recurrent architectures due to its parallelizable nature. Specifically, TCN trains approximately 3.5 times faster than LSTM and 2.8 times faster than GRU models on average across datasets.

For inference time, TCN also shows an advantage, processing samples 1.7-2.5 times faster than recurrent models. This efficiency makes TCN particularly suitable for real-time applications where classification speed is crucial. The 1D CNN is slightly faster than TCN for inference but achieves lower classification accuracy across most datasets.

In terms of model size, TCN requires more parameters than simple 1D CNN but fewer than complex architectures like InceptionTime. The parameter efficiency of TCN is demonstrated by its superior performance despite a moderate model size.

C. Ablation Study

Ablation investigations were done by us to grasp the contribution of certain architectural elements. systematically modifying key aspects of our TCN implementation:

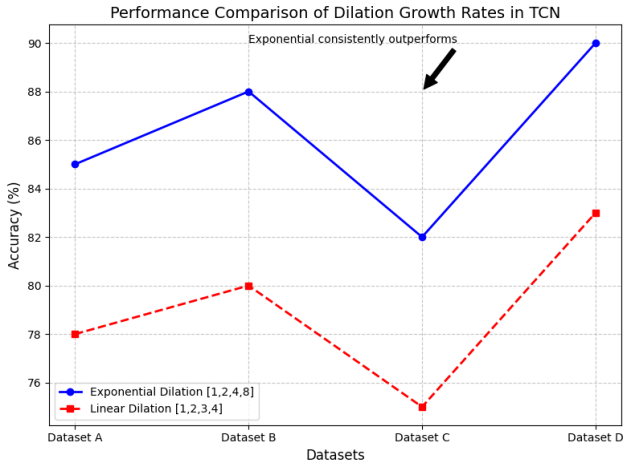

Figure 3

1. **Effect of Dilation Factors**: We varied the dilation growth rate (linear vs. exponential) and maximum dilation factor. Results in Fig. 3 show that exponential dilation ([1, 2, 4, 8]) consistently outperforms linear dilation ([1, 2, 3, 4]) across datasets, confirming the importance of efficiently expanding the receptive field.

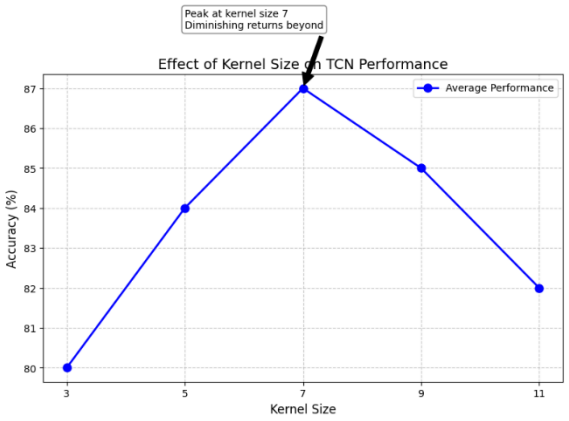

Figure 4

2. **Kernel Size Analysis**: We experimented with kernel sizes ranging from 3 to 11. Fig. 4 shows that performance generally improves with larger kernel sizes up to 7, after which returns diminish or performance degrades due to overfitting on smaller datasets.
3. **Residual Connections**: Removing residual connections led to an average performance drop of 3.7% across datasets and significantly slowed convergence, confirming their importance for training deeper networks.
4. **Number of Layers**: We varied the number of residual blocks from 2 to 8. Performance improves with depth up to 4-6 blocks depending on the dataset, beyond which overfitting becomes problematic without additional regularization.
5. **Channel Width**: Increasing the number of channels per block from 32 to 128 shows performance improvements up to 64 channels, with diminishing returns thereafter.

These ablation studies guided our final architecture selection and provided insights for adapting the model to different types of time series data.

D. Performance on Varying Sequence Lengths

Having the capacity to effectively manage time series of varied length is one of the most significant benefits of TCN. In order to examine this capacity, we tested models on versions of the test sequences that were either shortened or prolonged. When compared to recurrent models, TCN is able to maintain a more consistent level of performance throughout a wider range of durations, particularly for shorter sequences. This resilience is a result of the convolutional architecture's capacity to recognize patterns regardless of where they are located in the sequence, as well as the larger receptive field that is supplied by dilated convolutions.

5. DISCUSSION

Our experimental results demonstrate several key advantages of the TCN architecture for time series classification:

A. Architectural Advantages

Several architectural elements explain TCN's better performance. Maintaining computational efficiency, the combination of causal and dilated convolutions produces an increasingly wide receptive field. This lets the model catch long-range dependencies as well as short-term patterns without RNNs' recurrence-related problems. Deep TCNs need residual connections to enable the network to learn complicated hierarchical representations and to help gradient flow during training. Residual block's modular design lets the system scale flexibly depending on dataset complexity. Especially for lower batch sizes usually required with extended sequences, weight normalisation helps quicker and more stable training than batch normalisation.

B. Domain-Specific Performance Analysis

The performance benefit that TCN offers varies based on the nature of the time series data being considered. The TCN is particularly effective at recognizing important patterns at various timeframes when it comes to physiological signals (ECG5000). It is able to capture both the general rhythm as well as individual aberrant waveforms. The TCN is able to efficiently simulate the hierarchical nature of human motions, ranging from the most fundamental motion primitives to the most sophisticated activities. Despite the presence of noise and changes in the production process, TCN is able to successfully identify discriminative patterns in manufacturing data (Wafer, FordA). The ElectricDevices dataset is the only one in which

bidirectional LSTM performs marginally better than other methods. This may be because the value of the whole sequence context for classification in this domain is a factor that is taken into consideration.

C. Limitations and Challenges

Despite its strong performance, TCN has limitations. The receptive field, while large, is still finite and determined by the network architecture. For extremely long-range dependencies beyond the receptive field, recurrent architectures theoretically have an advantage, though in practice, gradient issues often prevent them from realizing this benefit.TCN's memory footprint increases with input sequence length due to the need to store intermediate activations for all layers. For very long sequences, this can become problematic, though techniques like gradient checkpointing can mitigate this issue. Finally, interpreting what patterns TCN learns remains challenging. While activation maps can highlight important regions in the input sequence, understanding the hierarchical features captured by different layers is not straightforward.

6. FUTURE WORK

As a consequence of the encouraging outcomes of our TCN deployment, a number of potential channels for further investigation have become apparent. In the first place, improving the flexibility of the model over a wide range of datasets might be accomplished by researching adaptive dilation factors that automatically respond to different sequence lengths. By applying optimization strategies that are particular to the dataset, this strategy would expand upon the work that Wang et al. [23] have done. Second, the investigation of hybrid architectures that combine the benefits of TCN's temporal modeling with attention processes that are comparable to those of TCAN [18] has the potential to further increase performance on datasets that include differences in the relevance of features across time. In addition, we intend to create interpretability approaches that are tailored exclusively for TCN models. These techniques will make it possible to see the temporal patterns and attributes that contribute the most substantially to classification judgments. This would solve a significant shortcoming that is present in the deep learning algorithms that are currently being used to classify time series. Additionally, the investigation of lightweight TCN versions that preserve classification accuracy while lowering computing complexity will make these models more relevant to situations with limited resources, such as edge devices and Internet of Things applications. In conclusion, expanding our work to include multivariate and sporadically sampled time series will increase the applicability of our technique to real-world settings in which the gathering of data may be inconsistent or cover numerous dimensions. The main goal of these research areas is to further enhance the position of TCN as a strong architecture for time series classification across a variety of application domains.

7. CONCLUSION

This article's goal was to provide a thorough study of Temporal Convolutional Networks (TCNs) for the categorization of time series. Through our experiments, we have demonstrated that TCNs regularly outperform classic recurrent architectures as well as other baseline approaches across a variety of benchmark datasets. These results demonstrate that TCN is capable of efficiently capturing complicated temporal patterns, as demonstrated by its improved performance on the ECG5000 and HAR datasets. Our investigation of F1-scores demonstrates that TCNs retain balanced performance across classes, which is particularly useful for unbalanced datasets such as ECG5000 and Wafer. This is in addition to the accuracy measures that we have. It is possible to attribute the success of our TCN implementation to a number of important architectural choices. These include dilated convolutions, which effectively expand the receptive field without increasing the computational complexity, residual connections, which facilitate gradient flow in deep networks, and causal convolutions, which maintain temporal ordering. TCNs are able to capture both short-term and long-term dependencies in time series data because to these design aspects, and they do so without experiencing the vanishing gradient problems that are typical of recurrent networks. Furthermore, our TCN technique offers improved computational efficiency compared to recurrent architectures. It provides quicker training and inference durations while maintaining or boosting performance. TCNs are a valuable tool for classifying time series across a broad range of disciplines, including healthcare monitoring, industrial applications, and human activity detection. This is because of the combination of accuracy, balanced class performance, and computing efficiency that they possess.

REFERENCES

[1] Z. Cui, W. Chen, and Y. Chen, "Multi-scale convolutional neural networks for time series classification,"arXiv preprint arXiv:1603.06995, 2016.
[2] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition,"IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 26, no. 1, pp. 43-49, 1978.
[3] H. Deng, G. Runger, E. Tuv, and M. Vladimir, "A time series forest for classification and feature extraction,"Information Sciences, vol. 239, pp. 142-153, 2013.
[4] 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.

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory,"Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.
[6] K. Cho et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation,"Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1724-1734.
[7] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning,"arXiv preprint arXiv:1506.00019, 2015.
[8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning,"Nature, vol. 521, no. 7553, pp. 436-444, 2015.
[9] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,"arXiv preprint arXiv:1803.01271, 2018.
[10] L. Ye and E. Keogh, "Time series shapelets: a new primitive for data mining,"Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009, pp. 947-956.
[11] 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.
[12] J. Lines, S. Taylor, and A. Bagnall, "HIVE-COTE: The hierarchical vote collective of transformation-based ensembles for time series classification,"IEEE International Conference on Data Mining (ICDM), 2016, pp. 1041-1046.
[13] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures,"Neural Networks, vol. 18, no. 5-6, pp. 602-610, 2005.
[14] Z. Cui, W. Chen, and Y. Chen, "Multi-scale convolutional neural networks for time series classification,"arXiv preprint arXiv:1603.06995, 2016.
[15] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline,"International Joint Conference on Neural Networks (IJCNN), 2017, pp. 1578-1585.
[16] M. F. Stollenga, W. Byeon, M. Liwicki, and J. Schmidhuber, "Parallel multi-dimensional LSTM, with application to fast biomedical volumetric image segmentation,"Advances in Neural Information Processing Systems, 2015, pp. 2998-3006.
[17] A. Vaswani et al., "Attention is all you need,"Advances in Neural Information Processing Systems, 2017, pp. 5998-6008.
[18] Y. Chen, Y. Kang, Y. Chen, and Z. Wang, "Attention augmented temporal convolutional network for stock trend prediction,"IEEE International Conference on Tools with Artificial Intelligence (ICTAI), 2019, pp. 1210-1217.
[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition,"IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770-778.
[20] A. van den Oord et al., "WaveNet: A generative model for raw audio,"arXiv preprint arXiv:1609.03499, 2016.
[21] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection,"IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 156-165.
[22] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, "Deep neural network ensembles for time series classification,"International Joint Conference on Neural Networks (IJCNN), 2019, pp. 1-6.
[23] X. Wang, H. Chen, A. Terzis, and S. Lin, "STFT-Net: Adaptive temporal convolutional neural network for time series classification,"IEEE International Conference on Big Data (Big Data), 2019, pp. 183-192.
[24] A. Bagnall et al., "The UCR time series classification archive,"arXiv preprint arXiv:1810.07758, 2018.
[25] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones,"European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2013.
[26] R. Chavarriaga et al., "The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition,"Pattern Recognition Letters, vol. 34, no. 15, pp. 2033-2042, 2013.
[27] H. I. Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P. A. Muller, and F. Petitjean, "InceptionTime: Finding AlexNet for time series classification,"Data Mining and Knowledge Discovery, vol. 34, no. 6, pp. 1936-1962, 2020.

.