


Article

# A Deep Multi-Task Learning Approach for Bioelectrical Signal Analysis

Jishu K. Medhi <sup>1</sup> , Pusheng Ren <sup>1</sup>, Mengsha Hu <sup>2</sup> and Xuhui Chen <sup>1,\*</sup>

<sup>1</sup> College of Aeronautics and Engineering, Kent State University, Kent, OH 44240, USA; jmedhi@kent.edu (J.K.M.); pren1@kent.edu (P.R.)

<sup>2</sup> Department of Computer Science, Kent State University, Kent, OH 44240, USA; mhu8@kent.edu

\* Correspondence: xchen58@kent.edu

**Abstract:** Deep learning is a promising technique for bioelectrical signal analysis, as it can automatically discover hidden features from raw data without substantial domain knowledge. However, training a deep neural network requires a vast amount of labeled samples. Additionally, a well-trained model may be sensitive to the study object, and its performance may deteriorate sharply when transferred to other study objects. We propose a deep multi-task learning approach for bioelectrical signal analysis to address these issues. Explicitly, we define two distinct scenarios, the *consistent source-target scenario* and the *inconsistent source-target scenario* based on the motivation and purpose of the tasks. For each scenario, we present methods to decompose the original task and dataset into multiple subtasks and sub-datasets. Correspondingly, we design the generic deep parameter-sharing neural networks to solve the multi-task learning problem and illustrate the details of implementation with one-dimension convolutional neural networks (1D CNN), vanilla recurrent neural networks (RNN), recurrent neural networks with long short-term memory units (LSTM), and recurrent neural networks with gated recurrent units (GRU). In these two scenarios, we conducted extensive experiments on four electrocardiogram (ECG) databases. The results demonstrate the benefits of our approach, showing that our proposed method can improve the accuracy of ECG data analysis (up to 5.2%) in the MIT-BIH arrhythmia database.

**Keywords:** multi-task learning; bioinformatics; deep learning

**MSC:** 68T07



**Citation:** Medhi, J.K.; Ren, P.; Hu, M.; Chen, X. A Deep Multi-Task Learning Approach for Bioelectrical Signal Analysis. *Mathematics* **2023**, *11*, 4566. <https://doi.org/10.3390/math11224566>

Academic Editors: Gourav Modanwal and Rakesh Shiradkar

Received: 28 September 2023

Revised: 1 November 2023

Accepted: 3 November 2023

Published: 7 November 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Recent decades have witnessed a surge of machine learning techniques in biomedical data analysis. Bioelectrical signals, some of the most significant biomedical data, refer to the change in electric current produced by the sum of an electrical potential difference across a specialized tissue, organ, or cell system like the nervous system. These changes are recorded by electrodes in contact with the skin surface or directly with physiological tissue that conducts electricity. The best-known methods to record bioelectrical signals are the electrocardiogram (ECG), which records heart activity, the electroencephalogram (EEG), recording brain electrical activity, and the electromyogram (EMG), recording muscle electrical activity [1–3]. Academia and industry have proposed many machine learning schemes to analyze bioelectrical signals, such as decision tree [4], support vector machine (SVM) [5], hidden Markov models [6], and artificial neural networks [7]. These systems are capable of identifying the commonalities and features behind the bioelectrical signals through properly managed training. For example, recurrent neural networks (RNN) are well-suited for analyzing ECG and EEG signals with features having high temporal complexity, whereas decision trees and SVMs are excellent for handling EMG signals that exhibit higher spatial complexity. With increased training intensity, they can handle more complex analysis tasks, which in turn support physicians in diagnosing diseases and

can significantly save the huge human resources consumption in the traditional way of data analysis.

One of the complexities of bioelectrical signals is their nonlinear nature, which makes it difficult to process them accurately using traditional statistical tools. Currently, only fully trained and experienced physicians can perform such tasks and, even then, the clinician's judgment is still accompanied by a certain amount of error. Fortunately, deep learning, an advanced method among the aforementioned systems, has strong capabilities in nonlinear functional learning and has found extensive applications in multi-dimensional signal processing problems [8]. Therefore, combining deep learning with bioelectrical signal analysis is a promising exploration direction. Some previous works [9,10] have shown that the method can achieve state-of-art performance. Lipton et al. [9] trained a long short-term memory (LSTM) recurrent neural network to diagnose heart diseases automatically and obtained 86.6% classification accuracy. Rajpurkar et al. [10] classified heartbeat arrhythmias with a proposed 34-layer convolutional neural network, which exceeds the average cardiologist's performance in both sensitivity and precision. The above attempts ensure the accuracy of signal processing while greatly saving the consumption of manpower, which undoubtedly strengthens our confidence to continue exploring in this direction.

One of the barriers to the development of deep learning techniques is the limited amount of labeled data available for training, which causes the data sparsity problem, and further becomes a major bottleneck for applying deep learning in bioelectrical signals analysis. Normally, it requires millions of labeled data points to train a deep neural network [11–13]. In addition, data labeling usually needs to be performed manually, leading to the extremely time-consuming process of signal labeling.

Moreover, a well-trained deep model may be sensitive to the study objects. When transferred to other study objects, the performance of the predictive model may deteriorate sharply. The main reason behind this phenomenon is that bioelectrical signals can differ substantially among patients [7,14]. For instance, seizure morphology on EEG signals can vary among different patients due to the diverse neuroanatomical and pathophysiological causes of epileptic disease; the same arrhythmia may have divergent morphology on ECG signals for different patients. Two different arrhythmias can also produce nearly identical effects on standard ECG signals as well because of the various electrodes' positions [15]. Therefore, the scope and complexity of the data used to train the deep network needs to be further enhanced, and the structure of the system itself needs to be optimized accordingly in order to improve the transferability, generalizability, and robustness of the system.

Multi-task learning offers an attractive solution, via reusing knowledge learned by other similar tasks, to address the issues of data sparsity and object sensitivity in the target task. Multi-task learning trains several similar tasks jointly to capture the commonalities among them. The individual task utilizes the shared knowledge to boost its own performance. It has achieved great success in many fields, including speech recognition [16,17], computer vision [18,19], text mining [20,21], and drug discovery [22,23].

A preliminary version of this work has been reported [24], which focuses on utilizing deep multi-task learning to increase the performance of ECG arrhythmias detection and classification. We defined two scenarios to apply multi-task learning, the *consistent source-target scenario* and the *inconsistent source-target scenario*. We initially proposed one deep multi-task learning structure for both scenarios and person-wise study objects in source and target domains. The experiments are conducted on one dataset (MIT-BIH Arrhythmia Database), which is not thorough enough to evaluate the proposed scheme's efficacy and transferability.

This paper extends the deep multi-learning scheme to the broader bioelectrical signal analysis. Specifically, based on the initially proposed two distinct scenarios, the study objects are no longer considered person-wise but are expanded to a group of patients' data from four databases. Moreover, we proposed two distinct generic parameter-sharing neural networks for these two scenarios to improve the transferability. The experiments on four databases demonstrate that our proposed system can increase the performance

of bioelectrical signal analysis. Thereby, the predictive model learned from the multi-task learning approach has a better transferable performance.

We summarize the major contributions as follows:

- Based on the purpose and motivation behind data analysis tasks, we define two different scenarios for utilizing multi-task learning in analyzing bioelectrical signals, the *consistent source-target scenario* and the *inconsistent source-target scenario*, which is proved to enhance the transferability of the proposed schemes.
- For each scenario, we propose a method to decompose the analysis task into several subtasks and convert the original dataset to adapt to the multi-task learning. We also design the generic parameter-sharing neural networks for each scenario and illustrate the details of implementing different basic neural layers, like convolutional layers and recurrent layers.
- We conduct extensive experiments on four electrocardiograms databases. The experiment's results demonstrate that the proposed systems can improve the analysis performance and enable the predictive models to be transferable.

The rest of this paper is organized as follows: Section 2 preliminarily introduces deep learning and multi-task learning. In Section 3, we propose two different scenarios—the *consistent source-target scenario* and the *inconsistent source-target scenario*—to apply multi-task learning in the bioelectrical signal analysis. Sections 4 and 5 illustrate the details of applying deep multi-task learning in those two scenarios, respectively. Section 6 describes the experiments on arrhythmia classification using ECG signals, and its results are presented in Section 7. We discuss the findings of our work in Section 8 and, finally, summarize the paper in Section 9.

## 2. Preliminaries

### 2.1. Deep Learning

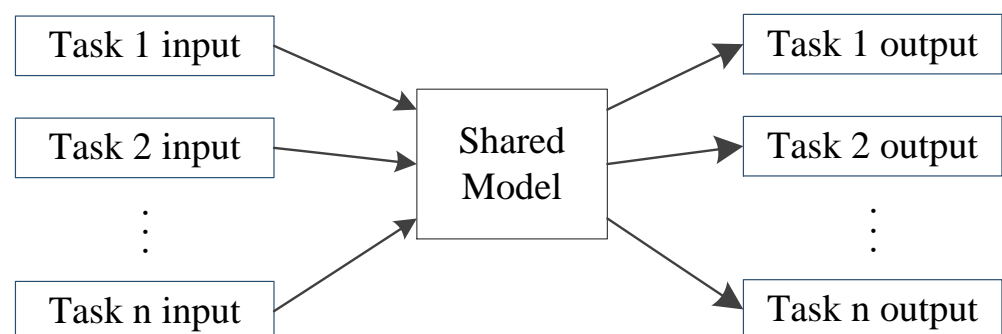
Deep learning serves as an incredibly powerful tool for various machine learning applications [25,26]. It can easily generate accurate predictive models through the utilization of deep neural networks (DNNs). Typically, DNNs consist of multiple layers of nonlinear processing nodes, allowing them to learn feature representations through each layer. The earliest framework of deep neural networks is based on multi-layered artificial neural networks (ANNs) inspired by biological neurons in the brain [27]. ANNs are still widely used for approximating mathematical functions and regression analysis making them very useful for tasks like predicting the behavior of financial markets, modeling physical systems, and predicting real estate prices [28,29]. Nonetheless, the broad impact of deep learning becomes apparent in 2006 [30–32]. Since then, deep learning has been successfully applied to a wide range of fields, including computer vision, natural language processing, and bioinformatics [7,24,33].

In this paper, we focus on bioelectrical signal analysis, in which the available data constitute a time sequence-series. Traditionally, DNNs lacked the ability to model the dynamic temporal behavior of such data. Thus, many researchers proposed plenty of DNN variants to exhibit the temporal features of the data, like 1-Dimension convolutional neural networks (1D CNNs) [34], vanilla recurrent neural networks (RNNs) [35], convolutional long-short term memory units (ConvLSTMs) [36], recurrent neural networks with long-short memory units (LSTMs) [37], and recurrent neural networks with gated recurrent units (GRUs) [38]. In 2019, Hasan et al. [39] trained a 1D convolutional neural network to diagnose cardiovascular disease automatically. Hannun et al. [40] proposed a bidirectional LSTM convolutional neural network to detect and classify the heartbeat arrhythmias and achieved an accuracy of 96.59%, a sensitivity of 99.93% and a specificity of 97.03%. More recently, Hu et al. [41] proposed a novel CNN-transformer based deep learning model to classify heartbeats from continuous ECG signals for arrhythmia detection and achieved an overall accuracy of 99.12%.

## 2.2. Multi-Task Learning

Multi-task learning (MTL) involves addressing multiple related tasks concurrently, aiming to leverage shared characteristics and distinctions among these tasks to improve the performance of each of the tasks. In contrast, single task learning (STL) primarily focuses on optimizing metrics for a specific target task, which is the conventional approach in machine learning. Nevertheless, the MTL systems are given a collection of related tasks that all need to be solved. Theoretical and experimental studies have shown that the overall prediction quality can be improved by processing the tasks jointly and thus sharing information among them. Moreover, if the tasks are similar but the data are in different domains, the system can acknowledge domain invariant features. Thus, the MTL systems are more transferable than the STL systems. Due to these benefits, MTL systems have been successfully adopted in various applications ranging from speech recognition to computer vision [15]. In 2020, Shahin et al. [42] built a deep neural network (DNN) system consisting of three sub-networks to deal with multi-tasking simultaneously, which was tested with the MIT-BIH Arrhythmia Dataset. However, this article only focuses on the classification task of ECG signals and does not clearly demonstrate the transferability of training results between tasks. Although a novel multi-task learning architecture is developed in this article, they do not specify tasks. We believe that the potential of multi-task learning can be exploited to a greater extent by subdividing the target tasks and applicable scenarios in combination with various neural network architectures. In 2020, Mormont et al. [43] investigated multi-task learning as a way of pre-training models for classification tasks in solving the problem of the small-medium size of the pathological image database. They effectively improve the accuracy and efficiency of recognition and classification tasks of small and medium-sized databases through multi-task learning. However, due to limited data resources and computing power, this paper did not fully complete the experiments as expected, which may not prove their claims. On this basis, we developed an architecture of deep parameter-sharing neural network applied to scenarios of consistent source-target and inconsistent source-target in order to achieve a better multi-task learning performance. We also experimented with different neural network architectures such as 1D-CNNs, RNNs, LSTMs, and GRUs in the above scenarios to further explore the potential of deep neural networks in bioelectrical signals analysis.

In order to process the tasks jointly, neural-based models provide an efficient way to combine the representation of multiple tasks that makes it attractive, ranging from computer vision [44,45] to natural language processing [46,47]. We show the architecture of multi-task learning in Figure 1. First, each task is input into a shared model that consists of neural layers. Such shared layers are used to extract common feature representations. Then, the remaining networks are split into multiple branches to capture individual feature representations for each particular task [48,49].



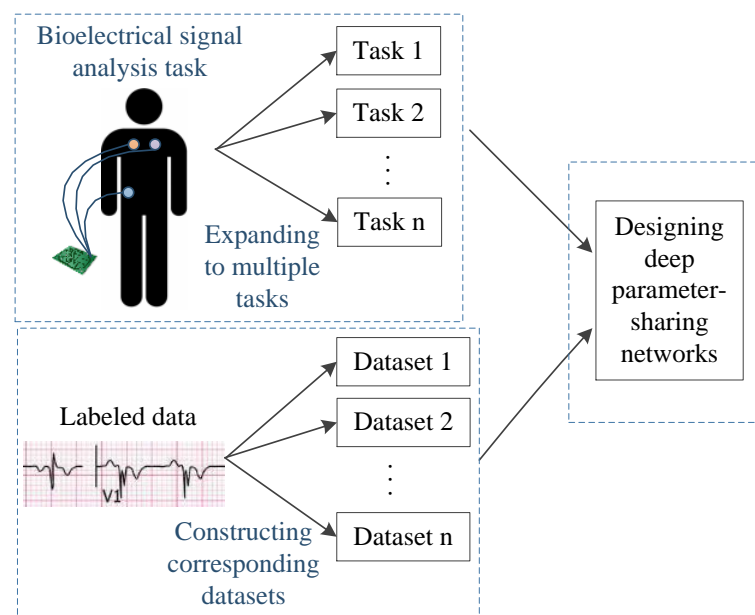
**Figure 1.** Model architecture of neural-based multi-task learning.

## 3. Deep Multi-Task Learning for Bioelectrical Signal Analysis

This section presents the proposed deep multi-task learning approach for bioelectrical signal analysis. As shown in Figure 2, our scheme consists of three main subprocesses.

Specifically, for a bioelectrical signal analysis task, we first expand it into several similar and related subtasks, which is a task space partitioning process. Then, based on these subtasks, we convert the original dataset to sub-datasets to adapt to the multiple subtasks. Finally, we design a generic parameter-sharing neural network to train these subtasks simultaneously in a deep multi-task setting.

The primary challenge of designing a deep multi-task learning scheme for bioelectrical signal analysis is how to transform one single analysis task into several similar and related subtasks. We assume that a patient/a group of patients, upon whose data we obtain the predictive model, is the source domain or source study object. Correspondingly, we call another patient/group of patients, to whom the predictive model will be applied to analyze the data, as target domain or target study object. In this paper, we propose two distinct methods to conduct task space partitioning for two scenarios, respectively, where the scenarios are distinguished based on the consistency of study subject. Specifically, if the source and target study objects are the same patient/group of patients, we name this scenario the *consistent source-target scenario*, where the goal of utilizing multi-task learning is to increase the analysis precision. Otherwise, if the source and target study objects are different patients/groups of patients, we name this scenario the *inconsistent source-target scenario*, where the objective of utilizing multi-task learning is to learn a transferable predictive model for bioelectrical signal analysis. Since the sub-datasets building process and parameter-sharing neural network designing process heavily depend on the task partitioning, we introduce them for the source-target consistent and inconsistent scenarios in the following, respectively.



**Figure 2.** The architecture of the deep MTL bioelectrical analysis system.

## 4. Deep Multi-Task Bioelectrical Analysis in Consistent Source-Target Scenario

### 4.1. Multiple Tasks and Related Datasets

As mentioned in Section 3, in the *consistent source-target scenario*, we focus on improving the performance of bioelectrical signal analysis. To address this issue, we decompose a complex analysis task into several simple subtasks. For instance, traditionally, researchers take ECG arrhythmia recognition as a complex multi-class classification, where they endeavour to categorize a heartbeat correctly among multiple types. However, in the multi-task learning setting, we decompose this complicated multi-class classification problem into several simple binary classification subproblems. For each subtask, it only concentrates on determining whether the beat belongs to the class it is interested in.

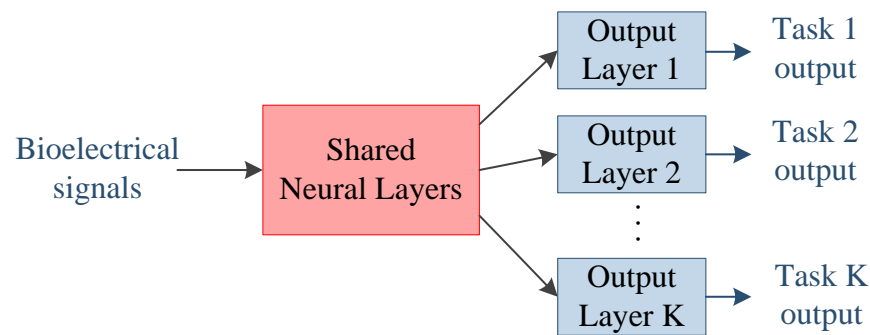


Intuitively, each subtask only focuses on solving a simple subproblem in the entire complicated task space. Meanwhile, as a system, subtasks collaboratively assist each other to improve their performance in their own problems. Therefore, the multi-task learning system has a better performance than a single task system.

According to the task decomposition in the *consistent source-target scenario*, we rebuild the original datasets. For original dataset  $D$  in a multi-class classification setting, data sample  $d_i$  is  $\{x_{i_s:i_e}, y_i\}$ , where  $x_{i_s:i_e} = \{x_{i_s} \oplus x_{i_s+1} \oplus \dots \oplus x_{i_e}\}$  is the bioelectrical signal sequence and  $y_i \in \{1, 2, \dots, K\}$  is the class label, and  $K$  is the number of categories. We rebuild the multi-class datasets to  $k$  binary classification dataset  $D^*$ .  $d_i^*$  is  $\{x_{i_s:i_e}, y_i^1, y_i^2, \dots, y_i^K\}$ , where  $y_i^m \in \{0, 1\}$  is the label for subtask  $m$ .

#### 4.2. Deep Parameter-Sharing Neural Networks for Consistent Source-Target Scenario

We propose deep parameter-sharing neural networks for the *consistent source-target scenario*. Noticeably, the deep parameter-sharing neural networks are a generic network structure, and any neural network components can be applied to it. As shown in Figure 3, every subtask shares several common layers and has its private output layers.



**Figure 3.** Model architecture of deep parameter-sharing neural networks in the *consistent source-target scenario*.

In particular, the output of the shared neural layers is the hidden feature representation:

$$h_s = \text{SharedLayers}(x_{i:i+j}). \tag{1}$$

The output of subtask  $i$  is:

$$\tilde{y}_i = \text{sigmoid}(\mathbf{w}^T h_s + \mathbf{b}). \tag{2}$$

Since we decomposed the multi-class classification task into several binary classification subtasks, we formulate the loss function for task  $i$  as:

$$L_i = - \sum_{p=1}^n \sum_{q=1}^2 y_{ip}^q \log(\tilde{y}_{ip}^q), \tag{3}$$

where  $n$  is the number of sequence data.

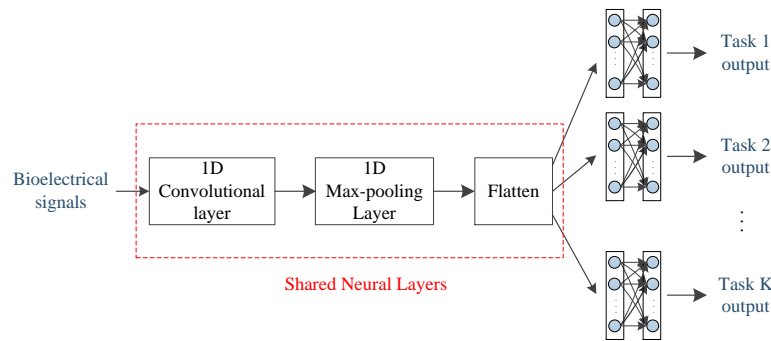
Thus, combining the loss functions of subtasks, the overall loss function is derived for the whole network:

$$L = \sum_{i=1}^K L_i. \tag{4}$$

Usually, a bioelectrical signal is the time series data of electrical measurement. In deep learning, researchers always utilize convolutional neural networks and recurrent neural networks to capture the features of time sequence data. Accordingly, we design parameter-sharing convolutional neural networks and parameter-sharing recurrent neural networks for the *consistent source-target scenario*, respectively.

#### 4.2.1. Parameter-Sharing Convolutional Neural Networks

We utilize 1D convolutional neural networks to capture the temporal features from bioelectrical signals. Specifically, as shown in Figure 4, the signal goes through a shared 1D convolutional neural layer and a shared 1D max-pooling layer. Then, a shared flattened layer converts the extracted feature tensor to a feature vector  $\mathbf{h}$ . For task  $i$ , it concatenates its private fully connected layer to the flattened layer to extract its interested features. After this, the task  $i$  utilizes another fully connected layer with sigmoid activation  $\sigma(\cdot)$  to determine whether the signal sequence belongs to its type.

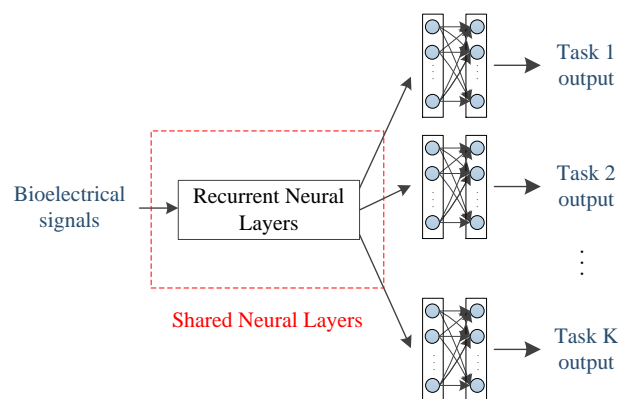


**Figure 4.** Model architecture of deep parameter-sharing convolutional neural networks in the *consistent source-target scenario*.

#### 4.2.2. Parameter-Sharing Recurrent Neural Networks

We also implement RNN to extract features. As shown in Figure 5, for task  $i$ , the parameter-sharing neural networks extract hidden representation vector  $\mathbf{h}$  from the bioelectrical signal via a shared recurrent neural layer. In the parameter-sharing convolutional neural networks, task  $i$  appends two fully-connected layers to the output of the recurrent neural layer to conduct binary classification.

Moreover, since RNN with LSTM units and RNN with GRU units have proved their performance in time sequence feature extraction for many applications, we can use the recurrent neural layer with LSTM units and the recurrent neural layer with GRU units to replace the shared vanilla recurrent neural layer.



**Figure 5.** Model architecture of deep parameter-sharing recurrent neural networks in the *consistent source-target scenario*.

### 5. Deep Multi-Task Bioelectrical Analysis in Inconsistent Source-Target Scenario

#### 5.1. Multiple Tasks and Related Datasets

When the source and target study objects are different patients/groups of patients, the target study object may experience a prediction performance decline if he directly uses a model that is trained by source study objects using single-task learning. Thus, the motivation of utilizing multi-task learning in the *inconsistent source-target scenario* is to

learn a transferable predictive model for bioelectrical signal analysis. Similarly, we divide the multi-class classification task into several multi-class subtasks. Specifically, each subtask conducts the multi-class analysis as the original task, but the bioelectrical data in its dataset are from one patient/one group of patients. By solving these subtasks simultaneously, the multi-task learning model can capture the invariant features among all the subtasks, which makes the model more transferable.

According to the task decomposition in the *inconsistent source-target scenario*, we rebuild the original datasets. For a multi-task setting, the original dataset  $D$  is formulated as:

$$\begin{pmatrix} x_{1_s:1_e} & y_1 \\ x_{2_s:2_e} & y_2 \\ \vdots & \vdots \\ x_{n_s:n_e} & y_n \end{pmatrix} \tag{5}$$

where  $n$  is the number of data samples,  $x_{i_s:i_e} = x_{i_s:i_s+1} \oplus x_{i_s:i_s+2} \oplus \dots \oplus x_{i_s:i_e}$  is the time sequence data, and  $y_i \in \{1, 2, \dots, K\}$ ,  $K$  is the number of categories.

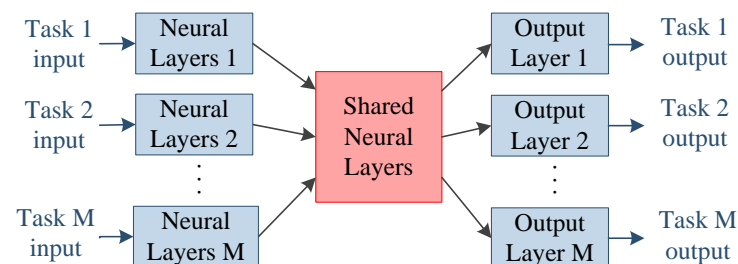
Therefore, we decompose  $D$  into several sub-datasets  $D_1^*, D_2^*, \dots, D_M^*$ , where  $M$  is the number of subtasks. For task  $i$ , its sub-dataset  $D_i$  is formulated as:

$$\begin{pmatrix} x_{1_s:1_e}^i & y_1^i \\ x_{2_s:2_e}^i & y_2^i \\ \vdots & \vdots \\ x_{n_i:n_{ie}}^i & y_{n_i}^i \end{pmatrix} \tag{6}$$

where  $n_i$  is the number of data samples in task  $i$ .

### 5.2. Deep Parameter-Sharing Neural Networks in Inconsistent Source-Target Scenario

We also propose deep parameter-sharing neural networks for the *inconsistent source-target scenario*. Still, the deep parameter-sharing neural network is a generic network structure, and any neural network components can be applied to it. As shown in Figure 6, each subtask can apply any existing neural network (blue block) to extract its individual features. Subsequently, the outputs from the shared neural layers (red block) from all the tasks are linked to the individual output layers of each task, generating their respective outputs.



**Figure 6.** Model architecture of deep parameter-sharing neural network in the *inconsistent source-target scenario*.

In particular, the output of task  $i$ 's neural layers is:

$$\mathbf{h}_i = \text{Input}(x_i).$$

The output of shared neural network layers is:

$$\mathbf{h}_s = \text{SharedLayers}(\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_M).$$



We consider the analysis as a multi-class classification problem. Then the output of task  $i$  is:

$$\tilde{y} = \text{softmax}(\mathbf{w}\mathbf{h}_s + b).$$

Correspondingly, the loss function for task  $i$  as:

$$L_i = - \sum_{p=1}^{n_i} \sum_{q=1}^K y_{ip}^q \log(\tilde{y}_{ip}^q).$$

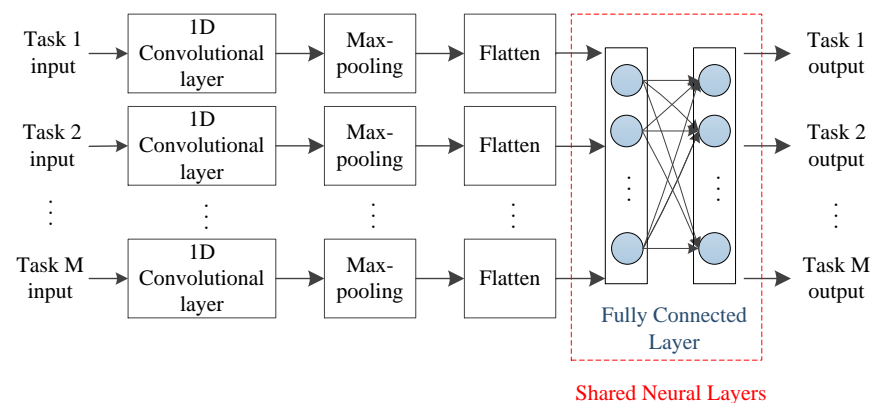
The overall loss function of the network can then be obtained as:

$$L = \sum_{i=1}^M L_i.$$

In the following, we illustrate the details of applying convolutional neural networks and recurrent neural networks to build multi-task learning networks in a consistent scenario.

### 5.2.1. Parameter-Sharing Convolution Neural Networks

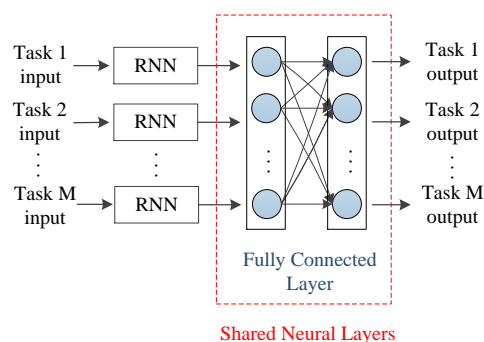
As depicted in Figure 7, the training dataset for each task  $i$  undergoes a series of transformations, beginning with a 1D convolutional neural layer and followed by a 1D max-pooling layer. Subsequently, a flattened layer is applied to convert the resulting feature tensor into a feature vector, represented by  $\mathbf{h}_i$ . By combining this feature vector with those from other tasks, we create a shared feature vector for all tasks, denoted by  $\mathbf{h} = \mathbf{h}_1 \oplus \mathbf{h}_2 \oplus \dots \oplus \mathbf{h}_M$ . Following this, the network incorporates several fully-connected layers, with parameters shared among all relevant tasks to connect to each task's dedicated output layer.



**Figure 7.** Model architecture of deep parameter-sharing convolutional neural network in the *inconsistent source-target scenario*.

### 5.2.2. Parameter-Sharing Recurrent Neural Networks

As illustrated in Figure 8, the training dataset for each task  $i$  is provided as input to a recurrent neural layer which extracts the hidden state vector represented by  $\mathbf{h}_i$ . Combining this hidden state vector with the corresponding vectors from other tasks gives us a shared feature vector for all tasks, denoted by  $\mathbf{h} = \mathbf{h}_1 \oplus \mathbf{h}_2 \oplus \dots \oplus \mathbf{h}_M$ . Subsequently, the network incorporates several fully-connected layers, with parameters shared among all relevant tasks to connect to each task's dedicated output layer. In the consistent scenario, we can apply RNN with LSTM units and RNN with GRU units to improve the performance of vanilla RNN.



**Figure 8.** Model architecture of deep parameter-sharing recurrent neural network in the *inconsistent source-target scenario*.

## 6. Experiments on ECG Signal Analysis

We evaluate the proposed multi-task learning schemes on cardiac arrhythmia classification using ECG signals from four public databases from PhysioBank [50]. To validate the efficacy of our scheme, we conduct the experiments in two scenarios as described in Section 3, the *consistent* and the *inconsistent source-target scenarios*. In each scenario, we consider different records (patients) or different databases as different domains and evaluate the performance of our proposed schemes.

In this paper, we follow the previous study in the literature and classify arrhythmias into the following four classes, depending on the life-threatening situations (from the highest to the lowest), we have [51]:

1. *Ventricular (V)*: Such arrhythmias start in the heart's lower chambers, which can be very dangerous and usually require medical care right away. There are two types of such arrhythmias: premature ventricular contraction (PVC) and ventricular escape (VE).
2. *Fusion (F)*: Continuation of fusion will lead to stroke and heart failure. There is only one type of arrhythmias: fusion of ventricular and normal (fVN).
3. *Supraventricular (S)*: It refers to the arrhythmias that need to be noticed, but not necessarily needs to be sent to the hospital immediately. There are four types: atrial premature (AP) that almost 60% of people have experienced, aberrated atrial premature (aAP), nodal (NP), and supraventricular premature (SP).
4. *Normal heartbeat (N)*: It includes normal heart beats that was wrongly detected by the previous arrhythmia detection algorithm (NOR), and some normal arrhythmias: left or right bundle branch block (LBBB/RBBB), atrial escape (AE) and nodal (junctional) escape (NE).

### 6.1. Datasets Description

#### 6.1.1. The MIT-BIH Arrhythmia Database (mitdb)

This database contains 23 ECG recordings (numbered from 100 to 124 with some numbers missing) obtained from inpatients and 25 records (numbered from 200 to 234 inclusive, again with some numbers missing) including a variety of rare but clinically important phenomena that would not be well-represented by a small random sample of Holter recordings. Each of the 48 records is slightly over 30 min long, and each heartbeat is labeled as either normal or different types of arrhythmia by clinical experts. In each recording, there are two leads' readings, and we choose the lead MLII readings, which is consistent with related works. The sampling frequency is 360 Hz.

#### 6.1.2. The MIT-BIH Long Term Database (ltdb)

This database contains 7 long-term ECG recordings with a duration from 14 to 24 h each, and six of them includes two-channel ECG signals sampled at 128 Hz per channel with 12-bit resolution, and one three-channel ECG sampled at 128 Hz per channel with 10-bit resolution. A total of 668,486 heartbeats in the whole database are manually labeled

by independent experts. There are two leads' readings in each record, and we choose the first lead signals in the experiments.

#### 6.1.3. The MIT-BIH Supraventricular Arrhythmia Database (svdb)

This database includes 78 half-hour ECG recordings with a sampling frequency of 128 Hz. We exclude file 841 because of essential annotation errors. The remaining 77 records contain 162,000 normal beats, 12,000 supraventricular premature beats, and 9900 ventricular ectopic beats. The annotations of the recordings were first automatically labeled by the Marquette Electronics 8000 Holter scanner and, later, were reviewed and corrected by a medical student.

#### 6.1.4. St.-Petersburg Institute of Cardiological Technics 12-Lead Arrhythmia Database (incartdb)

The incartdb contains 75 files extracted from 32 Holter records (32 patients), 12 channels per file. Each record is 30 min long and is sampled at 257 Hz. A total of 175,000 heartbeats in the whole database are annotated by an automatic algorithm and then corrected manually, and most of which are ventricular ectopic beats. From the 12 standard leads, lead II was selected for consistency with other databases.

### 6.2. Data Preprocessing

In the experiments, we select most representative recordings from these databases that have diverse types of arrhythmias and relatively balanced samples. For example, records 104, 109, 111, 118, 124, 214, 231, 232 in mitdb and records 845, 848, 850, 855, 888, 890 in svdb were not considered, as they only contained exclusively paced, left/right bundle branch block, abnormal beats or severe noise within the first ten beats.

Regarding the raw signal preprocessing, we performed minimal operations for these datasets. Since the sampling frequency is different in these four databases, we downsampled the rate from 360 Hz to 128 Hz for the mitdb database and from 256 Hz to 128 Hz for the incartdb database. Then, we segmented the signals into beats fragments with the length of 88-bit with R peak in the middle. Each beat segment was normalized to eliminate the baseline drift, and some artificial beats were deleted. There was no manual feature extraction process in our experiments because of the characteristic of deep learning networks. In the experiments, we also balanced the training and testing datasets by downsampling the normal beats for the reason that there are many fewer arrhythmia beats compared to normal beats. The final ratio of normal beats to arrhythmia beats was 3:1.

## 7. Results

### 7.1. Experiments on Consistent Source-Target Scenario

As described in Section 3, the domain can be person-wise or database-wise. Regarding person-wise, we select the MIT-BHI Long Term Database (ltdb) [50] because of its sufficient recordings of each person for deep neural networks' training purpose while the other three databases only contain 30-min data, which are not enough for training a deep neural network. Regarding the database-wise domain, we perform the simulation four times on each database independently to validate the efficacy of the proposed MTL framework. We give the MLT deep neural networks training and testing details separately and compare the results with the ones under STL.

#### 7.1.1. Person-Wise MTL

We define the tasks and dataset in this scenario. For arrhythmia classification, we define the different arrhythmia types as different tasks and each recording (person) is a domain. Since there are four arrhythmia classes in the classification criteria, we have four subtasks. Then, we choose the training and testing data for the deep neural networks. We select a recording from the ltdb database, 14,046, and use 4/5 of the data for training and

1/5 as testing. We repeat the same process using another recording, 14,134, and record these two experiment results in Table 1.

**Table 1.** Person-wise testing accuracy (in %) in the *consistent source-target scenario*.

Records	Deep NNs	STL	MTL
14,046	1D CNN	94.1	97.9 <sub>(+3.8)</sub>
	RNN	93.0	96.3 <sub>(+3.3)</sub>
	LSTM	96.2	98.1 <sub>(+1.9)</sub>
	GRU	96.9	98.2 <sub>(+1.3)</sub>
	avg.	95.1	97.6 <sub>(+2.5)</sub>
14,134	1D CNN	88.0	90.4 <sub>(+2.4)</sub>
	RNN	87.4	89.1 <sub>(+1.7)</sub>
	LSTM	90.0	92.2 <sub>(+2.2)</sub>
	GRU	89.7	92.1 <sub>(+2.4)</sub>
	avg.	88.8	91.0 <sub>(+2.2)</sub>

### 7.1.2. Database-Wise MTL

Similar to the previous experiment setting, we define the different arrhythmia types as different tasks, but we treat each database as a domain. For the mitdb database, we combine recordings 106, 114, 116, 119, 201, 205, 210, 215, 219, 223 and train the STL and MTL networks using 5-fold cross-validation on these data, respectively. Similarly, we prepare the other three databases, ltldb (using recordings 14,046 and 14,134), svldb (803, 804, 805, 820, 824, 847, 852, 859, 887, 893), and incartdb (105, I08, I13, I19, I27, I30, I38, I42, I69, I72), and conduct same experiment on each database. The testing results are shown in Table 2.

**Table 2.** Database-wise testing accuracy (in %) in the *consistent source-target scenario*.

Databases	Deep NNs	STL	MTL
mitdb	1D CNN	84.6	86.9 <sub>(+2.3)</sub>
	RNN	83.0	87.8 <sub>(+4.8)</sub>
	LSTM	86.2	90.6 <sub>(+4.4)</sub>
	GRU	87.5	91.1 <sub>(+3.6)</sub>
	avg.	85.3	89.1 <sub>(+3.8)</sub>
ltldb	1D CNN	90.5	93.8 <sub>(+3.3)</sub>
	RNN	89.0	93.0 <sub>(+4.0)</sub>
	LSTM	92.6	94.5 <sub>(+1.9)</sub>
	GRU	93.2	95.9 <sub>(+2.7)</sub>
	avg.	91.3	94.3 <sub>(+3.0)</sub>
svldb	1D CNN	84.0	87.7 <sub>(+3.7)</sub>
	RNN	83.0	87.3 <sub>(+4.3)</sub>
	LSTM	87.9	90.7 <sub>(+2.8)</sub>
	GRU	86.7	89.9 <sub>(+3.2)</sub>
	avg.	85.4	88.9 <sub>(+3.5)</sub>
incartdb	1D CNN	93.4	95.8 <sub>(+2.4)</sub>
	RNN	92.9	94.0 <sub>(+2.1)</sub>
	LSTM	94.5	96.7 <sub>(+2.20)</sub>
	GRU	95.0	97.6 <sub>(+2.6)</sub>
	avg.	94.0	96.3 <sub>(+2.3)</sub>

## 7.2. Experiments on Inconsistent Source-Target Scenario

Similar to the *consistent source-target scenario*, the *inconsistent source-target scenario* also divides the tasks into person-wise and database-wise. In terms of person-wise, we treat different records as different tasks. Thus, we train a deep neural network on three records (three different tasks/the source tasks) and transfer the shared layers to a fourth record (the target task). We conduct this experiment using the MIT-BIH Arrhythmia Database (mitdb) [50] because it has a variety type of arrhythmias and has a bigger difference in ECG morphology among records. In terms of database-wise, we treat different databases as different tasks. Thereby, we want to transfer the knowledge of three databases (the source tasks) to the fourth database (the target task). We give training details in this section and evaluate the results with STL.

### 7.2.1. Person-Wise MTL

We define the tasks and the dataset as described in Section 3. Notice that different records are collected from different patients, and we treat different patients as different tasks. Thus, the multi-task learning framework tries to learn the domain invariant features over these tasks. To test the transferability of our learned shared extractor, we design an experiment, in which we take turns choosing three tasks to train a deep neural network with multi-task learning, and then the learned shared layer is transferred to a second network that is used for the remaining one task. The parameters of the transferred layer are kept frozen, and the rest of parameters of the network are randomly initialized. For the tasks, we select 4 recordings from mitdb, 106, 116, 210, and 223, since they have a relatively larger number of arrhythmic beats. We further train the network using 5-fold cross-validation on the remaining record and test the second network. For comparison, the STL trains the deep neural networks on all the data from these three recordings plus 4/5 data from the fourth recording, and then test the trained networks on the 1/5 of the fourth recording. The simulation repeats 4 times for each record as the target task while other three are source tasks. The testing results are shown in Table 3.

**Table 3.** Person-wise testing accuracy (in %) in the *inconsistent source-target scenario*.

Source Tasks	Target Task	Deep NNs	STL	MTL
106, 116, 210	223	1D CNN	96.2	97.3 <sub>(+1.1)</sub>
		RNN	95.1	96.6 <sub>(+1.5)</sub>
		LSTM	97.2	98.2 <sub>(+1.0)</sub>
		GRU	97.3	98.9 <sub>(+1.6)</sub>
		avg.	96.5	97.8 <sub>(+1.3)</sub>
116, 210, 223	106	1D CNN	75.9	77.9 <sub>(+2.0)</sub>
		RNN	75.3	78.3 <sub>(+3.0)</sub>
		LSTM	78.3	81.2 <sub>(+2.9)</sub>
		GRU	77.9	80.5 <sub>(+2.6)</sub>
		avg.	76.9	79.5 <sub>(+2.6)</sub>
106, 210, 223	116	1D CNN	91.6	93.2 <sub>(+1.6)</sub>
		RNN	90.5	93.0 <sub>(+2.5)</sub>
		LSTM	94.6	96.4 <sub>(+1.8)</sub>
		GRU	94.0	95.6 <sub>(+1.6)</sub>
		avg.	92.7	94.6 <sub>(+1.9)</sub>
106, 116, 223	210	1D CNN	82.9	84.2 <sub>(+1.3)</sub>
		RNN	81.4	84.0 <sub>(+2.6)</sub>
		LSTM	84.2	86.0 <sub>(+1.8)</sub>
		GRU	82.2	86.1 <sub>(+3.9)</sub>
		avg.	82.7	85.1 <sub>(+2.4)</sub>

### 7.2.2. Database-Wise MTL

We treat the analysis task on different databases as different tasks. The experiment is similar to the one in Section 7.2.1, except that we change the recordings to databases. We use the four databases, mitdb (with recordings 106, 114, 116, 119, 201, 205, 210, 215, 219, 223), ltdb (14,046 and 14,134), svdb (803, 804, 805, 820, 824, 847, 852, 859, 887, 893), and incartdb (I05, I08, I13, I19, I27, I30, I38, I42, I69, I72) for the four tasks. Table 4 shows the database-wise testing accuracy for the *inconsistent source-target scenario*.

**Table 4.** Database-wise testing accuracy (in %) in the *inconsistent source-target scenario*.

Source Tasks	Target Task	Deep NNs	STL	MTL
ltdb, svdb, incartdb	mitdb	1D CNN	85.0	87.1 <sub>(+2.1)</sub>
		RNN	83.2	88.0 <sub>(+4.8)</sub>
		LSTM	86.7	91.9 <sub>(+5.2)</sub>
		GRU	88.0	92.3 <sub>(+4.3)</sub>
		avg.	85.7	89.8 <sub>(+4.1)</sub>
mitdb, svdb, incartdb	ltdb	1D CNN	89.3	91.9 <sub>(+2.6)</sub>
		RNN	86.4	90.5 <sub>(+4.1)</sub>
		LSTM	90.5	92.6 <sub>(+2.1)</sub>
		GRU	91.0	93.8 <sub>(+2.8)</sub>
		avg.	89.3	92.2 <sub>(+2.9)</sub>
mitdb, ltdb, incartdb	svdb	1D CNN	85.0	88.2 <sub>(+3.2)</sub>
		RNN	84.0	87.9 <sub>(+3.9)</sub>
		LSTM	88.6	91.2 <sub>(+2.6)</sub>
		GRU	86.1	90.7 <sub>(+4.6)</sub>
		avg.	85.9	89.5 <sub>(+3.6)</sub>
mitdb, ltdb, svdb	incartdb	1D CNN	91.3	94.6 <sub>(+3.3)</sub>
		RNN	90.3	92.9 <sub>(+2.6)</sub>
		LSTM	94.0	95.1 <sub>(+1.1)</sub>
		GRU	94.5	96.7 <sub>(+2.2)</sub>
		avg.	92.5	94.8 <sub>(+2.3)</sub>

## 8. Discussion

### 8.1. Consistent Source-Target Scenario

The testing accuracy is the ratio of the number of correctly classified beats to the number of total beats in the testing phase. The STL schemes conduct the traditional machine learning process where each deep neural network has one multi-class output. From Table 1, we can see that average accuracy of person-wise MTL neural networks increase 2.5% and 2.2% compared with STL networks. Since the MTL separates all the arrhythmia classes in the training process, it is easier for neural networks to address the difference between classes and thus has better classification performance compared with training all classes together. Additionally, we compare it with the results from [52], where the neural networks are trained on different proportions of the training data in each recording. In their results, the highest testing accuracy is 97% for record 14,046 while our MTL is 98.2%, and 87% for record 14,134 while ours is 92.2%. Therefore, the proposed MTL parameter-sharing framework can improve the performance of predictive model.

When compared to the results of STL networks, the average testing accuracy of database-wise MTL increases by 3.8%, 3.0%, 3.5%, and 2.3% for the four selected datasets, respectively, as shown in Table 2. The results from these experiments suggest that the beat classification accuracy can be improved by deep MTL parameter-sharing systems when training and testing on the same database.



### 8.2. Inconsistent Source-Target Scenario

From Table 3, we can observe that the person-wise MTL demonstrates an improved average testing accuracy of 1.3%, 2.6%, 1.9%, and 2.4% in comparison to STL networks for the four selected records respectively. The results indicate that we can save the existing knowledge into shared layers, which is quite useful for a new task. We also compare the recording 223 with the result in reference [7], where the authors use an artificial neural network (ANN) for arrhythmia classification and obtain 97.32% testing accuracy for recording 223. Our MTL system can obtain 98.9% with the GRU network. Therefore, the proposed MTL framework can improve the classification accuracy compared with STL and traditional ANN-based when transferring knowledge between recordings.

In case of database-wise MTL, the average accuracy of MTL networks outperforms the STL by 4.1%, 2.9%, 3.6% and 2.3% for the four target databases of mitdb, ltdb, svdb and incartdb respectively. While the mitdb target database shows the highest improvement in average accuracy, the MTL network with LSTM architecture achieves the most significant improvement in testing accuracy (5.2%) for any task as shown in Table 4. We also compare the results with Table 2, where the training and testing data are from the same database. The average accuracy increases in mitdb and svdb by 0.8% and 0.6%, respectively, and decreases in ltdb and incartdb by 2.1% and 1.5%, respectively. The reason is that there may be some conflict samples in these databases that can damage the training performance. As we discussed, the same arrhythmia may have different ECG morphologies for different patients, and we notice that same beats even have different labels in the different databases. Therefore, training machine learning models on the combined databases require attention on the possible conflicting samples to avoid accuracy decreasing.

During our experiments, MTL has undeniably shown significant promise in improving model performance by sharing knowledge between related tasks. While the transferability of training from one task to another offers many benefits, it also presents challenges. The choice of which tasks to combine and the appropriate degree of parameter sharing require careful consideration. Overly aggressive sharing can hinder individual task performance, while too little sharing may limit transferability. Future research could focus on developing methods for automatically assessing task relatedness and complexity to adjust the degree of knowledge transfer accordingly.

## 9. Conclusions

In this paper, we have investigated the accuracy improvement problem of analyzing bioelectrical data by employing methods based on deep learning. To achieve this goal, a deep MTL scheme has been proposed to reuse the knowledge from source domains to target domain. In particular, we initially reframe the bioelectrical signal analysis problem as a multi-task learning problem by segmenting the data analysis into a list of tasks and then create corresponding datasets for those tasks. Then, we train the parameter-sharing neural network for these tasks and apply the shared layers to the target domain. Any generic deep learning network can be utilized in the framework, and we implement four networks as examples—1D CNN, vanilla RNN, LSTM, and GRU. To evaluate the proposed approach, we conduct extensive experiments on arrhythmia classification using four public ECG databases. For each scenario, we test the system in person-wise and database-wise domains separately. The experiment's results show that the proposed framework can improve the classification accuracy in all situations, which means our system successfully transfers the knowledge from the source domain to the target domain, which also means the parameter-sharing layers can capture the common features and get rid of personal features from records or databases. The accuracy can be improved by up to 5.2% in the MIT-BIH Arrhythmia Database using LSTM networks.

**Author Contributions:** Conceptualization, X.C.; Methodology, X.C.; Software, J.K.M. and P.R.; Validation, J.K.M. and P.R.; Formal analysis, J.K.M.; Data curation, P.R.; Writing—original draft, X.C.; Writing—review & editing, J.K.M., P.R. and M.H.; Visualization, P.R. and M.H.; Supervision,

X.C.; Project administration, X.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The data presented in this study are openly available in PhysioNet as follows: mitdb database available at <https://doi.org/10.13026/C2F305>; ltdb database available at <https://doi.org/10.13026/C2KS3F>; svdb database available at <https://doi.org/10.13026/C2V30W>; incartdb database available at <https://doi.org/10.13026/C2V88N>.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Theis, F.J.; Meyer-Bäse, A. *Biomedical Signal Analysis: Contemporary Methods and Applications*; MIT Press: Cambridge, MA, USA, 2010.
2. Choi, B.J.; Kim, J.H.; Yang, W.J.; Han, D.J.; Park, J.; Park, D.W. Parylene-based flexible microelectrode arrays for the electrical recording of muscles and the effect of electrode size. *Appl. Sci.* **2020**, *10*, 7364. [[CrossRef](#)]
3. Aoyama, T.; Kohno, Y. Temporal and quantitative variability in muscle electrical activity decreases as dexterous hand motor skills are learned. *PLoS ONE* **2020**, *15*, e0236254. [[CrossRef](#)] [[PubMed](#)]
4. Behadada, O.; Chikh, M.A. An interpretable classifier for detection of cardiac arrhythmias by using the fuzzy decision tree. *Artif. Intell. Res.* **2013**, *2*, 45. [[CrossRef](#)]
5. Guler, I.; Ubeyli, E.D. Multiclass support vector machines for EEG-signals classification. *IEEE Trans. Inf. Technol. Biomed.* **2007**, *11*, 117–126. [[CrossRef](#)] [[PubMed](#)]
6. Frénay, B.; De Lannoy, G.; Verleysen, M. Improving the transition modelling in hidden Markov models for ECG segmentation. In Proceedings of the ESANN, Bruges, Belgium, 22–24 April 2009.
7. Chen, X.; Ji, J.; Loparo, K.; Li, P. Real-time personalized cardiac arrhythmia detection and diagnosis: A cloud computing architecture. In Proceedings of the 2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI), Orlando, FL, USA, 16–19 February 2017; pp. 201–204.
8. Sze, V.; Chen, Y.H.; Yang, T.J.; Emer, J. Efficient processing of deep neural networks: A tutorial and survey. *arXiv* **2017**, arXiv:1703.09039.
9. Lipton, Z.C.; Kale, D.C.; Elkan, C.; Wetzell, R. Learning to diagnose with LSTM recurrent neural networks. *arXiv* **2015**, arXiv:1511.03677.
10. Rajpurkar, P.; Hannun, A.Y.; Haghpanahi, M.; Bourn, C.; Ng, A.Y. Cardiologist-level arrhythmia detection with convolutional neural networks. *arXiv* **2017**, arXiv:1707.01836.
11. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
12. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
13. Chen, J.; Samuel, R.D.J.; Poovendran, P. LSTM with bio inspired algorithm for action recognition in sports videos. *Image Vis. Comput.* **2021**, *112*, 104214. [[CrossRef](#)]
14. Shoeb, A.H.; Guttag, J.V. Application of machine learning to epileptic seizure detection. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 975–982.
15. Jambukia, S.H.; Dabhi, V.K.; Prajapati, H.B. Classification of ECG signals using machine learning techniques: A survey. In Proceedings of the 2015 International Conference on Advances in Computer Engineering and Applications, Ghaziabad, India, 19–20 March 2015; pp. 714–721.
16. Woodland, P.C. Speaker adaptation for continuous density HMMs: A review. In *ISCA Tutorial and Research Workshop (ITRW) on Adaptation Methods for Speech Recognition*; ISCA: Sophia-Antipolis, France, 2001.
17. Li, X.; Bilmes, J. Regularized adaptation of discriminative classifiers. In Proceedings of the 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, Toulouse, France, 14–19 May 2006; Volume 1, p. I.
18. Raina, R.; Battle, A.; Lee, H.; Packer, B.; Ng, A.Y. Self-taught learning: Transfer learning from unlabeled data. In Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, USA, 20–24 June 2007; pp. 759–766.
19. Yang, J.; Yan, R.; Hauptmann, A.G. Cross-domain video concept detection using adaptive svms. In Proceedings of the 15th ACM International Conference on Multimedia, Augsburg, Germany, 25–29 September 2007; pp. 188–197.
20. Blitzer, J.; McDonald, R.; Pereira, F. Domain adaptation with structural correspondence learning. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, Sydney, Australia, 22–23 July 2006; pp. 120–128.
21. Dai, W.; Yang, Q.; Xue, G.R.; Yu, Y. Boosting for transfer learning. In Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, USA, 20–24 June 2007; pp. 193–200.
22. Ruder, S. An overview of multi-task learning in deep neural networks. *arXiv* **2017**, arXiv:1706.05098.
23. Lin, S.; Shi, C.; Chen, J. GeneralizedDTA: Combining pre-training and multi-task learning to predict drug-target binding affinity for unknown drug discovery. *BMC Bioinform.* **2022**, *23*, 367. [[CrossRef](#)] [[PubMed](#)]

24. Ji, J.; Chen, X.; Luo, C.; Li, P. A deep multi-task learning approach for ECG data analysis. In Proceedings of the 2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI), Las Vegas, NV, USA, 4–7 March 2018; pp. 124–127.
25. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436. [[CrossRef](#)] [[PubMed](#)]
26. Niu, Z.; Zhong, G.; Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing* **2021**, *452*, 48–62. [[CrossRef](#)]
27. Minsky, M.; Papert, S.A.; Bottou, L. *Perceptrons: An Introduction to Computational Geometry*; MIT Press: Cambridge, MA, USA, 2017.
28. Rafiq, M.; Bugmann, G.; Easterbrook, D. Neural network design for engineering applications. *Comput. Struct.* **2001**, *79*, 1541–1552. [[CrossRef](#)]
29. Liu, S.; Borovykh, A.; Grzelak, L.A.; Oosterlee, C.W. A neural network-based framework for financial model calibration. *J. Math. Ind.* **2019**, *9*, 9. [[CrossRef](#)]
30. Hinton, G.E.; Osindero, S.; Teh, Y.W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527–1554. [[CrossRef](#)]
31. Bengio, Y. Learning deep architectures for AI. *Found. Trends Mach. Learn.* **2009**, *2*, 1–127. [[CrossRef](#)]
32. Bengio, Y.; Courville, A.; Vincent, P. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828. [[CrossRef](#)]
33. Socher, R.; Bengio, Y.; Manning, C.D. Deep learning for NLP (without magic). In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, Jeju Island, Republic of Korea, 8–14 July 2012; p. 5.
34. Hu, B.; Lu, Z.; Li, H.; Chen, Q. Convolutional neural network architectures for matching natural language sentences. *arXiv* **2014**, arXiv:1503.03244. [[CrossRef](#)]
35. Medsker, L.; Jain, L. Recurrent neural networks. *Des. Appl.* **2001**, *5*, 2.
36. Hammad, M.; Abd El-Latif, A.A.; Hussain, A.; Abd El-Samie, F.E.; Gupta, B.B.; Ugail, H.; Sedik, A. Deep learning models for arrhythmia detection in IoT healthcare applications. *Comput. Electr. Eng.* **2022**, *100*, 108011. [[CrossRef](#)]
37. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
38. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
39. Hasan, N.I.; Bhattacharjee, A. Deep learning approach to cardiovascular disease classification employing modified ECG signal from empirical mode decomposition. *Biomed. Signal Process. Control* **2019**, *52*, 128–140. [[CrossRef](#)]
40. Hannun, A.Y.; Rajpurkar, P.; Haghpanahi, M.; Tison, G.H.; Bourn, C.; Turakhia, M.P.; Ng, A.Y. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nat. Med.* **2019**, *25*, 65–69. [[CrossRef](#)]
41. Hu, R.; Chen, J.; Zhou, L. A transformer-based deep neural network for arrhythmia detection using continuous ECG signals. *Comput. Biol. Med.* **2022**, *144*, 105325. [[CrossRef](#)]
42. Shahin, M.; Oo, E.; Ahmed, B. Adversarial Multi-Task Learning for Robust End-to-End ECG-based Heartbeat Classification. In Proceedings of the 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Montreal, QC, Canada, 20–24 July 2020; pp. 341–344.
43. Mormont, R.; Geurts, P.; Marée, R. Multi-task pre-training of deep neural networks for digital pathology. *IEEE J. Biomed. Health Inform.* **2020**, *25*, 412–421. [[CrossRef](#)]
44. Misra, I.; Shrivastava, A.; Gupta, A.; Hebert, M. Cross-stitch networks for multi-task learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3994–4003.
45. Zhang, Z.; Luo, P.; Loy, C.C.; Tang, X. Facial landmark detection by deep multi-task learning. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 94–108.
46. Collobert, R.; Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 160–167.
47. Luong, M.T.; Le, Q.V.; Sutskever, I.; Vinyals, O.; Kaiser, L. Multi-task sequence to sequence learning. *arXiv* **2015**, arXiv:1511.06114.
48. Liu, P.; Qiu, X.; Huang, X. Adversarial Multi-task Learning for Text Classification. *arXiv* **2017**, arXiv:1704.05742.
49. Liu, P.; Qiu, X.; Huang, X. Recurrent neural network for text classification with multi-task learning. *arXiv* **2016**, arXiv:1605.05101.
50. Goldberger, A.L.; Amaral, L.A.; Glass, L.; Hausdorff, J.M.; Ivanov, P.C.; Mark, R.G.; Mietus, J.E.; Moody, G.B.; Peng, C.K.; Stanley, H.E. Physiobank, physiotoolkit, and physionet. *Circulation* **2000**, *101*, e215–e220. [[CrossRef](#)] [[PubMed](#)]
51. Cantzos, D.; Dimogianopoulos, D.; Tseles, D. ECG diagnosis via a sequential recursive time series—Wavelet classification scheme. In Proceedings of the IEEE EUROCON, Zagreb, Croatia, 1–4 July 2013.
52. Bhanot, K.; Peddoju, S.K.; Bhardwaj, T. A model to find optimal percentage of training and testing data for efficient ECG analysis using neural network. *Int. J. Syst. Assur. Eng. Manag.* **2018**, *9*, 12–17. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.