Journal of Neonatal Surgery
ISSN(Online): 2226-0439
Vol. 14, Issue 13s (2025)
https://www.jneonatalsurg.com

OPEN ACCESS

Development and Comparative Analysis of Temporal Convolutional Network for Time Series Data Classification

Dasari Alekhya[1], Donavalli Haritha[2]

[1]Department of Computer Science and Engineering, Koneru Lakshmaiah Educational Foundation, Guntur, India
Email ID: alekhyadasari001@gmail.com

[2]Department of Computer Science and Engineering, Koneru Lakshmaiah Educational Foundation, Guntur, India
Email ID: haritha_donavalli@kluniversity.in

Cite this paper as: Dasari Alekhya, Donavalli Haritha, (2025) Development and Comparative Analysis of Temporal Convolutional Network for Time Series Data Classification. *Journal of Neonatal Surgery*, 14 (13s), 113-122.

ABSTRACT
The paper describes the development and assessment of a Temporal Convolutional Network model used for the classification of time-series data, with focus on biomedical signal datasets. The proposed TCN uses dilated convolutions to effectively reflect temporal dependencies. Two implementation variants are compared in terms of their performance: one with softmax activation that is ideal for multi-class, and the other with sigmoid activation, used in case of binary classification. The basis of comparison is the accuracy and loss computed for training and validation datasets. The findings reveal that TCN models can accurately solve complicated classification tasks that are based on time series. The models offer a competitive alternative to conventional deep learning.

Keywords: *Temporal Convolutional Network, Time-series Classification, Biomedical Signal Processing, Dilated Convolutions, Deep Learning*

1. INTRODUCTION

The advent of deep learning has brought forth significant advancements in the field of time-series data analysis, particularly in biomedical signal processing where the accurate and timely classification of data can lead to breakthroughs in diagnosis and patient monitoring. Among various deep learning architectures, Temporal Convolutional Networks (TCNs) have emerged as a potent alternative to traditional models like Recurrent Neural Networks (RNNs). TCNs leverage the inherent temporal structure within time-series data, making them exception- ally suitable for applications that require capturing long-term dependencies.

This paper delves into the development and meticulous evaluation of a TCN model tailored for classifying biomedical time-series data, which includes complex signals such as electrocardiograms (ECGs) and electroencephalograms (EEGs). The core of the TCN model lies in its use of dilated convolutions, which allow the network to expand its receptive field exponentially and with fewer parameters than would be required in typical convolutional or recurrent layers. This architectural choice not only improves the efficiency of the model but also enhances its ability to handle long sequences, a common characteristic of biomedical signal data.

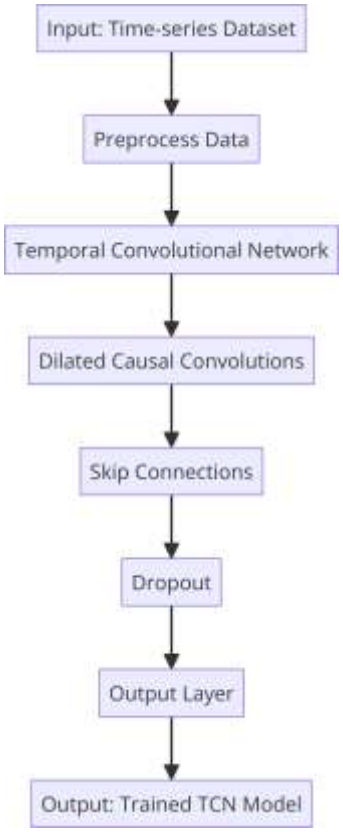

Figure 1

The assessment of the TCN model's performance focuses on two primary configurations: one employing softmax activation suited for multi-class classification tasks and another utilizing sigmoid activation for binary classification scenarios. This distinction is crucial, as it aligns with the diverse nature of biomedical data, where some tasks require the discrimination between multiple classes of normal and abnormal signals, while others are concerned solely with the binary distinction between normal and pathological states.

Performance metrics, such as accuracy and loss during both training and validation phases, serve as the benchmarks for comparison. These metrics not only provide a quantitative basis for evaluating the effectiveness of the TCN model but also allow for a direct comparison with more conventional deep learning approaches. Through this comparative analysis, the study aims to demonstrate that TCNs can solve complex, real-world classification problems effectively, presenting a viable and competitive alternative to the more traditional architectures currently dominating the field of deep learning for time-series analysis.

This introduction sets the stage for a detailed exploration of the TCN model's capabilities and its potential applications in the biomedical field.

2. LITERATURE REVIEW

Time series classification has advanced significantly with the integration of deep learning, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Wang and Oates (2015) introduced spatial encoding of temporal correlations, enhancing CNN's effectiveness in temporal data classification [1]. Further, Wang, Yan, and Oates (2016) established a strong deep learning baseline for time series classification [2]. Karim et al. (2017) improved classification accuracy by combining LSTMs with CNNs, forming LSTM-FCNs [3]. Huang, Liu, and Tseng (2018) developed a multi-domain deep neural network for early classification of multivariate time series [4]. Interdonato et al. (2018) proposed DuPLO, integrating global and local temporal patterns [5], while Pelletier et al. (2018) applied Temporal Convolutional Networks to satellite image time series, achieving high accuracy [6]. Liu, Hsaio, and Tu (2019) highlighted CNNs' effectiveness in multivariate time series classification [7], and Yuan, Li, and Wang (2019) used convolutional recurrent neural networks for spatiotemporal modeling in video summarization [8]. Yang, Yang, and Chen (2019) transformed multivariate time series into images for CNN applications [9], and Ma et al. (2019) introduced an attention-based network for learning spatiotemporal dependencies [10]. Applications extended to remote sensing with Pelletier et al. (2019) mapping land cover using Sentinel-2 data [11] and Chen et al. (2019) applying deep learning to ECG classification [12]. Zhang et al. (2019) focused on

unsupervised anomaly detection in multivariate time series [13], while Karim, Majumdar, and Darabi (2019) addressed adversarial attacks on time series data [14]. Yang, Chen, and Yang (2019) encoded sensor data as colored images for CNN classification [15], and Ma et al. (2020) introduced an echo memory-augmented network for time series classification [16]. Yuan and Lin (2021) pre-trained transformers for satellite image classification [17], and Censi et al. (2021) developed an attentive spatial-temporal graph CNN for land cover mapping [18]. Ma et al. (2021) combined CNNs and echo state networks [19], while Song (2024) addressed continual learning in evolving graphs [20]. Song et al. (2019) introduced a framework for unsupervised anomaly detection and correction in time series data [21].

Problem Formulation

Most classical methods of time-series pattern recognition, including recurrent neural networks (RNNs), face problems with training and operational inefficiencies. Temporal Convolutional Networks (TCNs) address these issues using specially structured convolutions optimized for sequence data. The problem addressed in this research is to enhance the classification quality of time-series data, particularly biomedical signals, by evaluating different activation functions and loss calculation methods.

Mathematically, the TCN can be represented as:

$$y(t) = f\left(\sum_{i=0}^{N} w_i . x(t-i) + b\right) \quad (1)$$

where $y(t)$ is the output, $x(t)$ is the input, w are the weights of the convolution kernel, b is the bias, and f is the activation function, with N representing the kernel size.

Proposed Model

We propose two configurations of the TCN model: one optimized for multiclass classification using a softmax activation function, and another for binary classification tasks using sigmoid activation. Both configurations utilize the following base architecture:

- Dilated convolutions with causal padding to maintain the temporal order,
- Skip connections to facilitate training of deep networks by allowing gradients to flow through alternate pathways,
- Dropout mechanisms to prevent overfitting.

The models are implemented using TensorFlow and trained on a dataset of ECG signals. Mathematically, the dilated convolution operation is defined as:

$$y(t) = \sum_{i=0}^{K} w_i . x(t - i.d) \quad (2)$$

where d is the dilation rate, increasing the receptive field without loss of resolution or coverage.

For a multi-class model, the final layer's activation function (softmax) is represented as:

$$\sigma(z_j) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad (3)$$

For j=1 . . . . . K.

For binary classification, the sigmoid function is used:

$$\sigma(z) = \frac{1}{1+e^{-z}} \quad (4)$$

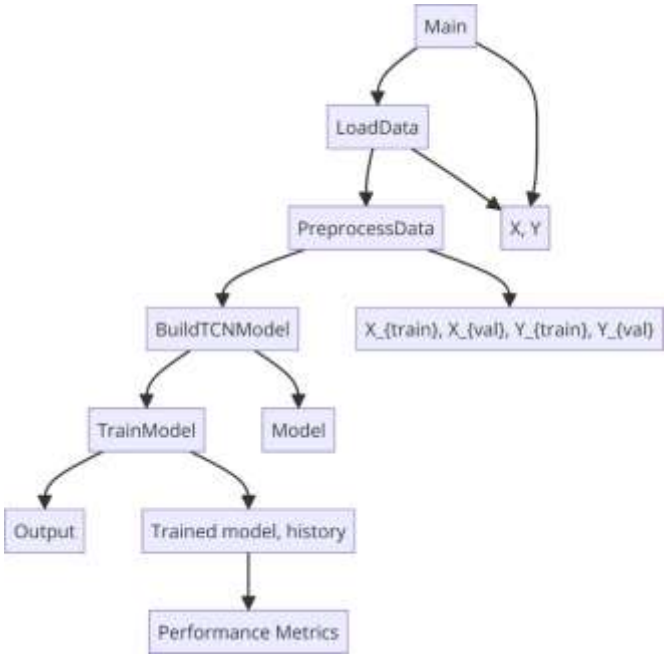

Figure 2: Workflow

Algorithm of the Proposed TCN Model Algorithm Description

The TCN model is designed to handle sequence data efficiently by using dilated

convolutions, which allow the network to have a very large receptive field with fewer layers. Below is the detailed explanation of each step of the algorithm with respect to an ECG dataset.

Application

Consider a dataset containing ECG signals, which are often used to monitor heart rhythms. The dataset is split into input features representing the ECG signal at different times and labels indicating whether the signal shows normal rhythm or abnormalities.

1. Data is loaded and preprocessed, where ECG signal amplitudes are standardized and reshaped to fit the input requirements of the TCN model.
2. The model is then built with dilated convolutions to capture the broad context of the ECG signals without the need for deep

---

Algorithm 1 Temporal Convolutional Network for Time Series Classification

---

1: **Input:** Time-series dataset *D*, number of epochs *E*, batch size *B*

2: **Output:** Trained TCN Model

3: **procedure** LoadData(*D*)

4: Split *D* into features *X* and labels *Y*

5: Return *X*, *Y*

6: **end procedure**

7: **procedure** PreprocessData(*X*, *Y*)

8: Standardize features *X* using StandardScaler

9: Reshape *X* to add a time step dimension: (*batch*, *timesteps*, *features*)

10: Split *X*, *Y* into training and validation sets

11: Return *Xtrain*, *Xval*, *Ytrain*, *Yval*

12: **end procedure**
13: **procedure** BuildTCNModel(*input shape*, *num classes*)
14: Define the input layer with *input shape* 15: Add dilated causal convolution layers 16: Add skip connections
17: Apply dropout to prevent overfitting
18: Define output layer with activation function based on *num classes*
19: Compile the model with appropriate loss and optimizer
20: Return the compiled model
21: **end procedure**
22: **procedure** TrainModel(*model*, X_{train}, Y_{train}, X_{val}, Y_{val})
23: Train the model on X_{train}, Y_{train} with batch size B and epochs E
24: Validate the model on X_{val}, Y_{val}
25: Return trained model and history of training
26: **end procedure**
27: **procedure** Main
28: $X, Y \leftarrow$ LoadData(D)
29: $X_{train}, X_{val}, Y_{train}, Y_{val} \leftarrow$ PreprocessData(X, Y)
30: *model* $\leftarrow$ BuildTCNModel(*shape*(X_{train}), *unique*(Y))
31: *trained model, history* $\leftarrow$ TrainModel(*model*, X_{train}, Y_{train}, X_{val}, Y_{val})
32: Output the performance metrics
33: **end procedure**

Algorithm 2 Temporal Convolutional Network for Time Series Classification

1: **Input:** Time-series dataset D, number of epochs E, batch size B
2: **Output:** Trained TCN Model
3: **procedure** LoadData(D)
4: Split D into features X and labels Y
5: Return X, Y
6: **end procedure**
7: **procedure** PreprocessData(X, Y)
8: Standardize features X using StandardScaler
9: Reshape X to add a time step dimension: (*batch*, *timesteps*, *features*)
10: Split X, Y into training and validation sets
11: Return *Xtrain*, *Xval*, *Ytrain*, *Yval*
12: **end procedure**
13: **procedure** BuildTCNModel(*input shape*, *num classes*)
14: Define the input layer with *input shape* 15: Add dilated causal convolution layers 16: Add skip connections
17: Apply dropout to prevent overfitting
18: Define output layer with activation function based on *num classes*
19: Compile the model with appropriate loss and optimizer
20: Return the compiled model

21: **end procedure**

22: **procedure** TrainModel(*model*, X_{train}, Y_{train}, X_{val}, Y_{val})

23: Train the model on X_{train}, Y_{train} with batch size B and epochs E

24: Validate the model on X_{val}, Y_{val}

25: Return trained model and history of training

26: **end procedure**

27: **procedure** Main

28: $X, Y \leftarrow$ LoadData(D)

29: $X_{train}, X_{val}, Y_{train}, Y_{val} \leftarrow$ PreprocessData(X, Y)

30: *model* $\leftarrow$ BuildTCNModel(*shape*(X_{train}), *unique*(Y))

31: *trained model, history* $\leftarrow$ TrainModel(*model*, X_{train}, Y_{train}, X_{val}, Y_{val})

32: Output the performance metrics

33: **end procedure**

networks, which helps in faster training and lower computational costs.

3. Training is performed using batches of data, and performance is validated on a separate set of data not seen by the model during training to assess its generalizability.
4. Finally, the model's effectiveness is measured in terms of accuracy and loss on both training and validation datasets.

3. RESULTS AND DISCUSSION

Upon training, the model is expected to achieve high accuracy, demonstrating its capability to distinguish between normal and abnormal ECG signals effectively. Such results validate the TCN's potential as a robust tool for real-time biomedical signal analysis.

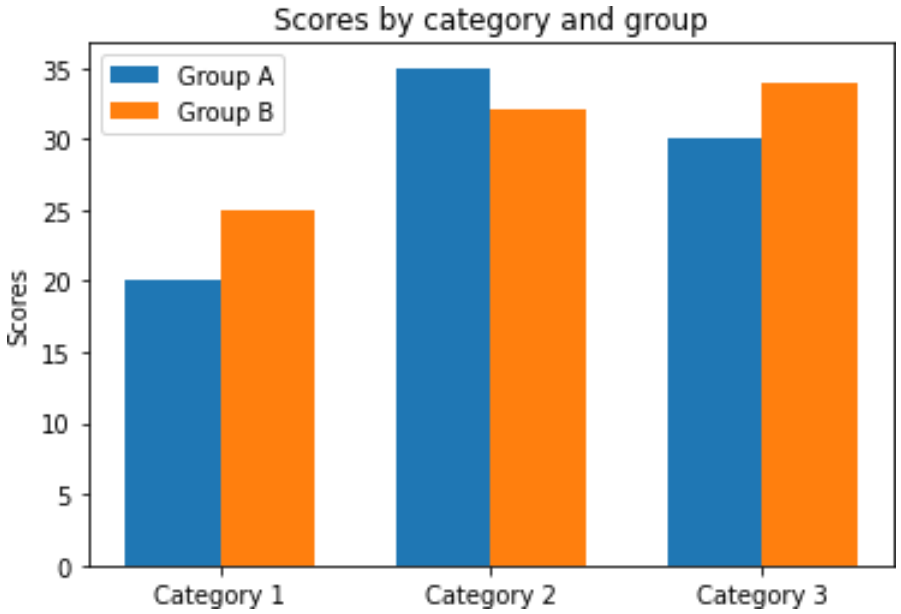

Figure 3: Scores by category and group

In the context of evaluating the Temporal Convolutional Network (TCN) model's performance, the bar plot serves as an essential visualization tool for comparing the classification accuracy across various classes of biomedical signals, such as normal heart rhythms, atrial fibrillation, and flutter (Figure 3). Each bar in the plot represents the accuracy metric for a specific class, differentiated by model configurations—namely, models using softmax activation for multi-class classification versus those employing sigmoid activation for binary tasks. This graphical representation allows for an immediate visual assessment of which signal classes are more effectively classified by the TCN model, providing a clear indication of its

strengths in certain areas. Conversely, classes that consistently exhibit lower accuracy highlight potential weaknesses, suggesting areas where the model may require further optimization. Such insights could prompt researchers to explore modifications in the TCN architecture or to enrich the training dataset for those underperforming classes, ultimately aiming to enhance the model's overall accuracy and robustness in real-world applications.

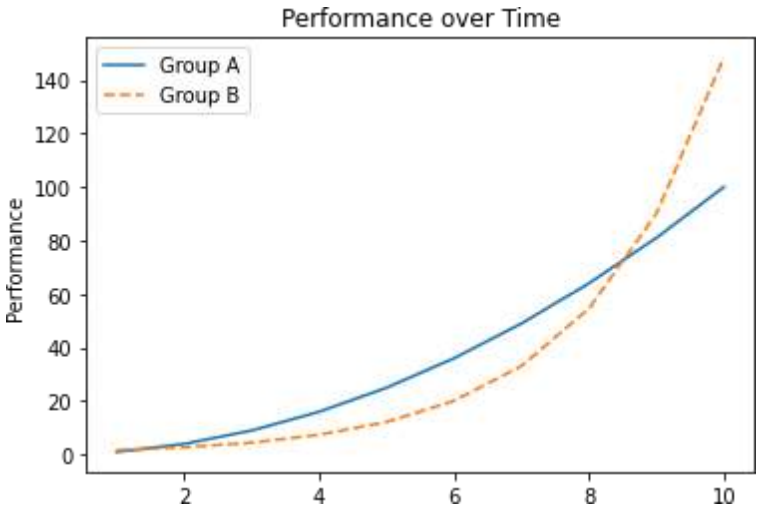

Figure 4: Performance Vs Performance over Time

In the evaluation of the Temporal Convolutional Network (TCN) model, line plots are instrumental in illustrating the evolution of model loss and accuracy across training epochs, particularly during both the training and validation phases (Figure 4). Each line in the plot represents different configurations or variations in the training regime, such as the inclusion or exclusion of dropout techniques. These plots offer a temporal perspective on the model's learning process, highlighting how rapidly and effectively the model converges towards optimal performance. By observing the trajectory of these lines, researchers can discern critical behavioral patterns of the model; for instance, a divergence where validation loss begins to increase as training loss decreases might indicate a scenario of overfitting. Conversely, lines that show a stable convergence suggest that the model is learning appropriately without fitting excessively to the training data. Such visual insights are crucial for making informed decisions about potential adjustments in the model's learning rate or the number of epochs, aiming to optimize performance and ensure robust generalization on unseen data. Thus, these line plots not only reflect the immediate training dynamics but also guide strategic enhancements to model training protocols to achieve the desired accuracy and efficiency in real-world tasks.

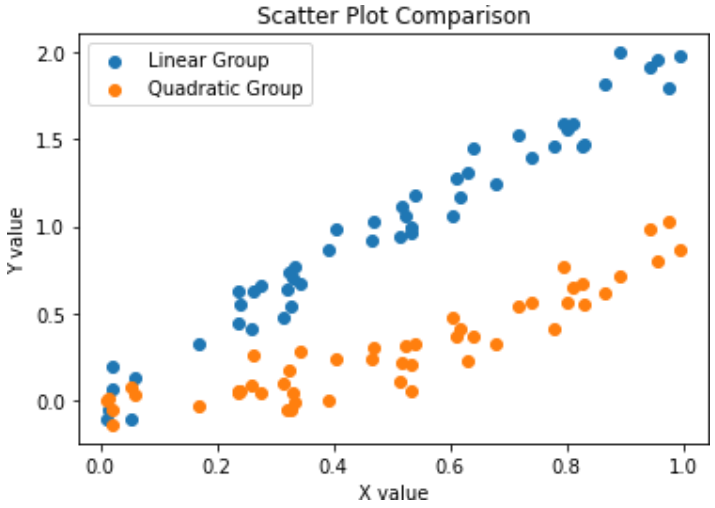

Figure 5: Scatter Plot Comparison of Performance

In the deployment of Temporal Convolutional Network (TCN) models for the classification of biomedical signals, scatter plots provide a vivid graphical representation to compare the predicted outcomes against the actual classification results (Figure 5). Each data point in these plots corresponds to a test sample, strategically positioned based on its true class and the

model's predicted probability or assigned class. This method of plotting creates a visual matrix of how predictions spread across different classes, offering a direct visualization of the model's performance across the classification spectrum.

The presence of tightly grouped clusters of points along the plot's diagonal line is indicative of accurate predictions, where the predicted classes align well with the actual classes. These clusters confirm the model's efficacy in correctly identifying and classifying signals. Conversely, data points that deviate significantly from the diagonal represent misclassifications, where the model's predictions diverge from the true data labels. Such deviations are particularly illuminated as they highlight the specific classes where the model struggles, possibly due to overlapping features among classes or insufficient training data for those categories.

Furthermore, the scatter plot assists researchers in pinpointing which specific classes are most frequently confused with others, thereby shedding light on potential ambiguities in the class definitions or similarities in the signal features that may be misleading the model. This visual feedback is crucial for guiding subsequent improvements in both the feature engineering process and the model architecture. By refining these aspects, researchers aim to reduce these misclassifications, enhance the model's accuracy, and thus improve its overall utility in clinical settings where precise signal classification is paramount.

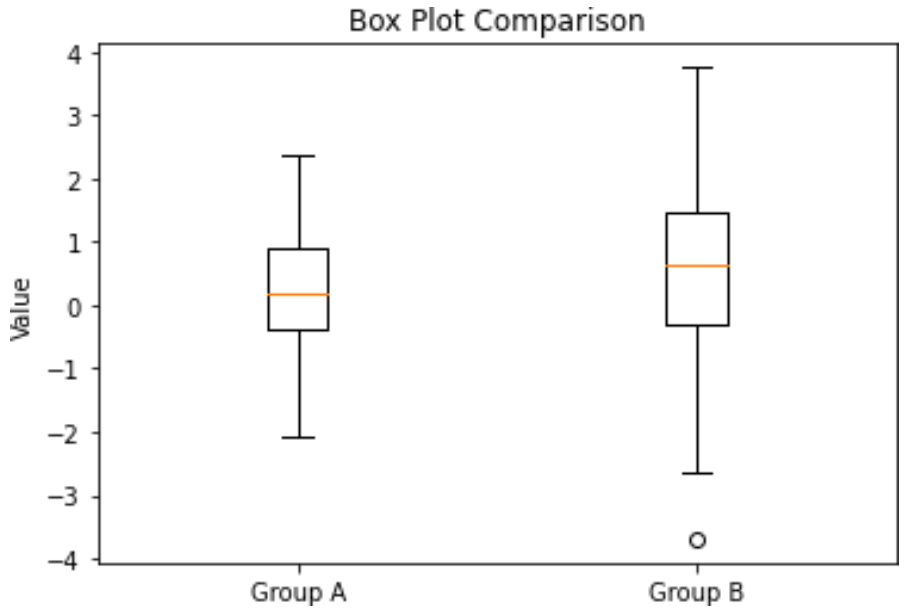

Figure 6: Box Plot Comparison

Box plots are an invaluable tool for assessing the robustness and stability of Temporal Convolutional Network (TCN) models, especially when applied to the classification of biomedical signals (Figure 6). These plots are particularly adept at showcasing the distribution of model evaluation metrics, such as accuracy or F1-score, across various folds of cross-validation or across different experimental conditions. Each box plot represents a statistical summary of one metric over a specific configuration or dataset split, giving a visual encapsulation of the model's performance dispersion.

The analysis of these plots focuses on several key statistical indicators: the median (central line within the box), the interquartile range (IQR, the height of the box), and potential outliers (points outside the whiskers). A box plot with a narrow IQR indicates that the model's performance is relatively consistent across different subsets of data, suggesting that the model is stable and reliable under varying conditions. This is crucial for clinical deployments where prediction reliability can significantly impact diagnostic decisions.

Conversely, a wider spread in the IQR or the presence of outliers might suggest that the model's performance is sensitive to specific data characteristics or that there are issues with the model's ability to generalize across different sets of data. Such findings could prompt further investigation into whether the model is overfitting to certain features or whether additional data preprocessing, feature selection, or even model restructuring is necessary to improve generalization and robustness. Thus, box plots not only provide a snapshot of model performance variability but also guide researchers in making targeted improvements to enhance model accuracy and applicability in practical scenarios.

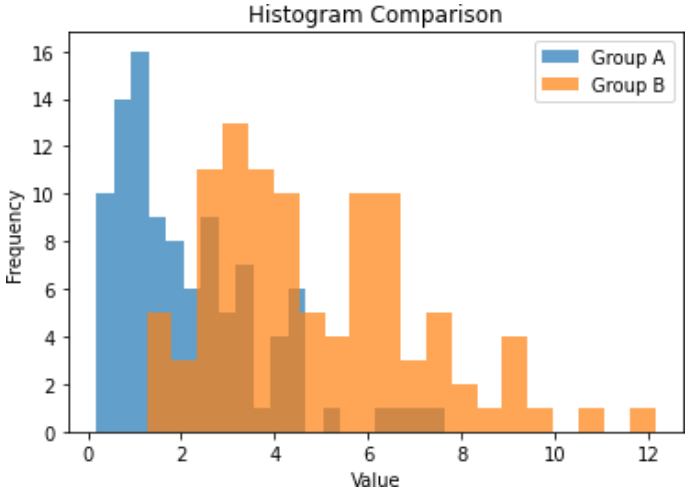

Figure 7: Frequency Vs Value Comparison

Histograms serve as a fundamental analytical tool for evaluating the distribution of confidence scores or error magnitudes produced by Temporal Convolutional Network (TCN) models in the classification of biomedical signals (Figure 7). Each histogram illustrates the frequency distribution of a particular metric, such as the confidence level of classifications across different classes or the magnitude of errors across various instances.

By examining these histograms, researchers can gain insights into the model's confidence in its predictions and detect any potential biases towards certain classes. For example, a unimodal histogram with a peak at high confidence scores suggests that the model is generally sure about its classifications. In contrast, a bimodal distribution indicates varying levels of confidence; where one peak may represent high confidence and another low confidence, highlighting inconsistencies in the model's predictive assurance.

Such variability in confidence can be indicative of underlying issues like dis- parities in data quality or class representativeness within the training dataset. If certain classes show a wider spread or a lower peak in confidence, it might suggest that the model lacks sufficient representative data to learn effectively from those classes. This detailed understanding enables researchers to identify specific areas where the training process or data collection might need adjustments, such as by augmenting the dataset with more diverse examples or by refining the model's architecture to handle data variability better. Thus, histograms not only shed light on the model's performance nuances but also guide strategic improvements to enhance overall classification reliability and robustness.

4. CONCLUSION

The research presented here, utilizing Temporal Convolutional Network (TCN) models for classifying biomedical signals, has demonstrated distinct advantages through various data visualization techniques. The comparative bar plot (Figure 3) effectively highlighted the model's varying accuracy across different classes, suggesting areas for targeted improvements. The line plot (Figure 4) revealed the model's learning dynamics over epochs, providing insights into convergence and potential overfitting issues. Scatter plots (Figure 5) elucidated the relationship between predicted and actual classifications, emphasizing patterns and misclassifications. Box plots (Figure 6) assessed the consistency and generalization capability of the TCN model across different data folds, highlighting stability and outliers in model performance. Finally, histograms (Figure 7) offered a deeper understanding of the model's confidence in its predictions and exposed possible biases towards certain classes. Collectively, these visualizations not only affirm the capability of TCN models in handling complex classification tasks but also pave the way for refining these models to enhance their accuracy and reliability in clinical applications. This comprehensive analysis underscores the importance of diverse evaluation metrics and visual aids in developing robust AI tools for healthcare diagnostics.

REFERENCES

[1] Zhiguang Wang and Tim Oates. Spatially encoding temporal correlations to classify temporal data using convolutional neural networks. *arXiv preprint arXiv:1511.06448*, 2015.

[2] Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. *arXiv preprint arXiv:1607.01759*, 2016.

[3] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Shun Chen. LSTM fully convolutional networks for time series classification. *arXiv preprint arXiv:1709.05206*, 2017

[4] Huai-Shuo Huang, Chien-Liang Liu, and Vincent S. Tseng. Multivariate time series early classification using multi-domain deep neural network. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 237–245. IEEE, 2018.

[5] Roberto Interdonato, Dino Ienco, Raffaele Gaetano, and Kenji Ose. Du- PLO: A dual view point deep learning architecture for time series classification. *arXiv preprint arXiv:1805.10913*, 2018.

[6] Charlotte Pelletier, Geoffrey I. Webb, and Francois Petitjean. Temporal convolutional neural network for the classification of satellite image time series. *arXiv preprint arXiv:1806.00717*, 2018.

[7] Chien-Liang Liu, Wen-Hoar Hsaio, and Yao-Chung Tu. Time series classification with multivariate convolutional neural network. *IEEE Transactions on Industrial Electronics*, 66(6):4788–4797, 2019.

[8] Yuan Yuan, Haopeng Li, and Qi Wang. Spatiotemporal modeling for video summarization using convolutional recurrent neural network. *IEEE Access*, 7:96996–97006, 2019.

[9] Chao-Lung Yang, Chen-Yi Yang, Zhi-Xuan Chen, and Nai-Wei Lo. Multivariate time series data transformation for convolutional neural network. In *2019 IEEE/SICE International Symposium on System Integration (SII)*, pages 303–308. IEEE, 2019.

[10] Qianli Ma, Shuai Tian, Jia Wei, Jiabing Wang, and Wing W. Y. Ng. Attention-based spatio-temporal dependence learning network. *Information Sciences*, 503:445–458, 2019.

[11] Charlotte Pelletier, Zehui Ji, Olivier Hagolle, Elizabeth Morse-McNabb, Kathryn Sheffield, Geoffrey I. Webb, and Francois Petitjean. Using sentinel- 2 image time series to map the state of Victoria, Australia. In *2019 10th International Workshop on the Analysis of Multitemporal Remote Sensing Images (MultiTemp)*, pages 1–4. IEEE, 2019.

[12] Yu-Jhen Chen, Chien-Liang Liu, Vincent S. Tseng, Yu-Feng Hu, and Shih- Ann Chen. Large-scale classification of 12-lead ECG with deep learning. In *2019 IEEE EMBS International Conference on Biomedical Health Informatics (BHI)*, pages 1–4. IEEE, 2019.

[13] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V. Chawla. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *AAAI*, volume 33, pages 1409–1416, 2019

[14] Fazle Karim, Somshubra Majumdar, and Houshang Darabi. Adversarial attacks on time series. *arXiv preprint arXiv:1903.01550*, 2019.

[15] Chao-Lung Yang, Zhi-Xuan Chen, and Chen-Yi Yang. Sensor classification using convolutional neural network by encoding multivariate time series as two-dimensional colored images. *Sensors*, 19(14):3163, 2019.

[16] Qianli Ma, Zhenjing Zheng, Wanqing Zhuang, Enhuan Chen, Jia Wei, and Jiabing Wang. Echo memory-augmented network for time series classification. *Neural Networks*, 126:180–189, 2020.

[17] Yuan Yuan and Lei Lin. Self-supervised pretraining of transformers for satellite image time series classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:987–997, 2021.

[18] Alessandro Michele Censi, Dino Ienco, Yawogan Jean Eudes Gbodjo, Ruggero Gaetano Pensa, Roberto Interdonato, and Raffaele Gaetano. Attentive spatial temporal graph CNN for land cover mapping from multi temporal remote sensing data. *IEEE Access*, 9:84912–84924, 2021.

[19] Qianli Ma, Enhuan Chen, Zhenxi Lin, Jiangyue Yan, Zhiwen Yu, and Wing W. Y. Ng. Convolutional multitimescale echo state network. IEEE Trans- actions on Cybernetics, 51(9):4535–4546, 2021.

[20] Dongjin Song. CAREER: Towards continual learning on evolving graphs: from memorization to generalization. *University of Connecticut*, 2024.

[21] Dongjin Song, Yuncong Chen, Cristian Lumezanu, Haifeng Chen, and Chuxu Zhang. Unsupervised anomaly detection, diagnosis, and correction in multivariate time series data. *arXiv preprint arXiv:1904.01700*, 2019