CNN-Trans: A Two-Branch CNN Transformer Model for Multivariate Time Series Classification

Sarra Hassine[1], Sourour Ammar[1,2][a] and Ilef Ben Slima[2,3][b]
[1]*Digital Research Center of Sfax, B.P. 275, Sakiet Ezzit, 3021 Sfax, Tunisia*
[2]*SM@RTS: Laboratory of Signals, systeMs, aRtificial Intelligence and neTworkS, Sfax, Tunisia*
[3]*ISMAIK, University of Kairouan, Kairouan, 3100 Tunisia*
benhassinesar1999@gmail.com, {sourour.ammar, ilef.benslima}@crns.rnrt.tn

Keywords: Multivariate Time Series Classification, Transformer, CNN, Deep Learning.

Abstract: The extensive presence of sensors in multiple domains has led to the generation of enormous amounts of multivariate time series data, presenting significant challenges for efficient classification. Although contemporary artificial intelligence methods show promising performance in addressing such data, they often struggle to capture both long-range dependencies and intricate local patterns within the sequences. This paper introduces CNN-Trans, an innovative deep learning model designed specifically for multivariate time series classification to address the mentioned challenge. CNN-Trans combines the strengths of transformers and convolutional neural networks (CNN). The proposed model uses a parallel strategy with both a transformer encoder and a CNN encoder working simultaneously on the time series data. The transformer captures global relationships through self-attention, while the CNN extracts localized spatial features tailored to each variable. We evaluate CNN-Trans on various benchmark datasets encompassing diverse sensor applications. The results show that our model is robust and highly effective for complex data. CNN-Trans outperforms others with 93.33% on NATOPS and 98.37% on PenDigits, excelling in high-dimensional datasets like Kitchen (95.74%) and HAR (87.41%). Additionally, CNN-Trans exhibits robustness and generalizability across different input features, showcasing its practical utility in real-world scenarios.

1 INTRODUCTION

Time series data, capturing the dynamic evolution of phenomena across time, pervades countless domains. From human activity recognition to medical diagnoses and financial forecasting, analyzing these sequences empowers critical decisions in real-world applications. Notably, multivariate time series (MTS), where data are collected across multiple variables simultaneously, pose distinct challenges due to their inherent complexity. Deep learning approaches like convolutional neural networks (CNN) have established themselves as valuable tools for tackling multivariate time series classification (MTSC) (Ismail Fawaz et al., 2019). Their capability to extract localized temporal and spatial correlations has led to significant advancements in various tasks. However, traditional CNNs encounter limitations in capturing long-range dependencies, which often hold crucial information for accurate classification. Additionally, their focus on local data interactions neglects the global context within the entire sequence. Seeking to address these limitations, recent research has explored the potential of recurrent neural networks (RNNs) like LSTMs (Karim et al., 2019). While capable of modeling long-range dependencies, their computational complexity and limited capacity hinder their widespread adoption (Vaswani, 2017). In contrast, attention models offer intriguing capabilities to capture long-range interactions efficiently. Their broader receptive fields allow for rich contextual information, enhancing the overall learning capacity of models. Not surprisingly, the success of attention models in natural language processing (Vaswani, 2017; Devlin et al., 2019) has spurred their adaptation to other domains like computer vision and, increasingly, time series analysis (Zerveas et al., 2021). At the heart of this revolution lies the transformer, a deep learning architecture that leverages powerful self-attention mechanisms (Vaswani, 2017). This mechanism excels at modeling relationships within the input time

[a] https://orcid.org/0000-0002-5851-9206
[b] https://orcid.org/0000-0003-0442-425X

Hassine, S., Ammar, S. and Ben Slima, I.
CNN-Trans: A Two-Branch CNN Transformer Model for Multivariate Time Series Classification.
DOI: 10.5220/0013169500003890

In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025) - Volume 2*, pages 418-428
ISBN: 978-989-758-737-5; ISSN: 2184-433X

series, uncovering intricate dependencies that influence classification outcomes. However, self-attention inherently neglects the crucial ordering information embedded within sequential data like time series. Addressing this poses a critical challenge, as the lack of explicit positional encoding can hinder the model's ability to fully understand the data's temporal dynamics. This issue is especially amplified in time series data, where context is often weaker compared to domains like text or image data. To overcome the limitations of existing methods and capitalize on the strengths of both transformers and CNNs, this paper introduces CNN-Trans, a novel deep learning architecture specifically designed for multivariate time series classification. CNN-Trans operates in parallel, employing a transformer encoder to capture long-range dependencies and a CNN encoder to extract localized features from each variable. This synergistic approach leads to a comprehensive representation that combines global and local contexts, enabling superior classification performance. Extensive evaluations on public and private datasets demonstrate that CNN-Trans consistently outperforms state-of-the-art approaches, highlighting its robustness and generalizability, particularly when handling complex, high-dimensional data.

Our study offers the following key contributions:

- Novel architecture: We propose CNN-Trans, the first architecture to combine transformers and CNNs for parallel processing of multivariate time series data.
- Enhanced long-range dependency modeling: Through the transformer encoder, we effectively capture long-range relationships within the time series, overcoming a limitation of traditional CNNs.
- Robust feature extraction: The CNN encoder extracts fine-grained features from each individual variable, providing valuable localized information for accurate classification.
- Improved performance: We demonstrate that CNN-Trans achieves competitive performance on various benchmark datasets across diverse domains, and its performance excels as dataset size increases, especially with high-dimensional data.

The remainder of this paper is structured as follows: Section 2 provides a comprehensive review of related research, highlighting previous approaches to MTSC and situating our work within the existing literature. Section 3 begins by introducing the basic concepts and properties of univariate and multivariate time series data and provides a mathematical formulation of the MTSC problem. Section 4 details our proposed method, including the architectural design of the model, the specific innovations introduced to handle MTS data effectively, and the integration of these components into a unified framework. Section 5 presents the experiments we conducted and discusses the findings. Finally, Section 6 concludes the paper while suggesting directions for future research.

2 RELATED WORK

2.1 CNN Based Models

The success of CNNs in various domains, including computer vision, speech recognition, and natural language processing, has led to their adoption in time series classification (TSC) tasks. Since the breakthrough of AlexNet in 2012 (Krizhevsky et al., 2012), CNN architectures have undergone significant improvements, such as the use of deeper networks, smaller and more efficient convolutional filters, and batch normalization to enhance training stability (Gu et al., 2018). These advancements have enabled CNNs to achieve state-of-the-art performance in numerous applications (Gu et al., 2018; Foumani and Nickabadi, 2019), and have paved the way for their application to TSC problems. For instance, the authors in (Wang et al., 2017) proposed a robust baseline model for TSC based on a Fully Convolutional Network (FCN). This approach operates end-to-end, involving training CNNs from scratch while requiring minimal preprocessing of raw data. Additionally, this approach utilizes Class Activation Maps to highlight significant regions in the data related to specific labels. Another significant contribution in the field of TSC is the InceptionTime (Ismail Fawaz et al., 2020). Inspired by Inception networks used in computer vision, InceptionTime is an ensemble of deep CNN models designed specifically for TSC. The authors showed that InceptionTime not only matches but often surpasses the accuracy of the HIVE-COTE algorithm, which was previously considered the state-of-the-art in TSC, while addressing its high training time complexity.

Although CNNs are designed to capture local patterns in data, they may struggle with long-range dependencies, especially when the dependencies span across many time steps. This can affect their performance in TSC where the temporal relationships are crucial.

2.2 Attention Based Models

Attention-based models have shown promise in addressing some of the limitations of CNNs in TSC. Attention mechanisms allow models to dynamically weigh the importance of different time steps in the input sequence, which enables the models to focus on critical moments that contribute significantly to the classification task. Models like the Attention LSTM Fully Convolutional Network (MALSTM-FCN) (Karim et al., 2019) and TapNet (Zhang et al., 2020) have been specifically designed to leverage attention mechanisms in the context of multivariate time series classification. MALSTM-FCN is an extension of the ALSTM-FCN (Karim et al., 2017) model, proposed to deal with multivariate time series data. It integrates attention within the LSTM framework, allowing it to capture long-range dependencies while simultaneously focusing on the most pertinent time steps (Karim et al., 2019). This results in improved accuracy and robustness in classifying complex time series data. Similarly, TapNet (Zhang et al., 2020) employs an attention mechanism to enhance feature extraction from multivariate inputs, ensuring that the model can adaptively learn which variables and time steps are most influential for the classification outcome. By focusing on the most relevant features, the attention mechanism of TapNet helps improve the model's ability to distinguish between different classes, especially in scenarios where labeled data is limited.

Attention-based models represent a significant advancement in the field of time series classification, as they not only improve classification performance but also enhance interpretability. As mentioned in (Hsu et al., 2019), the attention weights offer valuable insights into the model's decision-making process, highlighting which time steps and features are considered significant for specific classifications.

2.3 Transformers for MTS Classification

Transformers are a more recent development in attention-based models (Zerveas et al., 2021; Devlin et al., 2019; Liu et al., 2021; Zhang et al., 2023). Unlike previous attention mechanisms that were often paired with RNNs (such as ALSTM-FCN), Transformers rely entirely on self-attention mechanisms. The self-attention mechanism is a variant of the attention mechanism which allows the model to weigh the significance of different parts of the input relative to a specific position (Vaswani, 2017). This mechanism enables the model to process all elements in a sequence simultaneously, which is crucial for understanding context and semantics (Zhang et al., 2023). This architecture allows Transformers to capture long-range dependencies and global context more effectively than CNNs or traditional attention-augmented models. In the context of MTS classification, the authors in (Liu et al., 2021) proposed a transformer-based approach named "Gated Transformer Networks (GTN)". This approach combines the strengths of Transformer Networks with gating mechanism which merges two towers of Transformer networks and captures both channel-wise and step-wise correlations in multivariate time series data (Liu et al., 2021). Other transformer-based models have been proposed in the context of MTS classification such as (Zerveas et al., 2021) which proposed a transformer-based framework for unsupervised representation learning of multivariate time series and (Yang et al., 2024) which proposes a transformer-based dynamic architecture with a hierarchical pooling layer to decompose time series into subsequences representing different frequency components to facilitate time series classification. Although these transformer-based architectures are effective, they are complex and require large amounts of data for training. In addition, they involve an unsupervised learning phase to achieve optimal performance.

3 PROBLEM FORMULATION

In time series analysis, understanding the structure and characteristics of the data is crucial before tackling the problem of classification. Below, we first define univariate and multivariate time series, followed by a detailed explanation of the multivariate time series classification problem.

***A Univariate Time Series*:** is a sequence of observations collected over time from a single variable or feature. Mathematically, it can be represented as a one-dimensional vector $x = (x_1, x_2, \ldots, x_T) \in \mathbb{R}^T$, where T denotes the sequence length (i.e., the number of time steps). Each value x_t corresponds to the observation at time step $t \in 1, 2, \ldots, T$.

A multivariate time series (MTS), on the other hand, consists of multiple variables or features recorded simultaneously over time. Each sample in an MTS dataset is structured as a two-dimensional vector with a shape of (nf, T), where nf denotes the number of features (variables), and T indicates the sequence length (time step). A data sample can be represented as $X = (x_1, x_2, \ldots, x_{nf}) \in \mathbb{R}^{nf \times T}$, where each feature vector $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,T})$ represents the sequence

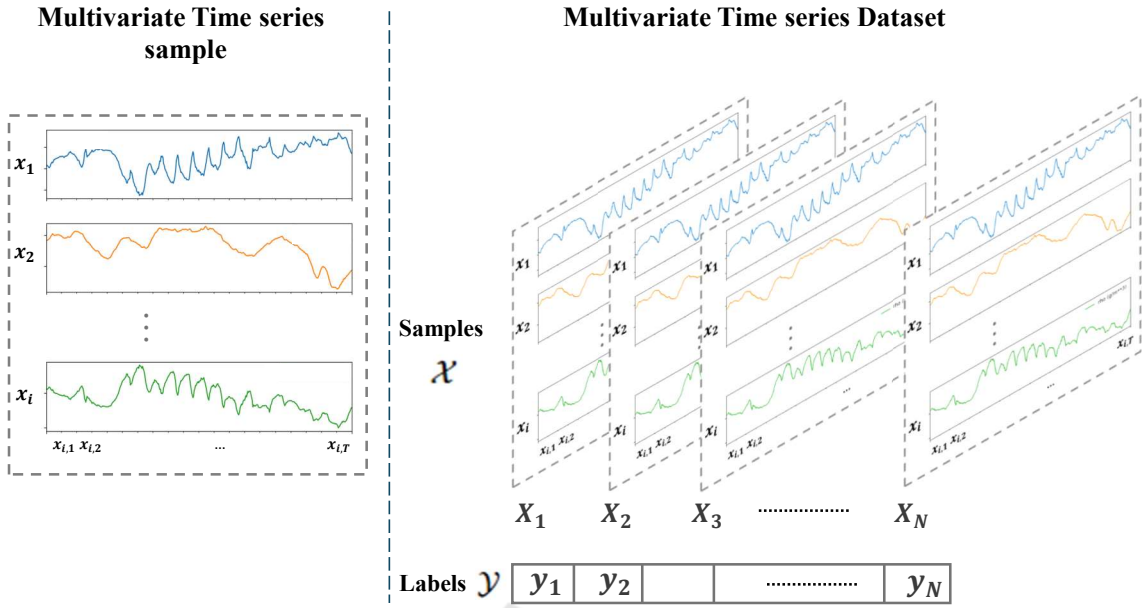


Figure 1: Representation of multivariate time series sample and multivariate time series Dataset.

of observations for the i^{th} feature over T time steps (see Figure 1 (left)).

In the context of multivariate time series data, the model's input consists of two-dimensional samples. Each sample, represented as $X = (x_1, x_2, \ldots, x_{nf}) \in \mathbb{R}^{nf \times T}$, is an ordered sequence with T time steps and nf features. Each sample X is associated with a class label $y \in \Omega$, where Ω is a predefined set of possible labels.

Given a collection of N multivariate time series samples (see Figure 1 (b)), represented as $\mathcal{X} = (X_1, X_2, \ldots, X_N) \in \mathbb{R}^{N \times nf \times T}$, along with their corresponding true labels $\mathcal{Y} = (y_1, y_2, \ldots, y_N) \in \Omega^N$, our goal is to classify each input sample X_i into one of the classes in Ω. The task of ***multivariate time series classification (MTSC)*** is thus to predict the label y for a given MTS data sample X. Figure 1 (right) illustrates a MTS dataset, consisting of N samples along with their labels. In this study, we employ our proposed CNN-Trans model to learn the mapping between the input data $\mathcal{X}$ and the target labels $\mathcal{Y}$.

4 CNN-TRANS FOR MULTIVARIATE TIME SERIES CLASSIFICATION

This paper presents a novel architectural design that utilizes the strengths of both convolutional neural networks and transformers. Figure 2 illustrates the proposed model architecture for MTS classification. The design consists of a two-branch model in a parallel configuration, with 1) a *CNN-based branch* dedicated to extracting local features and 2) a *Transformer-based branch* focused on capturing temporal dependencies. The outputs of the two branches are then concatenated and passed into a classification head.

The transformer encoder plays a vital role in the proposed model for classifying multivariate time series data, as it captures correlations between variables using an attention mechanism. This mechanism enables the model to selectively focus on relevant parts of the input sequence and assigns different weights to different variables based on their importance. By incorporating the transformer, the model can effectively interpret relationships between variables and capture intricate dependencies within the data.

In our proposed model, the CNN is utilized to automatically learn and capture important local features from the input data. This helps in identifying key patterns that are essential for distinguishing between different classes. Time series data often exhibit local patterns, such as peaks, troughs, or recurring shapes, which are important for classification. CNNs use convolutional filters to capture these local dependencies and patterns across time steps, allowing the model to focus on the most relevant parts of the data.

The combination of a CNN branch and a Transformer encoder branch in the proposed architecture enables simultaneous analysis of temporal features and correlation between variables, thereby effectively overcoming a key limitation in existing classification models.

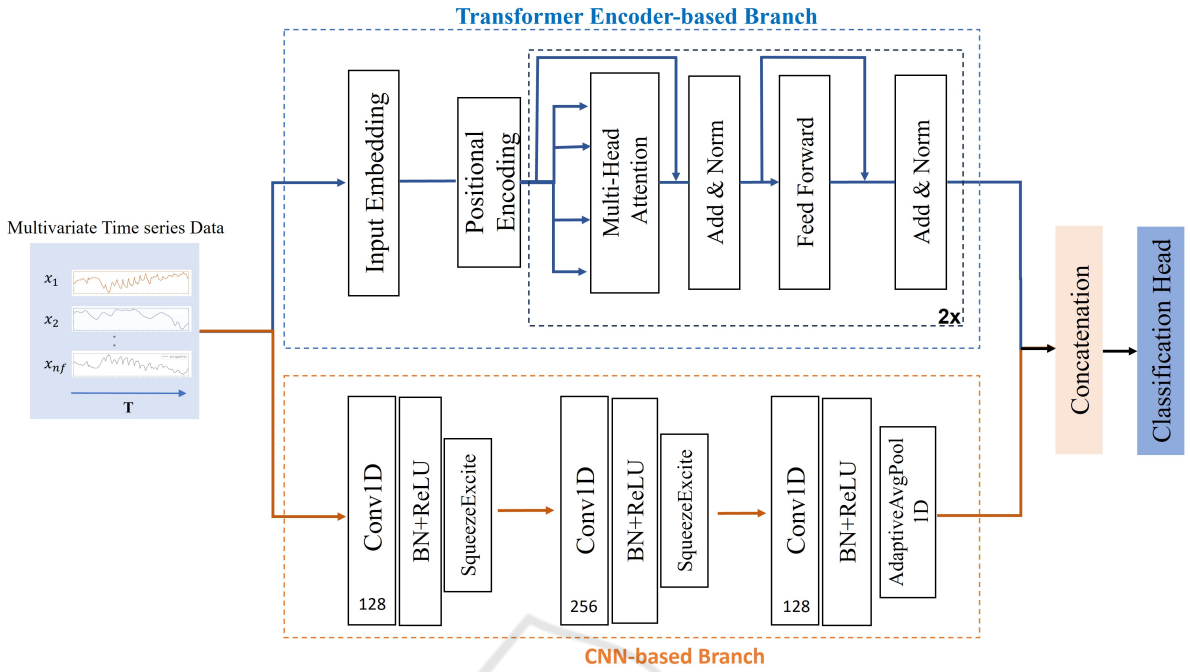


Figure 2: Overview of the CNN-Trans Architecture. CNN-Trans consists of a two-branch model in a parallel configuration: a CNN-based branch and a Transformer encoder-based branch.

4.1 Network Input

Our proposed model utilizes two distinct processing branches to handle multivariate time series data: a CNN-based branch, and a Transformer-based branch. The CNN branch directly processes the data in its original format, where it expects a specific number of time steps T with a certain number of distinct variables nf per step (we consider these variables as channels). On the other hand, the Transformer-based branch requires the data to be reshaped into a 3D tensor. This tensor organizes the data into dimensions $[B, T, nf]$, where B represents the batch size, T represents the number of time steps, and nf represents the number of features per step. This restructuring is aligned with the strengths of the Transformer model, enabling it to effectively capture intricate relationships between features across different time steps within each sequence in the batch.

4.2 CNN-Based Branch

The CNN component of the proposed architecture is inspired by the work (Karim et al., 2019). It consists of a sequence of three stacked temporal convolutional blocks. These blocks progressively extract characteristics from the input data using filters with varying sizes (128, 256, 128). Each block employs a temporal convolutional layer to capture temporal dependencies, followed by batch normalization for enhanced training stability and a ReLU activation function to introduce non-linearity. To improve feature representation, the first two convolutional blocks incorporate Squeeze-and-Excitation (SE) blocks (Hu et al., 2019). These SE blocks capture dependencies between channels within the feature maps and dynamically recalibrate the importance of each channel, which may highlight more relevant features. Finally, a global average pooling layer is applied after the final convolutional block to decrease the number of parameters before passing the data to the subsequent layers for classification.

4.3 Transformer-Based Branch

Since Multivariate time series classification focuses on identifying patterns within sequences of numerical data, we utilize a Transformer encoder, which excels at capturing long-range dependencies within sequences, making it well-suited for this task. Our Transformer-based branch begins with a linear input layer that transforms the raw feature space into a higher-dimensional representation. To account for sequence order, positional encoding is added to the input. The core of the encoder consists of a stack of two layers, each containing a multi-head attention mechanism with four heads and a subsequent feedforward neural network. The multi-head attention layer captures dependencies across different positions in the sequence, while the feedforward network enhances

the model's ability to discern intricate patterns within the data.

4.4 Classification Head

The classification head takes as input the concatenated output from the two branches. It consists of a Dense layer, of size equal to the number of classes, with softmax as the activation function to produce a probability distribution across the classes. We use the categorical cross-entropy as the loss function, and we train the network by minimizing this loss between the predicted class probabilities $\hat{y}$ and the true labels y, as expressed in the following:

$$\mathcal{L} = -\sum_{i=1}^{n} y_i \log(\hat{y}_i)$$

with n is the number of classes.

5 EXPERIMENTS

In this section, we describe our experiments designed to evaluate the performance of CNN-Trans in terms of its accuracy in multivariate time series classification. We first describe the datasets used in our experiments. We then summarize the implementation details. Finally, we present the obtained results and provide a comparison of our proposed method with state-of-the-art ones.

5.1 Datasets

We evaluate the proposed method using 7 datasets from the latest multivariate time series classification archive (Bagnall et al., 2018) (the first 7 rows in Table 1). This archive features real-world multivariate time series data from diverse applications, including Human Activity Recognition, Motion Classification, and ECG/EEG Signal Classification. The datasets vary in dimensionality from 2 dimensions in trajectory classification to 28 dimensions in EEG classification. The time series lengths range from 8 to 144, and the dataset sizes span from 80 to 10,992 samples.

Additionally, we tested our method on the Human Activity Recognition dataset (HAR dataset) from the UCI repository (Lichman, 2013). This dataset consists of recordings from 30 subjects performing various activities of daily living while carrying a waist-mounted smartphone equipped with inertial sensors. The subjects, aged between 19 and 48, performed six activities (Walking,Walking-Upstairs, Walking-Downstairs, Sitting, Standing, and Laying) while wearing a Samsung Galaxy S II on their waist.

Finally, since Transformer-based architectures require large amounts of data for effective training, we consider two private datasets (the last 2 rows in Table 1) to ensure sufficient data availability, thereby demonstrating the potential of the proposed CNN-Trans model for the MTSC task compared to other methods when there is sufficient training data. These two datasets are collected in two different contexts (one in a *Kitchen* and the other in a *Meeting room*), and for two different tasks: *Action recognition* and *Activity recognition*, respectively. They are collected from 8 ambient sensors, leading to 93 and 85 features, respectively for the Kitchen and the MeetingRoom datasets. For the Kitchen dataset, there are 3 classes, while there are 4 classes for the MeetingRoom dataset.

All datasets are split into training and testing sets as described in Table 1. For the purpose of ensuring comparability with previous studies, we retained the original training and testing splits provided in each MTS dataset. Each dataset is normalized to have zero mean and unit standard deviation, and the time series are padded with zeros to ensure that each time series has the same length as the longest series in the training set.

5.2 Implementation Details

All our experiments were implemented using Pytorch and run on a single Tesla T4 GPU (16GB). The CNN-Trans model is trained using the Adam optimizer with a learning rate of 0.00001 and a dropout rate of 0.1. We use the categorical cross-entropy loss function and implement a learning rate schedule that adjusts for plateaus, as recommended by (Ismail Fawaz et al., 2019). The training processes a batch size of 64. We assess the model's performance on both the training and validation sets at regular intervals, recording the best test results along with their corresponding hyperparameters. To ensure a fair comparison, we report the test accuracy of the model that achieves the lowest training loss, following the approach outlined by (Ismail Fawaz et al., 2019).

We evaluate the proposed method using accuracy (ACC) on the test set as the metric. ACC is calculated by the following formula:

$$ACC = \frac{Correct\ predictions}{All\ predictions}$$

We also provide the confusion matrices of the proposed CNN-Trans method for each dataset to offer a detailed view of its performance across different classes and highlight areas where misclassifications occur.

Table 1: Properties of all datasets considered in this work. HAR: Humain Activity Recognition. EEG: ECG/EEG Signal Classification. AR: Activity Recognition.

Datasets	Classes	Variables	length	Train	Test	Type
BasicMotions	4	6	100	40	40	HAR
RacketSports	4	6	30	151	152	HAR
NATOPS	6	24	51	180	180	HAR
Libras	15	2	45	180	180	HAR
FingerMovements	2	28	50	316	100	EEG
ArticularyWordRecognition	25	9	144	275	300	Motion
PenDigits	10	2	8	7494	3498	Motion
HAR dataset	6	9	128	7352	2947	AR
Kitchen	3	93	50	2988	470	Action
MeetingRoom	4	85	50	8217	921	AR

5.3 Results

We evaluate the performance of the proposed model CNN-Trans by comparing its classification accuracy with state-of-the-art methods: FCN (Wang et al., 2017), MLSTM-FCN (Karim et al., 2019), TapNet (Zhang et al., 2020), and ResNet (Wang et al., 2017).

We provide in Table 2 a performance comparison in terms of classification accuracy of the CNN-Trans model compared with state-of-the-art methods on 10 datasets. We highlight the highest accuracy score in red and the second-highest score in blue for each dataset.

The obtained results show that the proposed CNN-Trans method, based on a two-branch architecture combining a CNN and a Transformer, demonstrates competitive performance across various MTS datasets compared to other models. CNN-Trans consistently shows strong performance across most datasets. For instance, it achieves high accuracy on datasets such as RacketSports, Libra, FingerMovements, Kitchen, and MeetingRoom, outperforming the other methods. On NATOPS, PenDigits and ArticularyWordRecognition datasets, CNN-Trans is very close to the highest scores among the compared methods.
On the HAR dataset, although the MLSTM-FCN model achieved the highest accuracy of 96.71% compared to all the models, the CNN-Trans model remains competitive (achieving an accuracy of 87.41%), demonstrating its effectiveness across various time series classification tasks.

5.3.1 Performance Analysis Based on Dataset Size

Here we analyze the performance of the CNN-Trans model in relation to the size of the datasets, including the number of variables, the sequence length, and the amount of training data. This analysis offers additional insights into its strengths and potential limitations.

CNN-Trans exhibits strong scalability and effectiveness, particularly as dataset size increases from medium to large, making it well-suited for complex, high-dimensional time series data. For instance, on medium-sized datasets like NATOPS (180 training samples) and Kitchen (2988 training samples), CNN-Trans achieves high accuracy, outperforming other models such as ResNet, and demonstrating its ability to learn from a sufficient but not excessive amount of data. The confusion matrices illustrated in Figure 3(a) show that failure cases are related to the most confusing activities, such as classes 1 (*All clear*) and 2 (*Not clear*) in the NATOPS dataset.

On large datasets like PenDigits (7494 training samples) and Meeting Room (8217 training samples), CNN-Trans performs exceptionally well, with accuracies of 98.37% and 84.25%, respectively, showing its capacity to handle extensive data and extract meaningful patterns. Figure 4(b) shows that for the HAR dataset, despite strong performance, there is noticeable misclassification between adjacent classes (class 3:*sitting* and class 4:*standing*). This reflects difficulties in distinguishing highly complex or overlapping activity patterns.

However, on smaller datasets like BasicMotions (40 training samples) and FingerMovements (316 training samples), while CNN-Trans remains competitive, it occasionally falls short compared to simpler models like FCN, which might be more efficient with limited data. This suggests that while CNN-Trans is highly effective in more complex scenarios, its performance can vary depending on the dataset size, particularly when data is scarce (see Figure 5).

In summary, CNN-Trans is a flexible model that performs well as the size of the dataset grows, especially for handling intricate, high-dimensional data. Its competitiveness on smaller datasets remains strong, but it may not consistently outperform simpler models that are more appropriate for limited data.

Table 2: Performance comparison of the proposed model with state-of-the-art methods. For each dataset, the red result denotes the model with the best performance and the blue result indicates the model with the second best accuracy. The symbol '-' means that the accuracy of the corresponding model is not available for the dataset.

Datasets	CNN-Trans	FCN	MLSTM-FCN	TapNet	ResNet
BasicMotions	90.00	100	95.00	100	100
RacketSports	84.86	82.23	-	-	82.23
NATOPS	93.33	87.78	88.9	93.9	89.44
Libras	86.66	85.00	-	-	83.89
FingerMovements	59.00	53.00	-	-	54.00
ArticularyWordRecognition	98.33	98.00	97.30	98.70	98.00
PenDigits	98.37	98.57	97.80	98.00	97.71
HAR	87.41	84.45	96.71	-	87.11
Kitchen	95.74	66.17	94.25	81.7	53.19
MeetingRoom	84.25	63.95	70.14	82.23	68.08

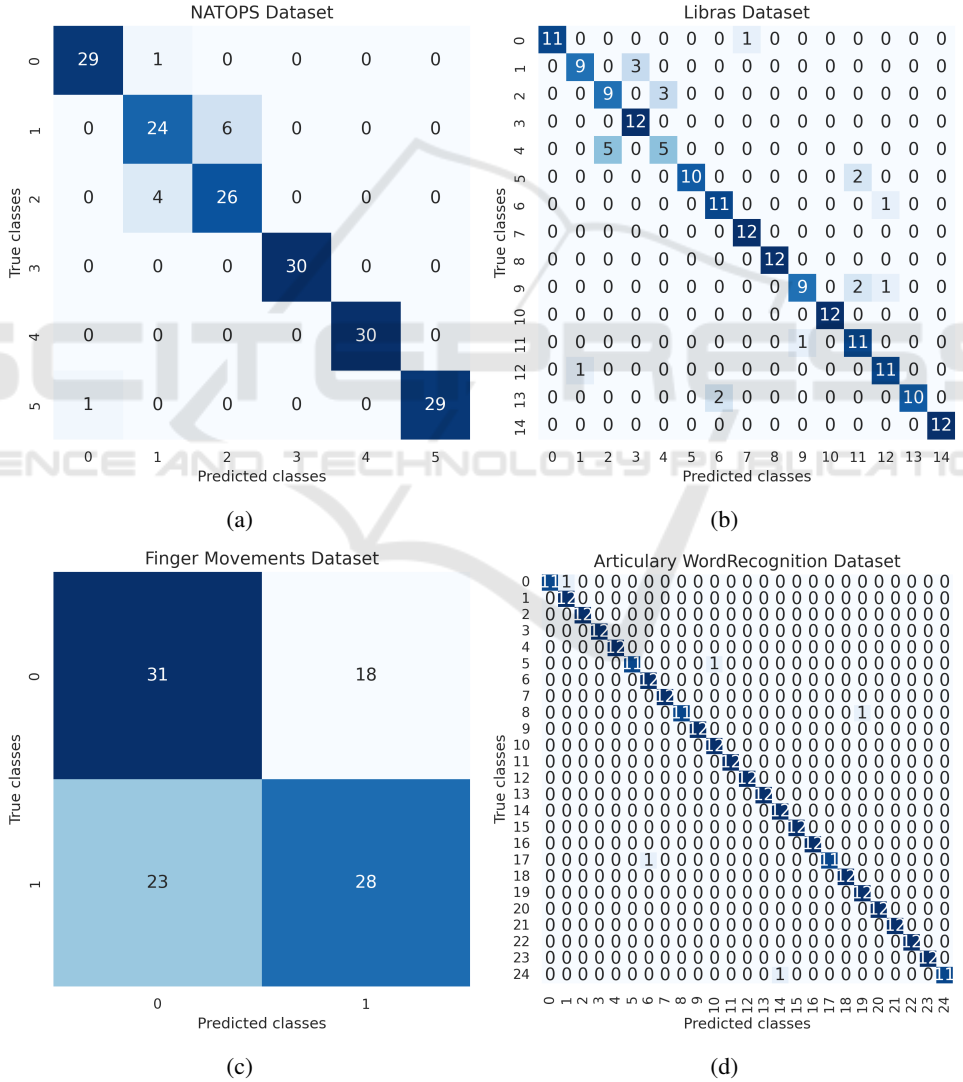


Figure 3: Confusion matrices of the CNN-Trans method on NATOPS, Libra, FingerMovements, and ArticularyWordRecognition datasets.

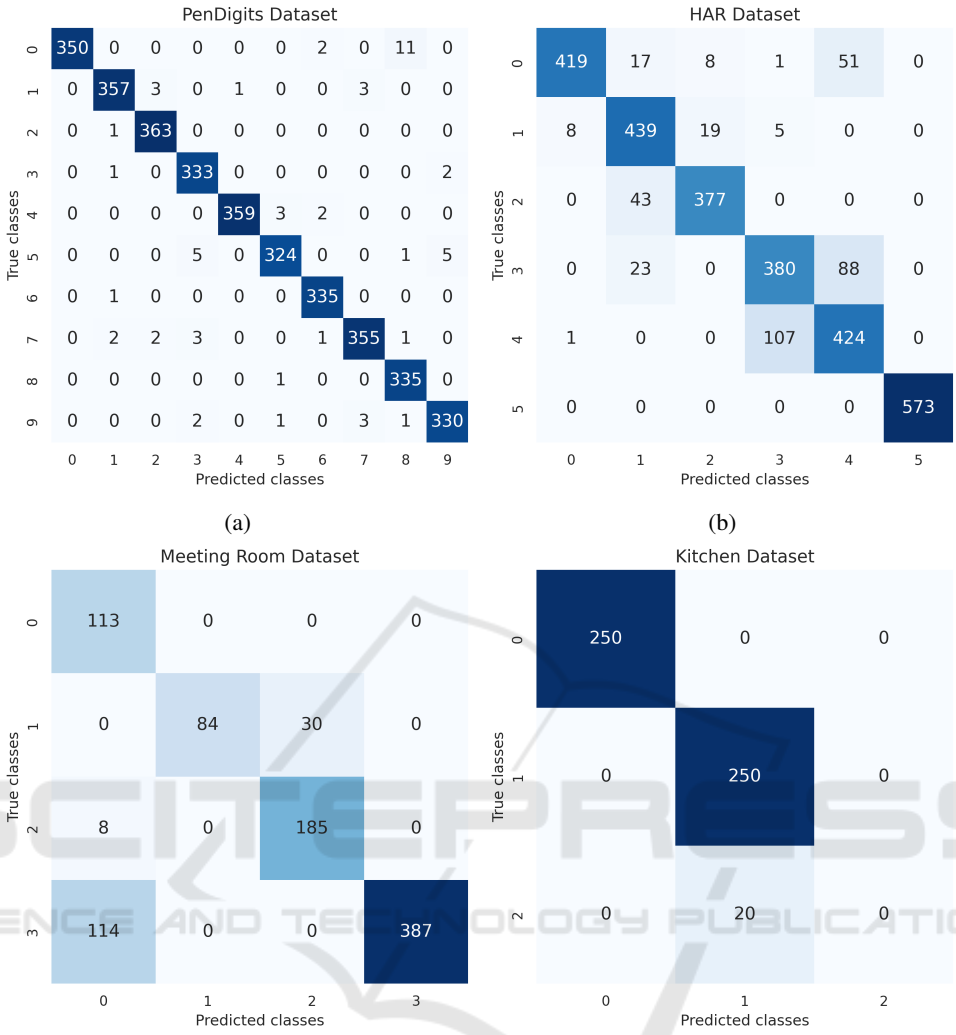


Figure 4: Confusion matrices of the CNN-Trans method on PenDigits, HAR, MeetingRoom and Kitchen datasets.

5.3.2 Performance Analysis Based on Data Type

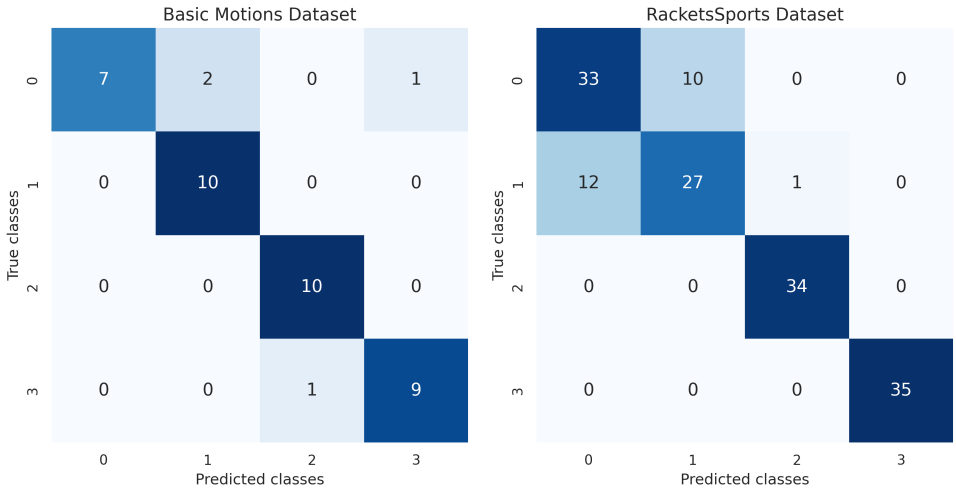
Here we analyze the behavior of the CNN-Trans model in relation to the type of dataset. By grouping datasets based on their type, namely HAR, EEG/ECG, motion-based tasks, and complex AR, we highlight the strengths and challenges of CNN-Trans model when dealing with diverse data characteristics.

Table 2 shows that CNN-Trans achieves consistently high accuracy in HAR tasks, particularly on NATOPS (93.33%) and BasicMotions (90%), indicating its effectiveness in capturing temporal and spatial features for activity recognition. For more challenging datasets like RacketSports, the performance is slightly lower (84.86%) but still competitive with the highest accuracy across all tested methods.

On the FingerMovements dataset, that represents EEG signals, the accuracy is relatively low across all models, with CNN-Trans achieving the highest (59%). This underscores the complexity of EEG data, which requires sophisticated feature extraction and may benefit from preprocessing or domain-specific adaptations. The confusion matrix, illustrated in Figure 3, highlights the confusion between the two classes, emphasizing the challenge of extracting meaningful features from the EEG signal.

On the other hand, on motion data type, all models perform well, with CNN-Trans achieving 98.33% and 98.37% on ArticularyWordRecognition and PenDigits datasets, respectively. This finding shows the strength of CNN-Trans in tasks with structured, continuous motion data.

On datasets that are collected from ambient sensors, Kitchen and MeetingRoom, CNN-Trans ex-



(a) (b)

Figure 5: Confusion matrices of the CNN-Trans method on BasicMotions and RacketSports datasets.

cels, achieving 95.74% (Kitchen) and 84.25% (MeetingRoom), significantly outperforming others. This highlights its robustness and adaptability, particularly in datasets with diverse and noisy features.

6 CONCLUSION AND FUTURE WORK

In this paper, we introduced CNN-Trans, a novel deep learning model designed specifically for multivariate time series classification. By integrating the strengths of convolutional neural networks and transformers, CNN-Trans effectively addresses the challenges posed by the extensive and complex nature of multivariate time series data. The model's parallel architecture allows for simultaneous processing of local features through CNNs and global relationships through transformers, leading to enhanced classification accuracy. Our extensive evaluations on both benchmark and private datasets demonstrate that CNN-Trans consistently outperforms state-of-the-art approaches, showcasing its robustness and generalizability across diverse sensor applications. CNN-Trans excels particularly when dealing with complex, noisy, and high-dimensional data, confirming its adaptability and potential for practical applications.

While CNN-Trans offers improvements in interpretability through attention weights, further research could delve into enhancing the explainability of the model's decisions. In future work, we aim to develop methods to visualize and interpret the contributions of different features and time steps, thereby enhancing trust in the model's predictions.

To further enhance CNN-Trans in future work, we aim to integrate a self-supervised learning phase to leverage vast unannotated datasets. This could involve pre-training with unlabeled data, extracting features through self-supervised methods. Such integration aims to improve model performance and adaptability by utilizing both labeled and unlabeled data more effectively.

REFERENCES

Bagnall, A. J., Dau, H. A., Lines, J., Flynn, M., Large, J., Bostrom, A., Southam, P., and Keogh, E. J. (2018). The UEA multivariate time series classification archive, 2018. *CoRR*, abs/1811.00075.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA*, volume 1, pages 4171–4186.

Foumani, S. N. M. and Nickabadi, A. (2019). A probabilistic topic model using deep visual word representation for simultaneous image classification and annotation. *Journal of Visual Communication and Image Representation*, 59:195–203.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., et al. (2018). Recent advances in convolutional neural networks. *Pattern recognition*, 77:354–377.

Hsu, E.-Y., Liu, C.-L., and Tseng, V. S. (2019). Multivariate time series early classification with interpretability using deep learning and attention mechanism. In

Pacific-Asia Conference on Knowledge Discovery and Data Mining, pages 541–553. Springer.

Hu, J., Shen, L., Albanie, S., Sun, G., and Wu, E. (2019). Squeeze-and-excitation networks.

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963.

Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D. F., Weber, J., Webb, G. I., Idoumghar, L., Muller, P.-A., and Petitjean, F. (2020). Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962.

Karim, F., Majumdar, S., Darabi, H., and Chen, S. (2017). Lstm fully convolutional networks for time series classification. *IEEE access*, 6:1662–1669.

Karim, F., Majumdar, S., Darabi, H., and Harford, S. (2019). Multivariate lstm-fcns for time series classification. *Neural networks*, 116:237–245.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Lichman, M. (2013). Uci machine learning repository. http://archive.ics.uci.edu/ml [Accessed: (August 21, 2024)].

Liu, M., Ren, S., Ma, S., Jiao, J., Chen, Y., Wang, Z., and Song, W. (2021). Gated transformer networks for multivariate time series classification. *arXiv preprint arXiv:2103.14438*.

Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Wang, Z., Yan, W., and Oates, T. (2017). Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pages 1578–1585. IEEE.

Yang, C., Wang, X., Yao, L., Long, G., and Xu, G. (2024). Dyformer: A dynamic transformer-based architecture for multivariate time series classification. *Information Sciences*, 656:119881.

Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., and Eickhoff, C. (2021). A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2114–2124.

Zhang, E. Y., Cheok, A. D., Pan, Z., Cai, J., and Yan, Y. (2023). From turing to transformers: A comprehensive review and tutorial on the evolution and applications of generative transformer models. *Sci*, 5(4):46.

Zhang, X., Gao, Y., Lin, J., and Lu, C.-T. (2020). Tapnet: Multivariate time series classification with attentional prototypical network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 6845–6852.