AMERICAN UNIVERSITY OF BEIRUT

DOMAIN ADAPTATION NEURAL NETWORKS FOR TIME SERIES CLASSIFICATION

by
AMIR NASIR HUSSEIN

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to the Department of Electrical and Computer Engineering
of the Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
May 2020

AMERICAN UNIVERSITY OF BEIRUT

DOMAIN ADAPTATION NEURAL NETWORKS FOR TIME SERIES CLASSIFICATION

by

AMIR NASIR HUSSEIN

Approved by:

[Signature]

Dr. Hazem Hajj, Associate Professor — Advisor
Electrical and Computer Engineering

(delegated to Prof. Hajj) 

[Signature]

Dr. Zaher Dawy, Professor — Member of Committee
Electrical and Computer Engineering

(delegated to Prof. Hajj) 

[Signature]

Dr. Stefano Monni, Assistant Professor — Member of Committee
Mathematics

Date of thesis/dissertation defense: May 6, 2020

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: Hussein (Last) Amir (First) Nasir (Middle)

☑ Master's Thesis ○ Master's Project ○ Doctoral Dissertation

☐ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

☑ I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after :

One ---- year from the date of submission of my thesis, dissertation, or project.

Two ---- years from the date of submission of my thesis, dissertation, or project.

Three ---- years from the date of submission of my thesis, dissertation, or project.

[signature] 21/5/2020

Signature Date

This form is signed when submitting the thesis, dissertation, or project to the University Libraries

Acknowledgements

I would like first to present enormous gratitude and thanks to my supervisor Professor Hazem Hajj for his guidance, motivation and unlimited support. I would also like to thank Prof. Stefano Monni and Prof. Zaher Dawy for being a part of my thesis committee and for their helpful feedback and advices. Finally, I want to express a special appreciation and thanks to my family and my colleagues for providing unfailing support and continuous encouragement throughout my academic journey.

An Abstract of the Thesis of

Amir Nasir Hussein for Master of Engineering
Major: Electrical and Computer Engineering

Title: Domain Adaptation Neural Networks for Time Series Classification

In this thesis, we solve two problems related to time-series prediction and domain adaptation (DA). For the first problem, we focus on investigating robust models for time-series prediction with application to epilepsy. Epilepsy is a chronic medical condition that involves abnormal brain activity causing patients to lose control of awareness or motor activity. As a result, detection of pre-ictal states, before the onset of a seizure, can be life-saving. The problem is challenging since it is difficult to discern between EEG signals in pre-ictal states versus signals in normal inter-ictal states. There are three key challenges that have not been previously addressed:(1) the inconsistent performance of prediction models across patients, (2) the lack of perfect prediction to protect patients from any episode, and (3) the limited amount of pre-ictal labeled data for advancing machine learning (ML) methods. The first part of the thesis addresses these limitations through a novel approach that uses adversarial examples with optimized tuning of a combined Convolution Neural Network (CNN) with Gated Recurrent Unit (GRU). Experiments showed that our new proposed solution achieved state of the art. Compared to previous state of the art, the results showed an improvement of 3x in model robustness as measured in reduced variations with area under the curve (AUC) and superior AUC accuracy with an average increase of 6.7%.

In the second part of the thesis, we build on the success of the hybrid CNN-GRU model and investigate the problem of adapting models that have been trained for one source domain to a new target domain. When developing machine learning (ML) algorithms, it is commonly assumed that the training and testing data follow the same probability distribution. However, in real-world scenarios, non-stationary environments are more typical in applications such as Internet of Things (IoT) and wearables where the contexts frequently change over time. The problem can be formulated as domain adaptation (DA), where the settings of the

training labeled data represent the source domain, and the unlabeled test data represent the target domain. The goal of DA is to develop a model that can predict the labels for data in the target domain. The idea is to have one model for source domain and another target domain model that can learn from the source. There has been extensive research on DA for learning domain invariant features. However, those methods remained limited in several aspects when considering advances for time series. Learning between source and target has relied on either using hard parameter sharing limiting the source and target models to be identical or using separate models but making an assumption of a linear relation between source and target parameters. The second open challenge is the model's limited ability to generalize to unseen data for both source and target. The third challenge is ensuring the proper choice of loss function for time-series DA. To address these challenges, we propose a soft sharing DA architecture with squared Maximum Mean Discrepancy (MMD) loss function. The source and target have a similar architecture, consisting of the hybrid CNN-GRU used for epilepsy, but their parameters are modeled with a non-linear relation. For generalization, we augmented the DA architecture with representation learning. We conducted a comprehensive set of experiments for DA with different scenarios of data shifts between source and target domains and showed where hard parameter sharing approach fails. We evaluated the solutions with three cases of DA in the context of activity recognition (AR). The input to the prediction model is multivariate time series data from wearable sensors on a smartphone and a smartwatch. The output is a particular user activity. The first adaptation case captures the scenario where the source domain consists of labeled activities for a group of users, and the target domain is a new user. The second scenario consists of the case where the source domain consists of labeled activities with data collected from one set of devices, and the target domain is a subset of the devices. The third scenario combines the first two cases, and the target domain has a new user and a new set of devices. Compared to the state-of-the-art, the results showed superior improvements up to 8% on average measured in weighted F1-score and reduction in variations of 3.5x on average.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recently, advancements in the internet of things (IOT) and smart wearable technologies have received enormous attention due to their major role in applications such as health care, assessment of treatment efficacy and rehabilitation [2, 3, 4, 5]. By 2022, it is estimated that the number of connected smart wearable devices will be around 1.1 billion worldwide [6]. With a massive amount of time series data streamed by IoT (e.g., smart-watches, smart-phones), there is more demand for sophisticated machine learning algorithms to improve predictive analytics and treatment. This thesis aims at addressing two problems related to time series prediction and domain adaptation.

1.1 Challenges and Proposal for Epilepsy Robust Learning

Several studies investigated the relationship between seizures and brainwave synchronization patterns, highlighting the possibility of distinguishing epileptic patient's states [7]. However, seizure prediction from electroencephalogram (EEG) signals is a challenging task since EEG data varies from one patient to another with the presence of high uncertainty in the seizure onset. Each patient has different patterns and different time schedules for the seizures. In addition, the EEG signals are extremely noisy and affected by other normal brain activities [8].

Many researchers have proposed automatic seizure prediction methods using machine learning techniques. One set of methods rely on statistical methods to extract meaningful features from EEG signals [9, 10, 11]. These methods involve numerous manual intensive steps for feature extraction and differ from one patient to another. Although these methods achieved very high sensitivity and low false prediction rate for a particular dataset, they remain limited in their ability to generalize the performance when tested with new datasets. Recently, another set of methods have relied on Deep Learning (DL) which provide automatic feature

extraction that could reliably identify periods of increased probability of seizure occurrence from EEG signals by [12].

Despite the significant progress in espilepsy analysis, analysis of EEG continues to be faced with several challenges and complexities. The first challenge arises from the complexity of interpreting EEG signals. The task of discerning between pre-ictal, the state before the onset of a seizure, and inter-ictal, normal brain activity, states of different patients poses itself as a difficult task, even for medical experts, due to high inter-patient variability. Moreover, the task of discerning between pre-ictal and inter-ictal states for a given patient is challenging due to high intra-patient variability. The second open challenge is lack of prior methods that achieve robust performance on seizure prediction. In fact, prior state of the art [1] showed high variation in performance across patients. While the performance was perfect for some patients with an area under the curve (AUC) score of 1.0, the performance for other patients reached a low AUC score of 0.3. Such variation renders models unreliable to use for all patients. Third, prediction accuracies still do not meet ideal criteria of 100% accurate prediction. Ideally, we want seizure prediction models to achieve perfect scores of 1.0 AUC across all patients. Missing an opportunity to predict a seizure onset can have severe impacts including limitations on normal daily activities or even the potential of life-threatening scenarios. Best accuracy achieved to date remains at around an AUC score of 0.85 [1]. Finally, due to the nature of seizure disorders, historic patient data with labeled pre-ictal data is not easily available leading to a limited availability of pre-ictal training data for machine learning. This limitation, consequently, impacts the potential accuracy and generalizability that machine learning models can achieve.

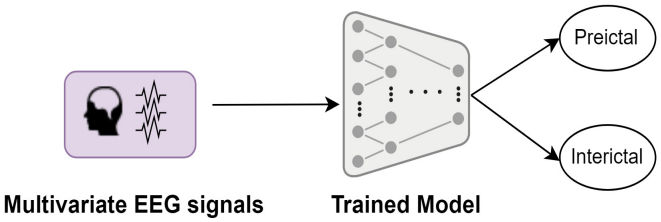


Figure 1.1: The input and the output of the proposed approach. The input is multivariate EEG signals and the output is model classification of pre-ictal or inter-ictal states.

We explore the adversarial learning for robust seizure prediction to address the aforementioned challenges and introduce several contributions:

- The use of adversarial examples (AE) augmentation with EEG time series data for seizure prediction. The approach helps ensure robustness as models get trained with data variations. The approach also helps overcome limitation of training data as more training data becomes available.

- Optimization of the hybrid convolutional neural networks (CNN) and gated recurrent unit (GRU) for seizure prediction.

- The proposed method achieves state of the art performances on two benchmark datasets as measured by AUC, false positive rate per hour (FPR/h), and sensitivity as highlighted next. The comparison to prior state of the art [1], which was based on spectral analysis, indicate the superiority of temporal-based processing.

In terms of robustness of epilepsy prediction, the results showed significant reduction in variations across patients for two benchmark datasets Freiburg [13] and CHB-MIT [10]. The robustness was manifested in the reduction in average standard deviation of AUC across patients with 2x and 2.5x for each data set, a reduction in average standard deviation in sensitivity by a factor of 2x, and a reduction the range of AUC (difference between the maximum and minimum AUC values across patients) with 2.5x and 3x for each data set. In terms of accuracy of pre-ictal prediction, the results showed an average AUC improvement of 2.8% and 6.7% for each dataset respectively. The sensitivity improved by an average of 4% and 1.8% for each dataset respectively. The FPR/h showed an improvement of 8% and 62.5% for each data set respectively.

1.2 Challenges and Proposal for Domain Adaptation

One of the major limitations in the existing approaches to time series prediction is that these algorithms are developed in controlled environments and don't consider the potential shifts in data distribution, called covariate shifts [14], and caused by the dynamic changes in the real world environment [15]. One of the important fields for wearable devices is activity recognition (AR) [16] and its related applications such as industrial assistant [17], fitness [18], and monitoring of elderly people [19]. Shifts in data distribution are very common in AR applications due to variations in the activity patterns among people [16, 20], or heterogeneities of wearable devices such as sensor sensitivity, calibration biases and sampling rate [21].

Domain Adaptation (DA) [22], has been proposed to overcome the problem of covariate shifts in the data distribution. DA approaches facilitate semi-supervised learning to adapt to new similar domains with unlabeled data [22]. Earlier DA methods for AR focused on extracting features that are not sensitive to shifts in data distribution [23, 24]. Their major limitations were in the features extracted that differed from one dataset to another and required domain knowledge [16, 25]. To overcome the limitations of feature-based approaches, deep learning DA techniques were introduced [26, 27], but the proposed architectures were limited to

hard parameter sharing, which enforces a common network for source and target domains. Recently, two seminal works in the field of computer vision (CV) were introduced to address the limitation of predefined shared architecture. In [28], the authors introduced a DA architecture that consists of four components including: one component to learn similarities between source and target domains, two components to learn differences between source and target domain representations and the fourth component to reconstruct the input data from the learned representation. In addition, researches in [29] presented a less complicated architecture that consists of only two separate models, one for the source and one for the target domains, and assumed a linear relationship between the parameters of the two domains.

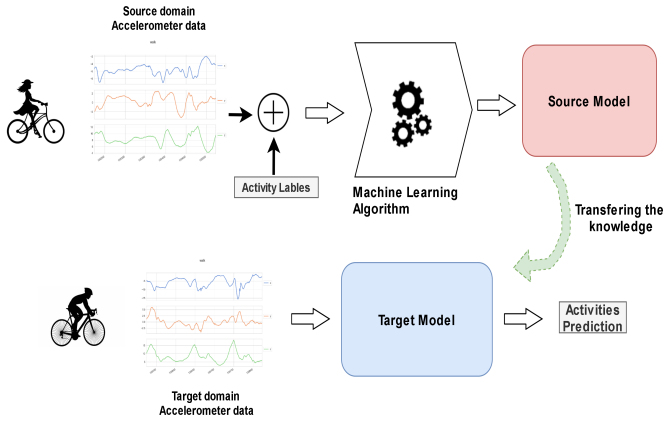


Figure 1.2: Illustration of the high level proposed approach

Despite these DA advances, previous work remains limited in several aspects that we address in this paper: 1) Relying on hard parameter sharing between source and target models forcing them to be identical 2) Relying on separate models for source and target but assuming a linear relation between the models' parameters 3) Poor generalization with DA methods used for time series prediction. To address these challenges, we propose soft parameter sharing architecture that includes representation learning for generalization and squared MMD as the domain discrepancy loss. The general description of the proposed approach is illustrated in Fig. 1.2 where the model is developed using source data on activity recognition collected from one participant, and the knowledge is then transferred to the target domain to another participant with a different data distribution. The proposed DASH approach is able to successfully make predictions in the presence of covariate shifts by learning domain specific and shared characteristics. We conducted a study of different data shift scenarios between source and target domains and showed success and failure cases. We also evaluated the proposed DA solution for different domain shift scenarios with AR data.

- *Cross user domain adaptation:* In this case, the source data is from one

user and the target data is from another user with same wearable device. The source of shift in data distribution is the differences in data for the same activity but coming from a different user.

- *Cross device domain adaptation:* In this case, the source data is from a smartphone and the target data is for the same user but using different smartphone. The source of shift in data distribution is in different specifications of the target device.

- *Cross user and cross device domain adaptation*: This is a more challenging and realistic scenario where there are two sources of shifts in data distribution: a new device and a new user. In this case, the source data is from one user with their own smartphone and the target data is from another user and another smartphone.

In summary the key contributions of our work include:

1. A soft parameter sharing DA architecture with nonlinear modeling of relation between source and target model parameters for time series data.

2. Improved DA generalizibility by integrating representation learning.

1.3 Thesis Organization

The rest of the thesis is organized as follows: Chapter 2 presents proposed robust learning approach for seizure prediction. Chapter 3 covers proposed domain adaptation approach for time series prediction. Chapter 4 summarizes the findings from both works and concludes the thesis.

Chapter 2

Robust Learning for Epilepsy Prediction

2.1 Related Work

Related work on the use of machine learning techniques for epilepsy classification can be categorized into feature-based and deep learning approaches. Within each category, researchers have examined two different problems of seizure detection and prediction but often examined similar features in the signal. These methods are further detailed here.

2.1.1 Feature-based Approaches

Seizure Detection:

Feature engineering techniques that are tailored for seizure detection for specific patients have successfully achieved very high sensitivity 89.66% with FPs/h value of 0.49 were obtained for 21 patients [30] and (100%) with very low false positive rate on the same patients [31, 32]. In [10] authors developed a machine learning framework that is capable of identifying the features critical for seizure detection. They used support vector machine to construct patient-specific models while considering sensitivity, specificity and latency as performance metrics. This work provided a new approach for EEG epilepsy data preprocessing as well as feature extraction and evaluation methodology. Although these methods achieved very high sensitivity, they are heavily biased to their specific dataset. Moreover, they are used only for seizure detection rather than prediction.

For generalizability of epileptic seizure detection using MTL, a feature-based patient-specific MTL-SVM model was proposed in [33]. The extracted features were obtained by filtering the EEG signal of each channel using four filter-banks with frequencies ranging from 0.5 to 25 Hz and then calculating the energy falling within each band similarly to the work done in [10]. Moreover, the proposed

model was developed to learn a general representation of the various patient-specific seizures in order to generalize better to all the different types of patient-specific seizures. Although this work targeted the problem of increasing the model's generalizability, the model's features were not sufficient for seizure prediction as they only consist of the energy features from the filter bank bands which are not sufficient to detect changes in the EEG signal representative of pre-ictal states.

Seizure Prediction:

Seizure prediction mainly targets detecting a transitional period between the inter-ictal and ictal states called the pre-ictal state [34]. Different signal processing techniques were explored in previous work for seizure prediction using machine learning approaches. For example, in [35], the plausibility of a combination of frequency and time domain features was explored for epilepsy seizure prediction. The proposed feature-vector included auto-regressive fitting error, decorrelation time, energy, Hjorth mobility and complexity, spectral power in the delta, theta, alpha, beta, and gamma bands, spectral edge (power, frequency), the four moments (mean, variance, skewness, kurtosis), energy wavelet coefficients for six decomposition levels. The proposed approach was tested for 216 patients and achieved 38.47% sensitivity and 0.2 FPR/h on average for all patients while achieving statistical significance for only 24 patients. In [11], EEG signal segments were filtered to obtain four frequency bands which are the delta, theta, alpha, beta, and gamma. From each band the following features were extracted: 1) normalized spectral power features 2) the four moments 3) Hjorth activity, mobility, and complexity features 4) the accumulated energy of signal 5) the auto-regressive (AR) error resulting from fitting an order 10 AR model 6) decorrelation time 7) spectral edge power 8) wavelet coefficients. In addition, the experiments included deciding the optimal combination of the pre-ictal time, normalization methods, smoothing and outlier removal. It was found that smoothing, outlier removal, and normalizing by the maximum value of each feature provided the best results for most of the patients. The proposed method achieved an average sensitivity of 73.9%, with an FPR/h of 0.15 on average over 10 patients. The previous methods mainly depended on univariate features rather than multivariate features- features that are extracted from a combination of multiple channels. For example, in [36], Bivariate spectral band power features were suggested for seizure prediction. The proposed features achieved an average sensitivity of 75.8% and FPR/h of 0.1 over 24 patients. In [7], the use of non-linear bivariate features such as wavelet synchrony was explored. Furthermore, the proposed features achieved an average sensitivity of 71%, with zero FPR/h on average over 15 patients.

Although different recommended set of features can be found in the literature for feature-based seizure prediction, no specific set of features has been proven

as the best set for predicting seizures [37]. In addition, the proposed methods require domain knowledge, and may not perform similarly for different patients or different datasets.

2.1.2 Deep Learning Approaches

Deep Neural Networks (DNN) models like Convolutional Neural Networks CNNs and Recurrent Neural Nets (RNNs) have proven to be very effective in automatically extracting features from time-series sequences and learning temporal dynamics [38, 39]. In [40] researchers introduced an automatic seizure detection approach that is robust against noise in real-life conditions. They used Long Short Term Memory (LSTM) with time distributed dense layer to automatically extract robust features from EEG signals. In a study of model performances for predicting epileptic seizures, a comprehensive comparison was conducted by [41]. It was noted that a model consisting of a CNN followed by a Long Short-Term Memory (LSTM) outperformed (Hidden Markov Model (HMM), HMM-Stacked denoising Autoencoder (SdA), HMM-LSTM, Incremental Principal Components Analysis (IPCA) -LSTM, CNN Multiple layer perceptron) in terms of sensitivity and false alarm rate.

For seizure prediction, the authors of [42] used 1D-CNN consisting of 5 convolution layers for automated seizure prediction from raw EEG signals. This approach was tested only on FB dataset [13]. In [43], authors proposed the use of wavelet transform (CWT) for EEG signals as a preprocessing step before feeding the data to CNN model. In [12], a deep learning model that can run on a low powered device was proposed for performing real-time seizure prediction using intracranial EEG signals that are obtained from the surface of the brain. The proposed model can be retrained automatically using the users' data during the usage period, where after each month the model can be fine tuned using the new recorded data from the patient. The model also can be run on a smart watch, and provide predictions that are better than a random predictor by 42% achieving a mean sensitivity of 69%. The provided model is user-specific, requires recording data from users for 2 months before starting the prediction, and is not reliable enough for real-world usage as it has only been tested for 15 patients and one dataset. In addition, all of the foregoing approaches were not evaluated in the case of keeping entire pre-ictal state for testing while training on the rest of the data to ensure generalization. In [1], researchers provided a generalized approach for seizure prediction where they used 2D CNN with only three layers to avoid overfitting. The EEG signals were converted to image like data using Short-Time Fourier Transform (STFT) to make it suitable for 2D CNN. However, the main weakness of this method is that it does not model the signal's temporal dynamics and long term dependencies.

2.1.3 Generalization Approaches

It has been shown by researches that studying how deep networks fail and hardening them against adversarial attacks would help in better understanding how DNNs work and improve their generalization ability [44, 45, 46, 47]. It was suggested by [48] that the AEs have different statistical distributions compared to the original data from which they were generated. This showed that AEs are statistically different from the dataset they were generated from. In [49], authors proposed augmenting training data with AEs to improve network robustness. Additionally, [50] introduced training with an adversarial objective function that behaved as a better regularizer in comparison to dropout and achieved better generalization. As a result, training on AEs was used by [51] to increase the robustness of deep neural network for speech recognition against noise and channel variations.

In summary, none of the existing methods address the model robustness to noise from other brain activities and the variation across patients for seizure prediction

2.2 Proposed Machine Learning Models

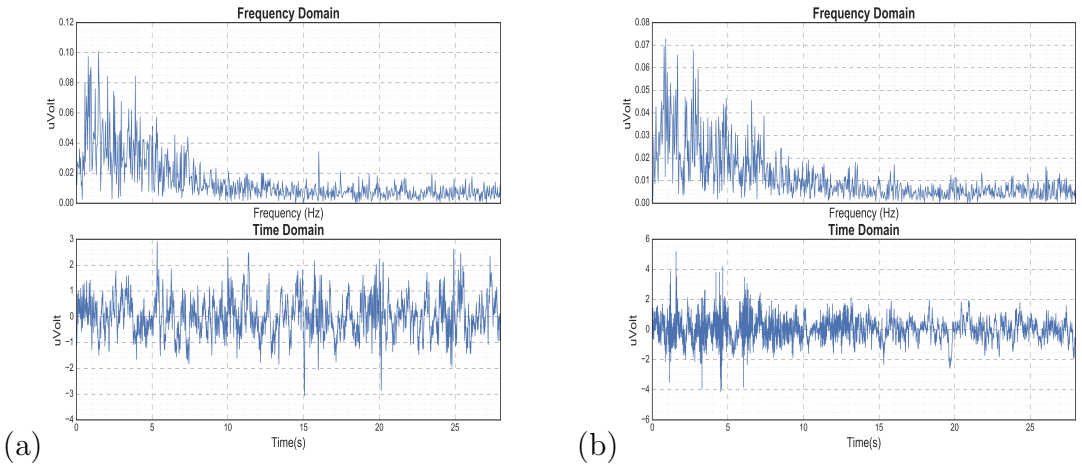
The objective of this work is to develop an approach suitable for automatic feature extraction from raw EEG signals, accurately detect pre-ictal seizure states and robust against the noise in EEG signals. One of the main challenges in seizure prediction is that some of the inter-ictal states resembles pre-ictal states as shown in Figure 2.1. It can be seen that sometimes, what seems a pre-ictal signal for one patient may seem to be an inter-ictal signal for another patient and vice versa.

Another more challenging situation is that the inter-ictal state might resemble pre-ictal states for the same patient as shown in Figure 2.2.

The high level solution for proposed adversarial examples (AEs) is shown in Figure 2.3. Figure 4 describes the high level steps of learning with (AEs) where the proposed model is first trained with the EEG signals and then the trained model is used to generate AEs. The training data is then combined with AEs to retrain the model on the augmented data. The input to the system consists of EEG signals recorded from skin electrodes outside of the skull (scalp EEG), and implantable electrodes on the surface of the brain, (intracranial EEG). The output of the system is a seizure state prediction, inter-ictal or pre-ictal state.

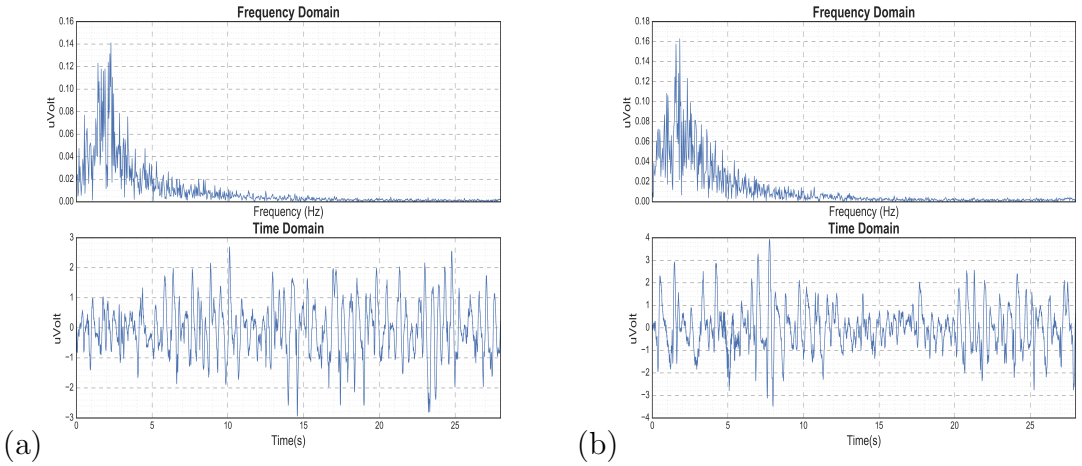
2.2.1 Data Pre-processing

The EEG data was filtered using notch filter to remove the power line noise. Freiburg dataset is contaminated with power frequency at 50 Hz, and CHB-MIT dataset is contaminated with power frequency at 60 Hz. As a result, the components at frequency range of 47–53 Hz and 97–103 Hz and 57–63 Hz and 117–123



(a) (b)

Figure 2.1: EEG signals from CHB-MIT dataset. a) Patient(1) inter-ictal signals in both frequency and time domains. b) Patient(3) pre-ictal signals in both frequency and time domains. The 'X' axis for each subfigure in frequency domain represents frequency in Hertz and in time domain represents time in seconds. The 'Y' axis for both subfigures is the amplitude in micro-volts.



(a) (b)

Figure 2.2: EEG signals from CHB-MIT dataset. a) Patient(14) pre-ictal signals in both frequency and time domains. b) Patient(14) inter-ictal signals in both frequency and time domains. The 'X' axis for each subfigure in frequency domain represents frequency in Hz and in time domain represents time in seconds. The 'Y' axis for both subfigures is the magnitude in uVolts.

Hz are removed for Freiburg dataset and CHB-MIT dataset, respectively. After that the data is normalized using z-score to ensure zero mean and unit variance across all channels. After that EEG signals are segmented with sliding window of length 30s and 50% overlapping to ensure stationarity. The stationarity of windowed signals were checked using Augmented Dickey-Fuller (ADF) test which is

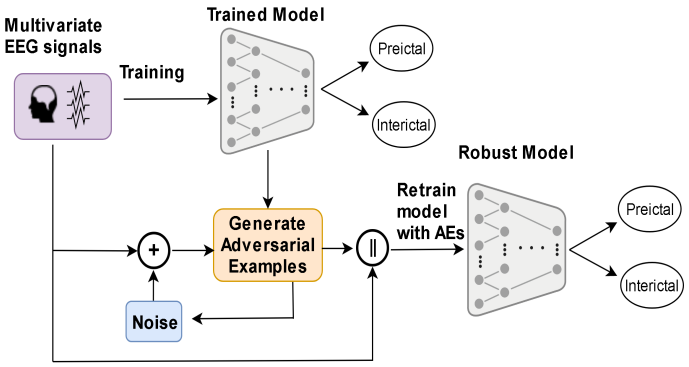


Figure 2.3: High level block diagram for Learning with Adversarial Examples approach

one of the unit root tests that uses an auto-regressive model and optimizes an information criterion across multiple different lag values [52]. In the ADF test we firstly define the null hypothesis which states that the signal can be represented by a unit root which indicates time-dependent structure in the signal and hence its non-stationarity. We specified the significance level to be 5%. After applying this test method on the windowed signals, we found that the p-values of all of the generated segments were significantly lower than 5%. Hence we reject the null hypothesis, and conclude that there is sufficient evidence that the generated segments are stationary.

2.2.2 CNN-GRU Model

The proposed model is shown in Figure 2.4 with shorthand descriptions as follows: $C(f, k, s)$: representing a convolution layer with $'f'$ number of filters, $'k'$ size of the kernel and $'s'$ number of strides. The model consists of convolution layers for feature extraction by stacking several operators to create a hierarchy of abstract features. To process the EEG time series, 1D convolution operation is used to model temporal sequences information. Each convolution kernel acts as a filter, that filters out the time series data and detect relative patterns. In addition, the convolution kernels performs depth-wise filtering of the multivariate signal where weights corresponding to each channel are learned during the training phase and hence result in the best integration of signals through channels.

The recurrent layer is composed of Gated Recurrent Units (GRUs) to model the time dependencies in the EEG signals. GRUs are special kind of recurrent units that have update and reset gates allowing the model to decide how much historical information to keep. This property enables the proposed deep architecture to model temporal dynamics of time series as well as the long term dependencies.

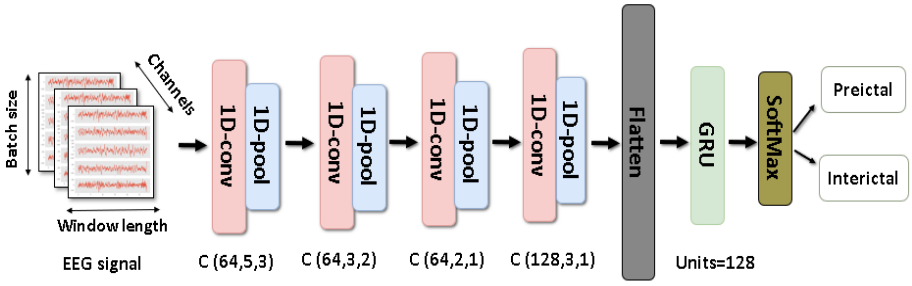


Figure 2.4: CNN-GRU architecture where the batch size is 256 , the window length of the EEG signal is 30 sec and the number of channels is 6 and 22 for the FB and CHB-MIT datasets, respectively.

2.2.3 Adversarial Learning

Generally, a well trained machine learning model $f(x;\theta)$ capture the nonlinear relation between the input x_i and the output of the model y_i, where θ represents model parameters. To provide better generalization for the learning model, we are proposing to use the idea of augmenting the training data with adversarial examples which improve model robustness against adversarial examples as well as the noise from real-life conditions that corrupts EEG signals: muscle artifacts and eye blinking.

It has been shown in the literature that the state-of-the-art models can easily misclassify examples that are slightly changed from the original data [50]. In addition, it has been shown that the classification decisions of the neural networks are linear in higher dimensions and it only requires to know the direction of the perturbation to cross that classification boundary and mislead the model prediction. As a result, we expect that training with adversarial examples approach will help the model to better differentiate between different EEG signals that have high resemblance and reduce the false alarm rate caused by noise.

AEs are a special kind of data that are generated by adding noise to the original input data that is optimized to mislead the model classification. The added adversarial noise is not perceptible by humans. In this work, we assume a white-box setting where we have access to the model parameters θ. In order to generate an adversarial example x' to the model $f(x';\theta)$, a small amount of noise σ is obtained by computing the gradient with respect to the input that leads the network to wrong classification y_t as shown in Eq. (3.12-2.4).

$$\sigma = (\nabla_{\boldsymbol{x'}} J(f(x';\theta), y_t)) \tag{2.1}$$

1.2

$$x' = x + \sigma \tag{2.2}$$

so that:

$$f(x';\theta) \neq f(x;\theta) \tag{2.3}$$

and

$$\sigma \ll x \tag{2.4}$$

The process of generating AEs is further illustrated in Figure 2.5. The first step is to train the model $f(x;\theta)$ on the actual data by minimizing the loss function $J(f(x;\theta),y)$ with respect to the model parameters θ. The loss function $J(f(x;\theta),y)$ is the average cross-entropy. In the second step, the model parameters are held constant and the required noise $'\sigma'$ to be added to the input, that makes the network mis-classify corresponding output, is obtained by calculating the gradients with respect to the input. Generating AEs requires finding x' where $x' = x + \sigma$, that minimizes the loss function of the corresponding chosen target class y_t, where $y_t \neq y_{true}$, y_{true} is the true corresponding class for x, M is number of classes, N is number of samples in the batch as presented in Eq (2.5, 2.6).

$$J(y_t,\hat{y}) = -\frac{1}{N}\sum_{n}^{N}\sum_{m}^{M} y_t^m \ log(\hat{y}^m) \tag{2.5}$$

$$\min_{x'}(J(y_t,\hat{y}) + \lambda\|\sigma\|^2) \tag{2.6}$$

L2 norm regularization added to the loss $J(y_t,\hat{y})$ in Eq. (2.5) is used to penalize for large noise values during the optimization, where λ represents the importance of minimizing the noise. The regularization ensures that the generated AEs are close to the original input.

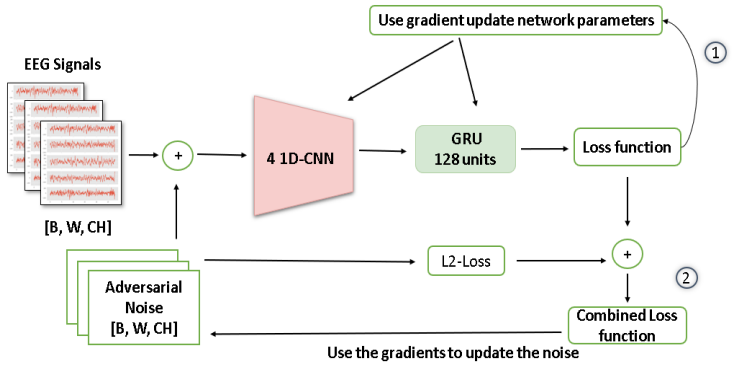


Figure 2.5: Block diagram of adversarial examples generating approach. The input is a three dimensional tensor [B, W, CH] where 'B' represents the batch size, 'W' represents the window length of the EEG signal and 'CH' is number of channels which corresponds to the number of electrodes used for EEG signals recording.

2.3 Experiments & Results

The proposed approaches were assessed through the receiver operating characteristic curve (ROC) that plots true positive rates versus false positive rates for all classification thresholds. The area under the curve (AUC), sensitivity and false positive rate per hour FPR/h were calculated to quantify each model's performance. To obtain FPR/h, k-of-n analysis was performed to maintain consistency with [1] where for every n prediction, the alarm only raised if there are at least k positive predictions. In [1], 'k' and 'n' values were chosen to be 8 and 10, respectively. Since the data is segmented with window of 30 seconds, the model produces a prediction every 30 seconds. If the alarm is raised it will be counted as 1 alarm during 35 minutes.

In order to perform fair comparison with [1], leave-one-seizure-out cross validation approach was used for each patient of the two datasets. At each fold if the patient has N pre-ictal states one state was held for testing and N-1 pre-ictal state were used for training. Inter-ictal states were randomly split into N parts but each of the inter-ictal states is much longer than the pre-ictal state. The training data was balanced such that the number of pre-ictal states is equal to the number of inter-ictal states. Hence at each fold N-1 of inter-ictal and N-1 pre-ictal were used for training and one pre-ictal state with one intarictal state were used for testing. Furthermore the training data is divided into 90% for training and 10% for monitoring the learning process of the model at each epoch as shown in Figure 2.6.

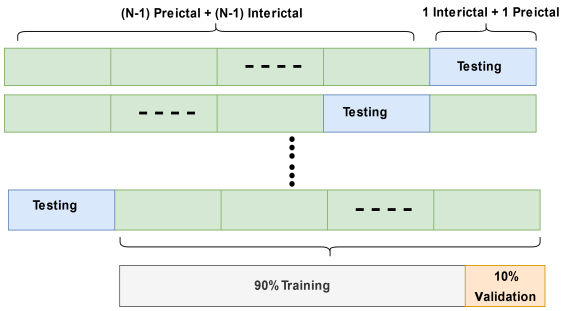


Figure 2.6: Illustration of leave-one-seizure-out cross validation approach

2.3.1 Dataset

To compare our results with those obtained by [1], same data preparation steps were followed to extract pre-ictal and inter-ictal seizure states for each patient for two datasets Freiburg Hospital dataset (FB) [13] and CHB-MIT dataset [10]. FB dataset contains intracranial EEG (iEEG) signals that were recorded from 21 patients using 6 channels and sampling rate of 256 Hz. However, only 13 patients were chosen due to lack of availability of the full FB dataset. The CHB-MIT

dataset consists of scalp EEG (sEEG) data from 23 patients. This data was recorded using 22 channels for each patient for 844 hours at a sampling rate of 256 Hz.

In the work of [1] the inter-ictal state is defined as the period between at least 4 hours before seizure onset and 4 hours after seizure end. Multiple seizures may occur closely to each other, hence the pre-ictal state that occur less than 30 min from the previous one is considered as one pre-ictal state and then the seizure prediction task became predicting the leading one. Furthermore, some patients have seizures every 2 hours on average which is not adequate for the seizure prediction. Thus, only patients with fewer than 10 seizures per day are considered. With these definitions, only 13 patients from both CHB-MIT and FB datasets were chosen. The pre-ictal state was considered to be 30 minutes that ends 5 minutes before the seizure onset, to give the patient enough time to act adequately.

2.3.2 Tuning Parameters for the CNN-GRU Model

To perform the hyper-parameter tuning for the CNN-GRU model, we followed same approach as [1] where training and testing samples where selected from different periods to avoid overfitting. Specifically the later 30% of both inter-ictal and pre-ictal samples were selected for testing whereas the first 70% of the data was used for training for both CHB-MIT and FB datasets. The choice of the architecture size and complexity have been carefully examined to ensure that the model is not too complex to overfit the data not too shallow to underfit. We used AUC metric as an indicator of balancing between true positive and false positive. Convolution layers were varied from (1-6) layers with number of kernels that were chosen from (32, 64, 128, 256). The kernel size and the number of strides were chosen such that the filter will overlap by at least 30% each time a stride is performed. The size of kernels were sampled from (3, 5, 7, 9, 11) and the corresponding strides for each kernel are (1, 2, 3, 4, 5). In addition, max-pooling operation was performed after each convolution layer with size and stride of 2. The number of layers that gave the best performance was found to be 4 layers as shown in Figure 2.7.

The number of recurrent layers was changed from (1, 2, 3) and the number of units were chosen from (32, 64, 128, 256). It was found that increasing the number of layers beyond 1 layer as well as increasing the number of units beyond 128 did not improve the accuracy. Hence one layer of GRU with 128 units was chosen.

2.3.3 Adversarial Learning

In order to generate adversarial examples, Eq. (2.5, 2.6) were used in the optimization process and the approach described in Figure 2.5 was followed. In

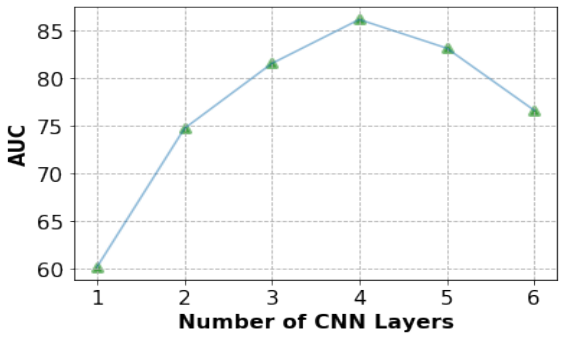


Figure 2.7: CNN-GRU AUC performance for different number of layers

addition, l_2 norm was used on the noise such that it did not exceed 1% of the l_2 norm of the original EEG signal. The generated examples consisted of 50% inter-ictal and 50% pre-ictal since signals with the inter-ictal state were down-sampled as in [1] to make the data balanced. The best value for λ shown in Eq.(2.6) was chosen 0.001, which ensured quick convergence (200 steps) and guaranteed having valid EEG signal that cannot be distinguished from the original EEG signal as shown in Figure 2.8. Only the training data was augmented with AEs at each fold during the one-seizure-leave-out cross validation. This configuration was chosen to measure the exact performance on the testing data after training data augmentation.

To validate that the AE augmentation does not break stationarity of the windows, we ran the Augmented Dickey-Fuller (ADF) test, which is commonly used to check for stationarity. The ADF test showed that the p-values of all of AE generated segments were significantly lower than 5%, ensuring that stationarity is conserved.

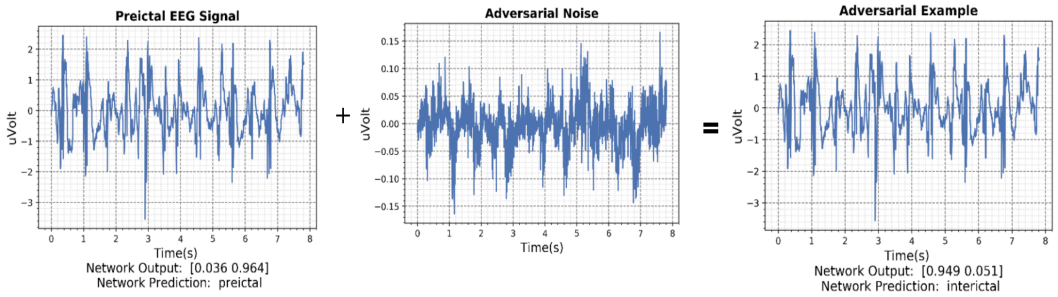


Figure 2.8: Example of perturbing the classification of a pre-ictal EEG signal with adversarial noise. The 'X' axis time in seconds and the 'Y' axis is the amplitude in micro-volt.

Table 2.1: Comparison of the results of CNN-GRU model with and without AEs augmentation, training with the MTL approach, and the prior state-of-the-art [1] approach on Freiburg Hospital inter-ictal EEG dataset. It can be seen that training on data augmented with adversarial example reduces the false rates and improves sensitivity.

Patients	Training Samples	Sensitivity (%)	FPR/h	Sensitivity (%)	FPR/h	Sensitivity (%)	FPR/h	Sensitivity (%)	FPR/h
		CNN-GRU without AEs augmentation		CNN-GRU with AEs augmentation		training with Multitask Learning		Prior state-of-the-art [1]	
1	765	75.78	0.00	76.22	0.00	74.21	0.12	100	0.00
3	1020	75.51	0.00	80.66	0.25	62.20	0.09	100	0.00
4	1020	92.12	0.00	92.53	0.00	91.97	0.00	100	0.00
5	1020	77.63	0.82	74.51	0.00	54.33	0.28	40	0.13
6	510	93.43	0.00	98.16	0.00	98.95	0.15	100	0.00
14	765	72.63	0.00	83.46	0.00	65.75	0.00	50	0.27
15	765	81.70	0.46	85.62	0.00	82.28	0.35	100	0.02
16	1020	82.04	0.00	92.53	0.46	70.87	0.09	80	0.17
17	1020	80.62	0.00	95.90	0.00	95.91	0.028	80	0.00
18	1020	81.62	0.00	75.07	0.00	62.20	0.09	100	0.00
19	765	70.50	0.00	73.82	0.00	78.35	0.12	50	0.16
20	1020	72.70	0.35	74.17	0.00	84.09	0.37	60	0.04
21	1020	74.65	0.57	78.70	0.52	81.10	0.46	100	0.00
Average		79.30 ±7.08	0.14 ±0.22	**83.18** ±8.90	**0.055** ±0.135	77.09 ±23.56	0.18 ±0.15	81.4 ±23.39	0.06 ±0.09

2.3.4 Discussion

Tables (2.1,2.2) summarize the results for CNN-GRU base model trained with and without AEs augmentation on both FB and CHB-MIT datasets in comparison to the MTL augmentation approach and the prior state of the art [1]. Furthermore, we compare our proposed augmentation with AEs to data augmentation with Gaussian Noise (GN). The sensitivity, FPR/h and AUC was obtained by averaging the results over the number of folds from one-seizure-leave-out cross validation for each patient.

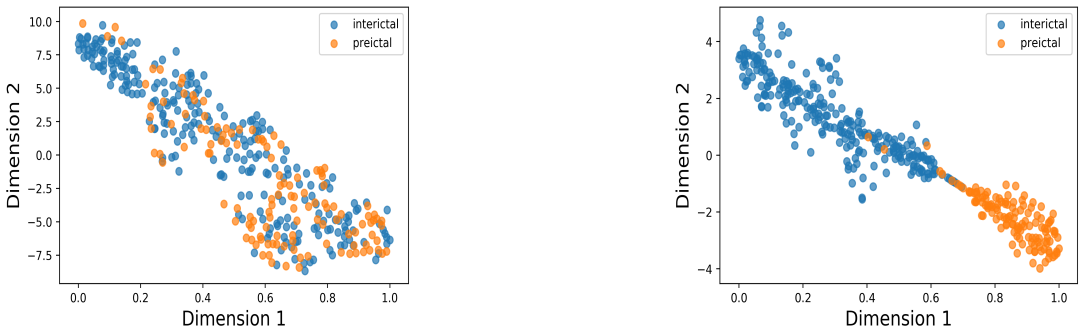
It can be seen that, our proposed model with AEs augmentation achieved significantly high sensitivity 83.18% and 85.16% on average with low FPR/h of 0.055 and 0.06 for FB and CHB-MIT datasets respectively. Comparing our proposed approach to the prior work of [1] we notice that it achieves 3.96% better sensitivity and reduction in relative percentage of 62.2% FPR/h on average for CHB-MIT dataset. On the other hand, for FB dataset our proposed approach achieved 1.8% better sensitivity and reduction in relative percentage of 8% FPR/h on average. In addition, the results show significant reduction in average standard deviation in the sensitivity by a factor of around 2X in both CHB-MIT and FB datasets compared to state-of-the-art [1]. It also can be noted that the proposed CNN-GRU model achieves a higher sensitivity on average and lower FPR/h on CHB-MIT dataset compared to the FB datasets with and without AEs augmentation. These results are reasonable since the CHB-MIT is richer in data with 22 channels versus the FB dataset iEEG signals, which only had 6 channels.

We now further analyze and discuss the effect of training the CNN-GRU model on data augmented with AEs on both FB and CHB-MIT datasets. At each fold, only the training data was used to generate AEs as described in Section 2.2.3 while testing data kept untouched. One can notice that after augmenting the

Table 2.2: Comparison of the results of CNN-GRU model with and without AEs augmentation, training with the MTL approach, and the prior state-of-the-art [1] approach on CHB-MIT dataset. It can be seen that training on data augmented with adversarial example reduces the false rates and improves sensitivity.

Patients	Training Samples	Sensitivity (%)	FPR/h	Sensitivity (%)	FPR/h	Sensitivity (%)	FPR/h	Sensitivity (%)	FPR/h
		CNN-GRU without AEs augmentation		CNN-GRU with AEs augmentation		training with Multitask Learning		Prior state-of-the-art [1]	
1	1530	92.03	0.00	95.95	0.00	92.46	0.07	85.7	0.27
2	510	67.70	0.00	72.07	0.00	61.42	0.15	33.3	0.00
3	1275	71.44	0.17	75.27	0.00	45.67	0.23	100	0.18
5	1020	68.12	0.22	74.17	0.13	83.15	0.00	80	0.19
9	585	93.07	0.00	96.40	0.00	90.55	0.00	50	0.12
10	1277	64.69	0.00	66.17	0.00	46.33	0.12	33.3	0.00
13	1020	92.32	0.13	95.95	0.00	72.44	0.18	80	0.14
14	1020	66.29	0.54	64.80	0.46	61.73	0.09	100	0.40
18	1275	85.83	0.00	88.62	0.00	44.23	0.15	100	0.28
19	510	98.95	0.00	88.62	0.00	44.23	0.15	100	0.00
20	1020	99.37	0.25	99.64	0.18	95.91	0.09	100	0.35
21	765	80.11	0.14	81.70	0.00	61.22	0.23	100	0.23
23	610	96.85	0.00	97.37	0.00	46.73	0.00	100	0.33
Average		82.82 ±13.55	0.11 ±0.16	**85.16** ±13.22	**0.06** ±0.13	64.97 ±26.43	0.11 ±0.15	81.2 ±25.91	0.16 ±0.135

data with AEs the sensitivity improved for most of the patients and FPR/h decreased for both datasets. Training on the adversarial examples works as a regularizer that prevents the model from overfitting. It was noticed from AUC results of patient 14, FB dataset, that jumped from 0.72 to 0.98 after augmenting the data with AEs. Similar results were observed for patients (2,3,5) from CHB-MIT dataset and patients (5, 15,20) FB dataset. The improvements in the augmented results can also be explained from the idea that training the model on adversarial examples allows the model to draw more robust boundary decisions for classification task since those examples were explicitly crafted to mislead the model prediction and at the same time to be visually indistinguishable from the original data. Further analysis was performed to investigate this idea by visualizing the embeddings of the base model trained on original data and embeddings of the base model trained with AEs augmentation using t-distributed stochastic neighbor embedding (t-SNE) [53] as shown in Figure 3.8. It is obvious that



(a) (b)

Figure 2.9: The 2D t-SNE visualization of patient 15 embeddings, FB dataset, for pre-ictal and inter-ictal classes. a) The embeddings of proposed CNN-GRU approach without AEs. b) The embeddings of of proposed CNN-GRU approach with AEs.

the two classes have much better separations boundaries after training with AEs augmentation. These results were noticed for other patients as well from both datasets. Another advantage of augmenting the training data with AEs is that this effectively increases the size of the training set without requiring new data. The obtained results prove the effectiveness of augmenting the data with AEs for EEG signals and suggests improvement in model generalizability. Finally, to compare both sensitivity and the FPR/h for all thresholds, AUC results were obtained for both FB and CHB-MIT datasets. Those results were obtained after 50 epochs using mini-batches of size 256. The averaged AUC across all patients and its variance were used to compare our proposed CNN-GRU approach with AEs augmentation and the work of [1], as shown in Figure 2.10. Overall our pro-

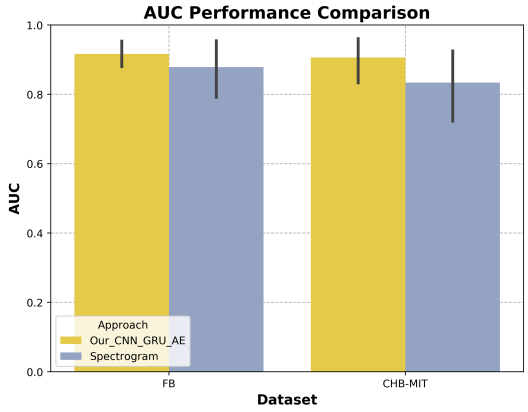


Figure 2.10: The comparison of AUC performance accross all patients between our proposed CNN-GRU model with AEs augmentation and State-of-the-Art spectrogram approach [1]. It can be seen that results of our proposed approach achieves higher AUC on average with has less variance within each dataset and across the two datasets.

posed approach achieved AUC of 90% and 89% on average on CHB-MIT and FB dataset respectively which are significantly higher than the spectrogram approach [1] with 83.3% and 86.2%. In addition, our AUC results show smaller variation in the range (difference between maximum and minimum AUC values across patients) across patients within each dataset where the range of AUC reduced by factor of 3X (from 0.7 to 0.23) and 2.5X (from 0.5 to 0.20) for CHB-MIT and FB respectively. As a result, we notice a significant reduction in average standard deviation of AUC across patients by a factor of 2X and 2.5X for CHB-MIT and FB respectively illustrated in Figure 6. These results show the robustness of our method where it achieves higher AUC with less variance. This is expected as most of the previously proposed methods for seizure prediction are overfitting to a particular dataset under the study and fail to maintain consistent results across different datasets.

Chapter 3

Domain Adaptation for Time Series Prediction

3.1 Related Work

This section presents the prior work in transfer learning to address covariate shifts in data distribution. Most of the previous work in the field can be grouped into multitask learning (MTL), domain adaptation (DA) under feature-based modeling techniques and DA under deep learning techniques.

3.1.1 Multitask learning approaches

Multitask learning (MTL) is the approach where the model learns multiple tasks by simultaneously optimizing more than one loss. Researchers have showed that MTL can address limitation with data availability, provide superior performance and improve generalization [54, 55, 56]. A lot of work has been done for time-series classification using MTL, like predicting the location of certain proteins within a cell [57] and performing personalized human activity recognition [58]. In [59] authors used MTL deep network to make predictions on correlated time series. They used Convolutional Neural Network CNN followed by Recurent Neural Network RNN configuration, and they also added auto-encoder layer to reconstruct the signal from representation encoded by the CNN. The major limitation of MTL approaches is that they depend on predefined similarities between the tasks. Often in situations when tasks are weakly related, the MTL approach performance falls short in accuracy prediction for the target domain [22]. In addition, MTL requires labeled data in the new domain which could be expensive and not always feasible [60].

3.1.2 Domain Adaptation

Domain Adaptation is a subset of transfer learning where the aim is to transfer knowledge from source domain to a related, but different, target domain in the presence of shifts in data distribution [22]. Approaches introduced in the literature for time series classification tasks can be grouped into two main categories: feature-based techniques and deep learning (DL) techniques.

Feature-based Domain Adaptation for Activity Recognition

Feature based DA heavily relies on manually extracting features that are robust to covariate shifts [24]. In [61] authors proposed extracting robust features like covariance between the axes and entropy to recognize activities in two scenarios: 1) Recognize complex activities with less number of sensors 2) Recognize complex activities for different users that were not included in model training process. They used random forest and SVM with RBF machine learning algorithms. In the work of [62] expectation maximization was combined with conditional random fields to recognize actions for different datasets. Furthermore in [63] unsupervised adaptive classifier was proposed to adapt to the changes in data distribution that result from sensor displacements or slippage. The proposed approach calibrates itself using an online version of expectation–maximization algorithm called Levenberg–Marquardt. The approach is mainly based on assumption that the main change in the feature distribution corresponds to only a shift of an unknown but related magnitude and direction. In [26] authors introduced stratified transfer learning to transfer the knowledge from the source domain to the target domain. They generated pseudo labels for target domain using majority voting with classifiers trained on the source domain. After that [26] utilized the similarity between the source and target domains with squared MMD to measure the distance between each class. These techniques showed low performance in the presence of shifts in data distribution and did not generalize for other datasets. The main reason of the low performance of feature based methods is that they heavily rely on the custom choices of engineered features that differ from one covariate shift to another.

Deep Learning Domain Adaptation for Computer Vision

Recent hype in deep learning (DL) motivated researches to incorporate domain adaptation (DA) into DL to address the problem of covariate shits in data distribution. Most of the proposed DA with DA approaches in the literature are developed for computer vision applications [64] and those ideas are then utilized in other fields like Natural Language Processing (NLP) [65] and Activity recognition [16]. Generally, proposed DA with DL approaches could be grouped into hard parameter sharing architecture and soft parameter sharing architecture. In hard parameter sharing all hidden layers share the same parameters between the

classification task and the adaptation task. Whereas in soft parameter sharing each task has its own model with separate parameters and the distance between the parameters for corresponding layers is regularized to prevent them from divergence [66]. One of the early DA with DL approaches is Domain Adaptation Neural Network (DANN) [67] which is a hard parameter sharing architecture that uses adversarial learning to generate domain invariant features. DANN network performs two tasks simultaneously, in the first task it minimizes the source domain classification loss and in the second task it maximizes the loss of the domain classifier. In [68] authors proposed architecture similar to DANN but with squared Maximum Mean Discrepancy (MMD) [69] instead of domain classifier approach to minimize the domain divergence. In addition, they used the squared MMD loss at multiple layers instead of only the feature layer in [67]. The problem with models that depends on hard parameter sharing architecture is that they highly rely on the predefined shared architecture which limits models' ability to domain shared characteristics only. In order to facilitate learning both domain shared and specific characteristics researches in [28] proposed Domain Separation Network (DSN) approach that consists of four models. The two shared models learn domain shared components and they share the same parameters as in hard parameter architecture. On the other hand, the two private models learn domain specific components. In addition, shared auto-encoder was added to ensure generalizability of generated features. While effective, the core of this approach relies on hard parameter sharing architecture and it increases the number of parameters by a factor of four which limits the applicability on small architectures. In addition, DSN does not include modeling of the relation between the parameters which is prone to overfit source domain data. On the other hand authors in [70] proposed Soft Parameter Sharing Architecture with Linear modeling (SPSAL) to overcome the limitations of predefined shared structure. The model consisted of two streams, one for the source and one for the target domain and the relation between the parameters of the corresponding layers of the two streams are linearly modeled to prevent them from divergence. However, the main limitation of this approach is the assumption of linear relationship between the parameters of the two domains. In addition, it is very difficult for this approach to generalize on the target unlabeled data as it does not include robust representation learning.

Deep Learning domain adaptation for activity recognition

Although many works have been done in DA with DL for computer vision field very limited work has been found that addresses covariate shifts for time series applications. In [27] authors presented deep learning domain adaptation approach with hard parameter sharing to predict the activity of body parts with missing sensors by utilizing signal information from other sensors on body. They used unsupervised approach by combining "A-distance", which depends on the domain classifier error, and cosine similarity distance to select the most similar

source domain to the target domain. In [71] authors used DANN architecture and compared different distance measures for domain adaptation including squared MMD, Wasserstein Distance (WD) [72] and domain classifier approach. They found that squared MMD achieved best results for AR applications. In [73] authors proposed deep learning architecture composed of two cpies of the same model, one for source and one for target that shared the same parameters. The key difference of this work compared to the previous is that they used symmetric Kullback-Leibler Divergence (KL) on multiple layers to minimizing the divergence between the two domains. Their work considered two scenarios of domain adaptation, user diversity, where the model was tested on user activities that were not included in the model training process. The second scenario was device diversity, where the model was trained on data from smartphones and tested on data from smartwatch with the same activities. All aforementioned approaches for AR are based on hard parameter sharing architectures that suffer from the predefined shared architecture and hence restricts learning domain shared characteristics only.

3.2 Proposed Solution for Domain Adaptation

The objective of this work is to design a model that is able to autonomously make predictions in situations where

the environment changes over time causing shifts in data distribution.

3.2.1 Problem Formulation

The problem can be formulated as having a source domain with labeled data $D_s = \{x_n^s, y_n^s\}_{n=1}^N$ and a target domain with unlabeled data $D_t = \{x_m^t\}_{m=1}^M$. The goal is to predict target labels y_t for new domain. The model will adapt to target data with probability distribution that is different from the source $P(x^s) \neq P(x^t)$, but the output conditional probability distribution given the input is assumed to be the same for both source and target domains $P(y^s|x^s) = P(y^t|x^t)$, which means that the target activities for both domains are assumed to be the same. In this paper, we will consider the case of activity recognition (AR), where the domains consist of a particular user, a particular device, or both. The input to the model is multivariate time series data from wearable sensors like smartphone or smartwatch. The output is a particular user activity. The method should adapt to a new domain with a different user, a different device, or both.

3.2.2 Source Domain with Robust Learning

Source and target domains use a similar architecture that needs to support prediction for time series. The proposed method in this paper provides the step to

adapt any source network, but for best performance results, the source domain model architecture needs to be already effective at the target task. It has been shown in the literature that using a hybrid network that combines convolution layers (CNN) followed by a recurrent layer sequentially enhance automatically feature extraction from raw signals data and model temporal dynamics [39]. We propose to use a convolution layers (CNN) followed by a Gated Recurrent Unit (GRU) architecture (CNN-GRU), but also augmenting it with denoising auto-encoder (DAE). The proposed source model is shown in Figure. 3.1 where the encoder consists of two 1D convolution layers (CNN) followed by a Gated Recurrent Unit (GRU) layer. The Convolution Neural Network CNN acts as feature

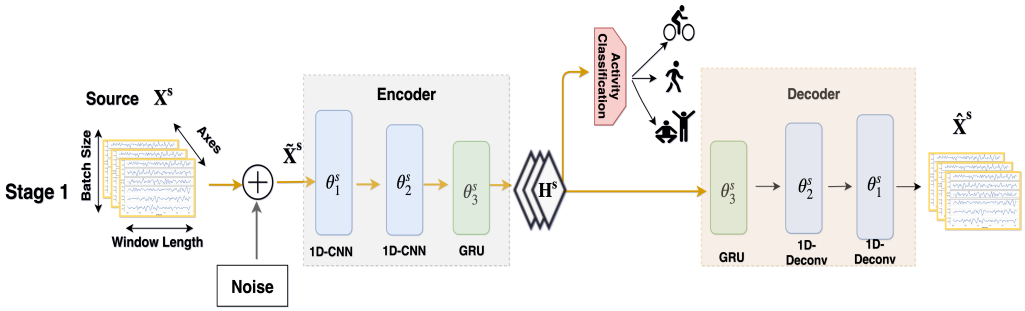


Figure 3.1: Source denoising auto-encoder. The encoder model consists of two 1D-CNN layers with max-pooling followed by a GRU layer. The decoder model consists of a GRU followed by 1D-transposed convolution with upsampling to reverse the operation of the encoder.

extractor, stacking several convolutional operators to create a hierarchy of more abstract features. However, the main difference in time series applications is that 1D convolution operation can be used to extract sequential information and capture temporal dynamics of time series. In addition, if the input is multivariate time series that composed of more than 1 channel, the 1D convolution performs depth-wise integration of signal channels. The depth-wise integration operation gives a weight to each axis, then each weight is learned during the training phase and hence result in the best integration of signals through axes (x, y, z). On the other hand, recurrent units are used to model the time dependencies of the sequence , hence give a model the ability to capture the context of the sequence. GRUs are special kind of recurrent units that have update and reset gates which allow them to decide how much of information to keep through time. This property enables the proposed deep architecture to model long term dependencies of the time series data. It has been shown by [74],[75] that in some tasks GRUs exhibit better performance than Long Short Term Memory (LSTM) on smaller datasets. It is also less complex and hence computationally less expensive. The decoder learns to reconstruct the input signal X^s and consists of a GRU followed by 1D-transposed convolution (deconvolution) layers [76] with upsampling to

reverse the operation of the encoder. The final reconstruction layer output is reconstructed multivariate time series of the same shape $\hat{X}^s$. The main motivation behind using DAE architecture is to address sparsity and improve generalization. The DAE compresses the data into lower dimensional which useful features representing the original signal and avoiding trivial solutions. The input to the model is multivariate time series data from wearable sensors like smartphone or smartwatch corrupted with Gaussian noise. The model learns two tasks: activity classification and reconstruction of the original data from compressed noisy signal as shown in Eq.3.1.

$$\mathcal{L}_{stage(1)} = \mathcal{L}_s(Y^s, \hat{Y}^s) + \mu\mathcal{L}_{rec\_s}(X^s, \hat{X}^s) \tag{3.1}$$

$\mathcal{L}_{stage(1)}$ is the total loss for source model model in the pre-adaptation stage, called stage (1), $\mathcal{L}_s$ is the cross entropy source classification loss given by equation (2)

$$\mathcal{L}_s(Y^s, \hat{Y}^s) = -\sum_{i=0}^{N_s} \mathrm{y}_i^s \cdot \log \hat{y}_i^s \tag{3.2}$$

$\mathcal{L}_{rec\_s}$ is the loss of reconstructing the signal X^s from the noisy signal $\tilde{X}^s$. μ is a weighting factor that trades off the two losses.

3.2.3 Robust Learning with DAE

In order to produce robust data representation and avoid trivial solutions we reconstruct the original input from noisy input after injecting Gaussian noise. The noisy signal $\tilde{X}$ is propagated through the encoder $E()$ to extract features $H = E(\tilde{X})$ which are then fed into decoder $D(E(\tilde{X}))$ that tries to reconstruct the input $\hat{X} = D(H)$. The reconstruction loss $\mathcal{L}_{rec}$ used to minimize the difference between the reconstructed input $\hat{X}$ and the original input X is the scale-invariant mean squared loss incorporated from work of [28].

$$\mathcal{L}_{rec\_s}(\mathrm{X}, \hat{\mathrm{X}}) = \frac{1}{k}\|\mathrm{X} - \hat{\mathrm{X}}\|_2^2 - \frac{1}{k^2}\left([\mathrm{X} - \hat{\mathrm{X}}] \cdot 1_k\right)^2 \tag{3.3}$$

Where k is the number of samples in the time series input, 1^k is a vector of ones of length k and $\|.\|$ is the L_2-norm. The root mean squared loss is traditionally used for reconstruction tasks which penalizes the predictions that are correct up to a scaling term, however it has been shown in [77] that the scale-invariant mean squared error results in better samples reconstruction because it penalizes differences between pairs of element-wise samples irrespective of absolute global scale.

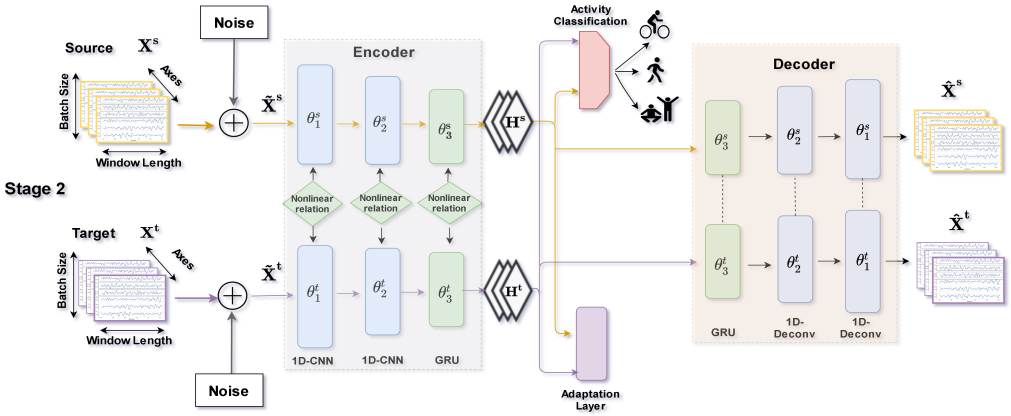


Figure 3.2: The proposed DASH approach initialized with the pretrained parameters obtained from stage (1) and adaptation objective. The encoder consists of two streams one for the source domain and one for the target domain. Each stream consists of the base CNN-GRU model illustrated in Figure 3.1. The parameters of the two streams of the encoder are related through a nonlinear transformation. The shared decoder learns to reconstruct the original data X from input corrupted with Gaussian noise $\tilde{X}$.

3.2.4 Soft Parameter Sharing with Robust Learning

Our choice of soft parameter sharing architecture is motivated by the MTL literature [78, 79] where we expect that soft parameters sharing will give more flexibility to learn domain specific as well as shared representations in domain adaptation configuration compared to hard parameter sharing architecture. The encoder for DA, shown in Figure 3.2, consists of two identical streams one for the source domain and one for the target domain. The parameters of the source and target models in the encoder are related through a nonlinear transformation. The choice of non-linear relationship is explained in Sec. 3.2.5. The decoder parameters are shared between the source and target domains since we want to extract similar features from source and target domains. Eq. 3.4 illustrates total learning loss during stage (2). Our proposed approach consists of first learning the source domain model as described in the Sec. 3.2.2. In the second stage, called stage (2) shown in Figure 3.2, we initialize both encoder and decoder with the pre-trained parameters obtained from stage (1) and retrain the model on both source labeled data and target unlabeled data with domain adaptation layer to minimize the divergence between the two domains. Eq. 3.4 illustrates total learning loss during stage 2.

$$\mathcal{L}_{stage2} = \mathcal{L}_s + \lambda_1 \mathcal{L}_{rec-total} + \lambda_2 \mathcal{L}_{da} + \lambda_3 \mathcal{L}_{tr} \tag{3.4}$$

.

Where $\mathcal{L}_{da}$ is the domain adaptation loss described in Sec. 3.2.6 and $\mathcal{L}_{tr}$ is

the nonlinear transformation loss between the parameters of each layer of source and target models described in Sec. 3.2.5. The reconstruction loss $\mathcal{L}_{rec-total}$ is applied for both domains as illustrated in Eq. 3.5. The weights λ_1,λ_2, λ_3 controls the interaction of each loss.

$$\mathcal{L}_{rec-total} = \mathcal{L}_{rec_s}(X^s, \hat{X}^s) + \mathcal{L}_{rec_t}(X^t, \hat{X}^t) \tag{3.5}$$

3.2.5 Modeling Covariate Shift Between the Domains

We assume that the learned parameters of the two models will be related, and the shift can be assessed by L_2 distance between parameters of the two domain models as shown in Eq.(3.6).

$$d(\theta_l^s, \theta_l^t) = \|g(\theta_l^s) - \theta_l^t\|_2 \tag{3.6}$$

The $d(\theta_l^s, \theta_l^t)$ represents distance between source model parameter θ_l^s and target model parameter θ_l^t of layer l. The $g(\theta_l^s)$ is the transformation from source model parameters θ_l^s to target model parameters θ_l^t. In the work of [70] the relation between the source and target model parameters was modeled as a linear relation in Eq. (3.7).

$$\theta_l^t = a_l\theta_l^s + b_l \tag{3.7}$$

Their approach was developed to model covariate shifts for image classification applications and has not been tested on sequential data that involves temporal changes with time.

In our work, we overcome the limitation of linear assumption and model the relation between source model and target model parameters as a non linear transformation.

$$\theta_l^t = \theta_l^s + \eta r(\theta_l^s) \tag{3.8}$$

Further justification anecdotal derivation is provided in the Appendix to provide support for the non-linear relation and a specific choice of function r(). We chose $tanh()$ function to model the nonlinear relation between the parameters.

$$\theta_t = \theta_s + tanh(a\theta_s + b) \tag{3.9}$$

The final expression of the objective that we propose to model the relation between the source and target domain parameters is shown in Eq.3.10. The parameters $'a'$ and $'b'$ are trainable weights to learn the nonlinear relation between source and target model parameters.

$$\mathcal{L}_{tr} = \|\theta_t - \theta_s - (tanh(a\theta_s + b))\|_2 \tag{3.10}$$

3.2.6 Domain Discrepancy Loss Function

In order to develop a model that generalizes well from one domain to another and adapts to covariate shifts in non stationary environment it is essential to reduce the discrepancy in the representation encoding layer that summarizes the information of source and target domains. As a result the success of the domain adaptation mainly relies on finding the invariant representations for both domains. We propose to use squared Maximum Mean Discrepancy (MMD) which represents distances between mean of the distributions of source and target encodings. Given n samples of the source features H^s and m samples of the target features H^t the squared MMD can be expressed mathematically as shown in equation (3.11).

$$\mathcal{L}_{da} = \mathcal{L}_{mmd}(H^s, H^t) = \left\| \sum_{i=1}^{n} \frac{\phi\left(\mathrm{h}_i^s\right)}{n} - \sum_{j=1}^{m} \frac{\phi\left(\mathrm{h}_j^t\right)}{m} \right\|_{\mathcal{H}}^2 \tag{3.11}$$

Where $\phi()$ represents mapping to Reproducing Kernel Hilbert Space (RKHS). Usually, the mapping $\phi()$ is an unknown non linear mapping and a kernel like Radial Basis Function (RBF) is used $K(p, q) = \exp\left(-\|p - q\|^2/\sigma\right)$. After expanding equation (3.11) each inner product that involves multiplication of $\phi(h^s)\phi(h^t)$ is replaced with the kernel $K(h^s, h^t)$, and the final expression is shown in (3.12).

$$\sum_{i,i} \frac{K\left(\mathbf{h}_i^s, \mathbf{h}_i^s\right)}{(n)^2} - 2\sum_{i,j} \frac{K\left(\mathbf{h}_i^s, \mathbf{h}_j^t\right)}{nm} + \sum_{j,j} \frac{K\left(\mathbf{h}_j^t, \mathbf{h}_j^t\right)}{(m)^2} \tag{3.12}$$

The characterization of this distance as the maximum mean discrepancy refers to the fact that computing the squared MMD is equivalent to finding the RKHS function that maximizes the difference in expectations between the two features probability distributions.

3.3 Experimental setup

The proposed approach is evaluated with state-of-the-art approaches using weighted F1 score. All models are implemented using Google's deep learning Tensor-Flow library. We ran our experiments on a PC equipped with an NVIDIA GTX 1080 GPU, and an Intel Core i7-7700 (3.60 GHz) CPU and 32 GB of RAM.

3.3.1 Datasets

To evaluate the applicability of our proposed DASH approach in real world context, we chose two benchmark datasets collected in the wild: Position Activity Recognition (PAR) [80] and Heterogeneity Activity Recognition (HAR) [21]. The Position Activity Recognition (PAR) dataset was recorded using seven wearable

commercial sensors (6 smart-phones and one smart-watch) from 15 participants performing eight activities (running, walking, lying, sitting, standing, stairs up, stairs down, and jumping). The devices were synchronized and the data was recorded at a sampling rate of 50 Hz. This dataset was collected in the wild to resemble their everyday life usage. On the other hand, HAR dataset was collected from 9 participants performing 6 common activities recorded (Biking, Sitting, Standing, Walking, Stair Up and Stair down) using 8 smartphones and 4 smartwatches from different manufactures. HAR dataset is also gathered in real world scenarios, to reflect sensing heterogeneities expected in real deployments. The smartphones used in HAR dataset and their sampling rate are summarized in Table 3.1.

Table 3.1: HAR dataset smartphone devices with their corresponding sampling rate (SR).

smartphone	Nexus4	Samsung S3	Samsung S+	S3 mini
SR (Hz)	200	50	150	100

3.3.2 Data Pre-processing

To compare our results with those obtained in [73], we followed the same data preparation steps. The accelerometer data was segmented through sliding window of length 128-sample (2.5 sec) and 50% overlap between the successive frames. The sampling rate was down sampled of each device in HAR dataset to 50 Hz. Linear interpolation was selected to mitigate the frequency heterogeneity as was suggested by [80]. Finally, the data from both datasets was normalized using standard scalar to ensure zero mean and unit variance across the three accelerometer axes.

3.3.3 Prior State-of-the-art

We reproduced the most related state-of-the-art DA approaches (DANN [67], DSN [28], SPSAL [70]) to examine their effectiveness on time series. In addition, we compare our approach to the state-of-the-art work in AR [73] that we found most related to the problem we are addressing. In order to transfer the state-of-the-art models from images data to time series data we replaced 2D-CNN layers with 1D-CNN layers for representation learning while maintaining the same general architecture of each model. For fair comparison, we used the source domain encoder CNN-GRU model architecture described in Sec.3.2.2 as feature extractor for all CV-based DA models. All models were trained using Adam Optimizer with learning rate of 0.001. The best value for reconstruction loss weight (μ) Eq. 3.1 was chosen 0.01, and the best values for (λ_1, λ_2, λ_3) Eq. 3.4

were chosen (0.001, 0.1, 0.5) respectively. The details of layers hyperparameters obtained for the proposed DASH approach are shown in Table 3.2. All hyper parameters were obtained using grid search. Note that only the source domain data and was used for hyperparameter tuning.

Table 3.2: Values of hyperparameters obtained for DASH approach from grid search.

Layers	**Hyperparameters**
2*Input	batch size: 128 Gaussian noise std: 0.2
2*1D-CNN	1st layer: 32 kernels of size 5 2nd layer: 64 kernels of size 3
GRU	64
2*1D-transposed convolution	1st layer: 64 kernels of size 3 2nd layer: 32 kernels of size 5

3.3.4 Proxy A-Distance

To quantify the distance between the source and target domain distributions, it is common practice to use a metric called the A-distance introduced by [81]. Given 2 distributions D_s and D_t over A, the A-distance is defined as $d_A(D_s, D_t) = 2sup_{A \in \mathcal{A}} |Pr_{D_s}[A] - Pr_{D_t}[A]|$. This metric can be approximated by training a linear SVM to discriminate between 2 domains: the error of this linear SVM is called the generalization error η. Then the proxy A-distance (PAD) can be calculated as $\hat{d}_{\mathcal{A}} = 2(1 - 2\eta)$.

3.3.5 Domain adaptation scenarios

To examine the effectiveness of the proposed DASH approach we describe below the three different real world situations: cross user domain adaptation, cross device domain adaptation and cross user cross device domain adaptation. The input in all three scenarios is 3-axes accelerometer data. All models are trained on source domain labeled data and target domain unlabeled data. The unlabeled target data is divided into 5 equal parts (5-folds), where at each fold 4 parts are used for training in unsupervised configuration and the fifth is held for testing. This approach is used to ensure that the model is not biased to a particular part of the target domain data and hence avoid overfitting. The final F1 score is obtained by averaging the results over the number of folds. The significance in the difference between the performance of models was statistically tested using modified t-test introduced by [82].

Cross User Domain Adaptation:

This is a situation where the source domain model was developed for a particular set of users and the DA goal is to adapt the model to a new user. We picked a group of users as source domain {5, 10, 3, 12, 13, 14, 15} from PAR dataset and {a, b, c, d} from HAR dataset. As for target domain we chose {1, 4, 7, 8, 9, 11} group from PAR dataset and {e, g, i, h} from HAR dataset. In this scenario the source model is developed on source users and the model is adapted to one user from target group at a time. In addition, cross user evaluation was performed on two different devices: a smartwatch and smartphone. The smartphones were fixed on participants waist position for both PAR and HAR datasets. As for the smartphones, the data used was collected from Samsung Galaxy S4 and Samsung S+ for PAR and HAR datasets respectively. In addition, LG smartwatch was chosen from PAR dataset and Samsung smartwatch from HAR dataset.

Cross Device Domain Adaptation:

This is a situation where the source domain model was developed on one smartphone and the DA goal is to adapt the model to a different smartphone for the same participant. The HAR dataset includes samples for each participant carrying different kind smartphones but at the same body position and orientation (carried in a tight pouch around waist). As a result, HAR provided a suitable dataset for cross device DA evaluation. For this case, accelerometer data was collected from different smartphones carried by users around their waists. A comprehensive comparison was performed for cases where one device was used at a time as a source domain and the rest of the phones were used as the target domain. Same user {a} used for both source and target domains.

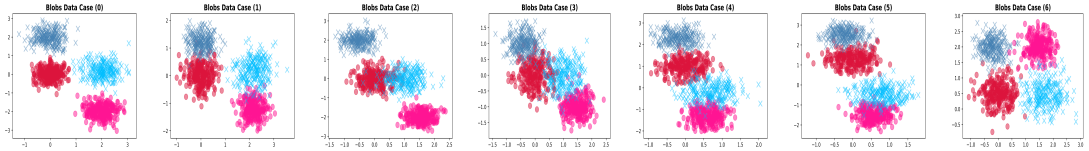
Cross User Cross Device Domain Adaptation:

The goal of this experiment was to examine the effectiveness of our proposed approach in a more common and more challenging real world scenario where we have two sources of covariate shifts: a new user and a new device. In this scenario the model is trained on labeled data from one smartphone for a set of users and tested on unlabeled data from another different smartphone for a new user. Using the HAR dataset, group of users {a, b, c, d} with {Samsung S+} were selected as source domain and {e, h, g} with {Nexus, S3, S3mini} were picked as target domain.

3.4 Results & Discussion

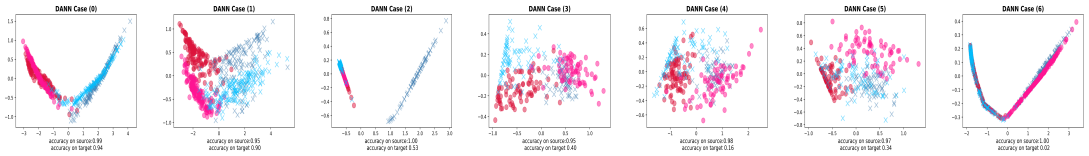
This part starts with providing guidelines for hard parameter sharing and its limitations with toy data. After that the relationship between source and tar-

get parameters of our proposed DASH model is evaluated. Next our proposed model is evaluated on three DA situations in AR applications and the performance is compared to state-of-the-art DA models. Finally, the effect of each part of the proposed DASH model is examined visually with t-distributed stochastic neighbor embedding (t-SNE) [53] and quantitatively using A-distance. Unlike the existing work in field of AR, we propose a soft parameter sharing domain adaptation DASH approach with flexibility to learn domain specific and shared characteristics. This is achieved by modeling the relation between the source and target parameters as a nonlinear transformation. The proposed DASH architecture is designed for multivariate time series data taking into considerations sequence temporal dynamics and long term dependencies.



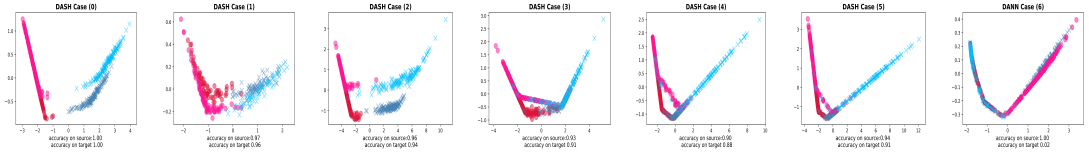
(a) (b) (c) (d) (e) (f) (g)

Figure 3.3: Different situations of shifts in data distribution using blobs toy data.



(a) (b) (c) (d) (e) (f) (g)

Figure 3.4: Feature maps after Domain adaptation using DANN approach. Once the two datasets start to have outer overlap Figure 3.3c the DANN approach results in aligning of dissimilar classes from the two domains



(a) (b) (c) (d) (e) (f) (g)

Figure 3.5: Feature maps after Domain adaptation using DASH approach. DASH approach adapts successfully to the target domain in all cases Figure 3.5a-3.5f

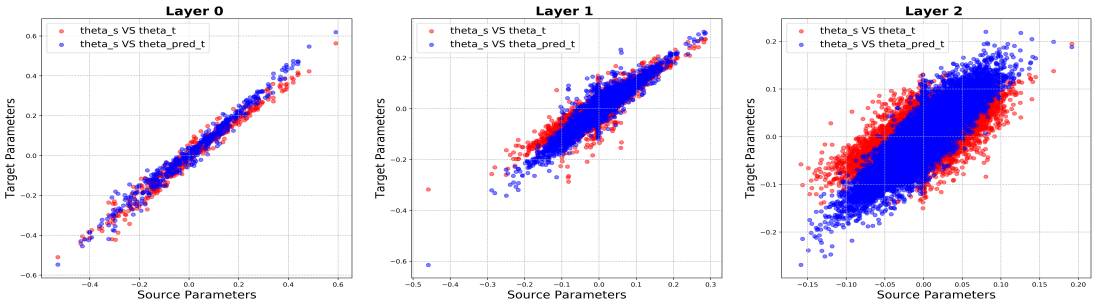
3.4.1 Evaluation of DASH vs DANN under various shift conditions

Recently, DA methods have shifted from simpler learning based on hard parameter sharing approaches to more complex soft parameter sharing approaches. However, there was no study found that shows when complex learning is needed. In this subsection we examine the limitations of state-of-the-art hard parameter sharing DANN approach and show how our proposed DASH model based on soft parameter sharing overcomes these limitations. We assume data from two domains with two classes in each domain. We hypothesize that certain shifts in data cause failure in hard parameter sharing. To demonstrate its success and limitations, we create seven scenarios of data shifts, including: 1) Shift with same orientation and no overlap, Figure 3.3a 2) Shift with same orientation and overlap between the classes within each domain (inner overlap), Figure 3.3b 3) Shift with same orientation and overlap between dissimilar classes across the domains (outer overlap) Figure 3.3c 4) Shift with same orientation and combination of inner and outer overlap, Figure 3.3d 5) Figure 3.3e, 3.3f are extreme cases of Figure 3.3d 6) Shift with flipped orientation, Figure 3.3g. The source domain and target domain data consist of 1000 samples generated from Gaussian distribution with standard deviation of [0.35,0.3] and [0.3,0.4] respectively. It can be seen from Figure 3.4a and Figure 3.4b that DANN approach works perfectly when the shift is with same orientation and there is no outer overlap. However once the two domains have outer overlap Figure 3.4c the 'x' class from target and 'o' from source become much closer compared to the distance between 'x' from target and 'x' from source the DANN approach maps the closes classes to similar representation in feature space and hence results in aligning of dissimilar classes. As a result, the accuracy on the target domain is significantly dropped to 53%. The performance drops to 40% as we combine both inner overlap to outer overlap Figure 3.4d. The extreme case of Figure 3.4d is Figure 3.4e where the performance drops even further to 16%. These experiments show the major limitation of DA models based on hard parameter sharing architectures: as the dissimilar categories of the two domains become much closer than the corresponding similar categories, hard parameter sharing architectures fail in adapting the similar categories of the two domains. This limitation is due to the fact that hard parameter sharing architectures are able only to learn domain shared representation and do not capture each domain specific characteristics and hence map the nearest classes of the two domains to the same representation regardless of their corresponding categories. On the other hand our proposed DASH approach utilizes soft parameter sharing architecture to learn domain specific and shared information to overcome the limitations of hard parameter sharing. It can be noticed that DASH approach adapts successfully to the target domain in all aforementioned cases Figure 3.5a-3.5f with minimum performance of 88% on the target domain. Finally, in the situation with flipped orientation Fig 3.3g both

DANN and our proposed DASH approach fail. We leave solving this limitation as a potential direction for future research.

3.4.2 Evaluation of Relationship Between Source and Target Parameters

To determine the true relationship between source and target model parameters, we separately train source and target domain models of DASH with labeled data from their respective domains with nonlinear modeling disabled. Using PAR dataset, a group of users is chosen for source domain {5, 10, 3, 12, 13, 14, 15} and user {6} is chosen along with the activity labels for target domain. The idea is to derive the parameters for the target domain with an ideal situation when labels are indeed available. The resulting relationship between source and target for the ideal case was then compared to the proposed non-linear relationship for DA case when target labels are not available. It can be seen from Figure 3.6a that the relationship between the first layer parameters is almost linear , which means that the parameters for both domains are almost similar. This makes



(a) (b) (c)

Figure 3.6: Visualization of the relation between the parameters of the source and target domains for each layer and the estimated parameters using nonlinear modeling with tanh() function. (a) Source weights vs target weights of the first layer (b) Source weights vs target weights of the second layer (c) Source weights vs target weights of the third layer.

sense since the source and target domain data are related and because the first layers learn generic features from the data that are similar across domains. As the layers become deeper shown in Figure (3.6b, 3.6c), the relation starts to become more nonlinear. This analysis confirms that the relationship between parameters seems to have a linear component and non-linearities. These results support the proposed nonlinear relation in Eq.(3.8) from Sec.3.2.5. Finally, Figure3.6 shows a comparison between the proposed $tanh()$ non-linearity in Eq.(3.10) and the relationship derived from the ideal case. It can be seen that the proposed non-linearity covers a wide range of the relationship even in the early layers when the

relationship is linear. While the $tanh()$ non-linearity is not perfect, it does seem to provide much stronger support than a linear relation.

3.4.3 Comparison to state-of-the-art

The results of all models on the three domain adaptation scenarios (cross user, cross device and cross user cross device) are summarized in Tables (3.3,3.4,3.5) respectively. The F1 score in each scenario was obtained by averaging the results over the 5-folds.

Table 3.3: Cross user domain adaptation results of our proposed DASH model compared to state-of-the-art approaches on the raw input of the PAR dataset. The results were obtained by averaging F1 score over the 5-folds. It can be noticed that the proposed DASH model outperforms all the other state-of-the-art approaches.

	DA Models				
Dataset	DANN	HDCNN	SPSAL	DSN	DASH
HAR(W)	53	57.2	80.3	59	**83.5**
PAR(W)	56.3	53	74	58.6	**82**
HAR(P)	71.7	58	82.3	69	**86.6**
PAR(P)	58	54.7	81.7	52.2	**85.3**

In the cross user scenario, the improvement in averaged F1 score of DASH is up to 8% on PAR and 4.3% on HAR datasets. In addition, it is noticeable that the results on HAR dataset are higher than PAR dataset because PAR datset was collected in more realistic scenarios. It also can be seen that the results of the smartphone are better than results of the smartwatches for both datasets. We think that this could be due to the fact that the smartphones used in the experiment are carried in a more generic body position (waist, pants pocket), which helps capturing more generic body dynamics during the activities. In the cross device scenario, and cross user cross device scenario the DASH model outperforms the highest F1-score achieved by SPSAL by 5.7% and 7.2% on average respectively.

Overall it can be seen that the averaged results of our proposed DASH model outperforms all the other state-of-the-art approaches in the three domain adaptation scenarios. In addition, we can clearly see that the results of cross user cross device are the lowest, because it includes two source of shifts in data distribution, new user and new device. We performed modified t-test [82] to illustrate that the results obtained by DASH are statistically significant. We compared the results of DASH with the highest state-of-the-art results obtained by SPSAL. We specified the significance level to be 5%. After applying this test method on the 5-folds results of all approaches we found that the highest p-value was

Table 3.4: Cross device domain adaptation results of our proposed DASH model compared to state-of-the-art approaches on the raw input of HAR dataset. The results were obtained by averaging F1 score over the 5-folds. It can be noticed that the proposed DASH model outperforms all the other state-of-the-art approaches.

Source	Target	Source only	DANN	HDCNN	SPSAL	DSN	DASH
Nexus	**S3mini**	47	70	67	75	40	80
	S3	37	52	48	71	62	73
	SamsungS+	34	71	52	67	44	82
S3	**Nexus**	33	37	56	71	62	88
	S3mini	59	80	40	83	46	85
	SamsungS+	30	35	45	56	37	57
S3mini	**Nexus**	32	70	54	72	71	80
	S3	49	76	52	84	36	89
	SamsungS+	36	42	38	57	38	57
SamsungS+	**Nexus**	42	75	46	83	62	87
	S3mini	46	84	48	80	72	86
	S3	47	78	50	81	73	84
AVG		41	64.16	49.60	73.3	53.75	**79**

Table 3.5: Cross user cross device domain adaptation results of our proposed DASH model compared to state-of-the-art approaches on the raw input of HAR dataset. The results were obtained by averaging F1 score over the 5-folds.

Source	Target	Source only	DANN	HDCNN	SPSAL	DSN	DASH
SamsungS+	**Nexus (e)**	32	60	40	70	55	77
	S3 (e)	30	48	36	71	46	70
	S3mini (e)	27	62	30	74	52	83
SamsungS+	**Nexus (h)**	32	56	40	72	57	80
	S3 (h)	28	60	34	61	50	73
	S3mini (h)	25	48	36	67	51	71
SamsungS+	**Nexus (g)**	30	55	40	58	48	62
	S3 (g)	29	59	37	65	45	74
	S3mini (g)	24	56	32	72	51	85
AVG		28.9	56	36	67.8	50.5	**75**

1.2% shown in Table 3.6, which is lower than the specified significance level 5%. Hence we reject the null hypothesis, and conclude that there is sufficient evidence that the results of DASH approach are significantly better compared to other state-of-the-art approaches.

Table 3.6: T-test showing the statistical significance of DASH superiority to SPSAL in all three domain adaptation scenarios.

	Cross user	Cross device	Cross user cross device
p-value	**0.2%**	**1.2%**	**0.8%**

DASH Ablation Analysis

In this section we examine the effect of each part of our proposed DASH approach and their contribution to the final model performance. In this experiment we used the results obtained from cross user scenario, PAR dataset as it showed the clearest variation among each part of DASH approach. The results are illustrated in Fig 3.7 using box plot.

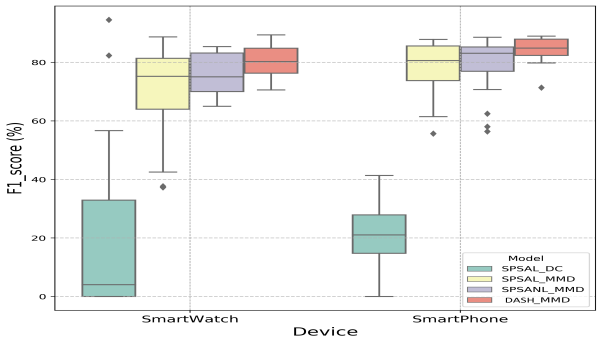
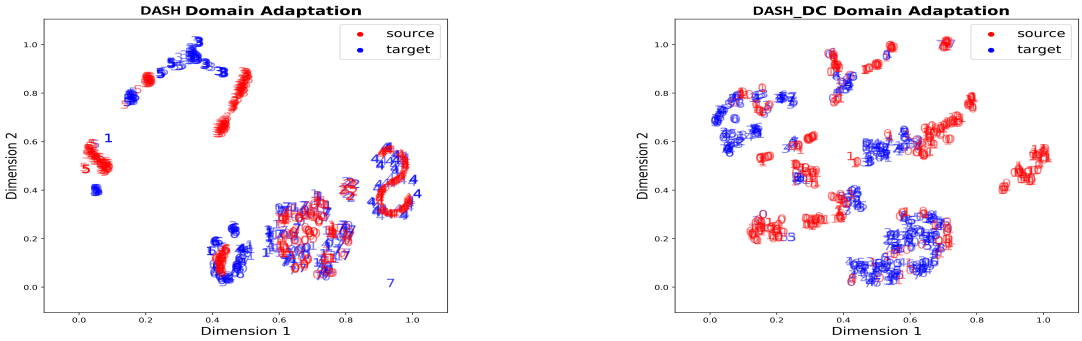


Figure 3.7: Box plots of F1 performance on the target domain in cross user scenario after adding each part of the proposed DASH approach. It can be seen that each part of the proposed DASH approach contributes significantly in improving the accuracy and reducing the variation of the results.

It can be seen that the main effect of non-linear modeling (SPSANL_MMD) is an increase by 3% and 6% on average compared to SPSAL_MMD for smartwatch and smartphone data respectively. However, both SPSAL and SPSANL with squared MMD have high variation in their results. It can be noted that adding auto-encoder (DASH_MMD) achived 5% and 3% higher result compared to SPSANL_MMD with significantly lower variation where the standard deviation was reduced by 2x for smartwatch and 3.5x for smartphone data. As a result the main advantage of the using DAE is to enforce the model to generate robust representations of the data, and hence result in more consistent results and improve model generalization. It also can be noted that using DC approach to minimize the domain divergence resulted in the worse performance and similar results were observed in the other domain adaptation scenarios. Furthermore, we visualize the representation embeddings of both DASH_MMD and DASH_DC using t-distributed stochastic neighbor embedding (t-SNE) projection [53] as shown

in Fig 3.8. Each activity class is represented by a different number while color



(a) (b)

Figure 3.8: T-SNE visualization of representation learned after domain adaptation using DASH with squared MMD and DC losses in cross user scenario, PAR dataset. Each number on the plot represents activity. a) The embeddings of proposed DASH_MMD representation . b) The embeddings of DASH_DC representation

coding the domains. It can be seen that representations of the two domains classes produced by DASH_DC are aligned to the wrong classes of the two domains, where corresponding classes from the source domain are overlapped to different classes from the target domains. On the other hand, the classes of each activity of the DASH_MMD approach are clearly separated and the similar classes of the source and target domains are mostly aligned. This is due to the fact that the squared MMD measures the maximum distance between the expectations of the two domain distributions, whereas the domain classifier generate domain invariant features by maximizing the loss on the domain classification task.

Evaluation of Domain Adaptation

The main goal of domain adaptation is to minimize the divergence between the two domains for the corresponding target classes. In this part we quantify the similarity between the two domains using the proxy A-distance (PAD) adopted [81] shown in Figure 3.9. We trained denoising auto-encoder (DAE), SPSAL and DASH on PAR dataset in cross user scenario for both smartphone and smartwatch and computed the PAD between source and target domains features representation. The DAE representation was obtained after training on both source and target unlabeled data. We note that both SPSAL and DASH contribute to reducing PAD distance between source and target domains. However, we can clearly see that the PADs of DASH are much lower than SPSAL pushing downwards.

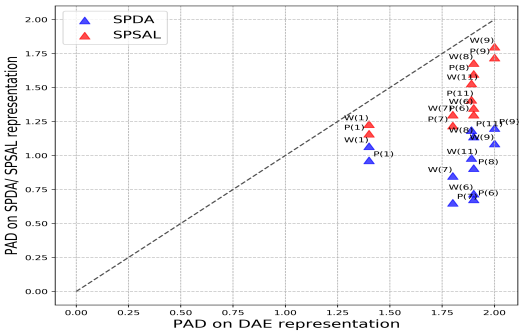


Figure 3.9: Proxy A-distance computed for the 3 representations: denoising auto-encoder, SPSAL and DASH representations of the data.

3.4.4 Effect of data size on DASH performance

To examine the effect of the size of the data we picked cross user scenario, smartphone data, PAR dataset with same configuration described in Sec.3.4.3. We chose participant {11} as target as his dataset contains the largest number of observations (6788). We vary the size of the training data of both source and target data from 500 observations to 5000 in step of 500. For each data size we train both DASH and SPSAL. We held out 1500 observations, which we kept same for all training size, to test the performance models performance. The performance of both DASH and SPSAL for each data size is illustrated in Figure3.10. It is

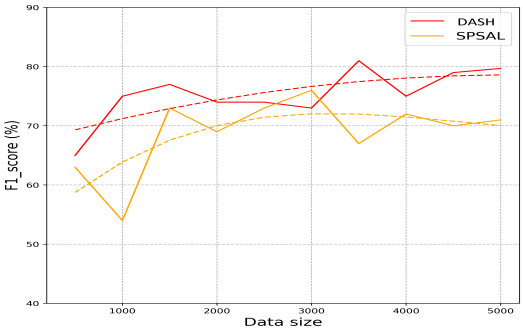


Figure 3.10: F1 performance of DASH and SPSAL on the target domain versus training datasize.

noticeable that the performance of DASH increases almost linearly with datasize reaching F1 score up to 80%. This is expected as the auto-encoder base model requires more data to produce better representation for both source and target domains. On the other hand, SPSAL F1 score increases rapidly at the beginning with higher variation, however, the performance saturates at around 70% and do not increase with the increase of the data. We can also observe that DASH outperformed SPSAL at all dataset sizes.

Chapter 4

Conclusion

In this work, we examined the robust learning to environmental noise with adversarial examples for seizure prediction. The proposed method aims to address three main limitations in seizure prediction literature: (1) the high variations in the signals between pre-ictal and normal activity of different patients and within the same patient data, (2) prediction accuracies still do not meet the required criteria in order to avoid life-threatening scenarios for impacted patients, and (3) limited amount of labeled training data per patient for pre-ictal state samples. The proposed method demonstrates significant improvements in comparison to prior state-of-the-art [1] on two benchmark datasets, Boston Children's Hospital (CHB-MIT) and Freiburg Hospital (FB), in terms of sensitivity, false positive rate per hour (FPR/h) and area under the curve (AUC). We show an improvement of 3x in model robustness across patients and across datasets as measured in AUC variations. Moreover, we show an increase of up to 6.7% in average AUC on both datasets.

In addition, we presented a novel domain adaptation architecture, called DASH, for multivariate time series data that is based on soft parameter sharing architecture. The proposed DASH method aims to improve DA by learning domain specific and shared characteristics of source and target domains. We studied the limitations of previous approaches in domain adaptation that are based on hard parameter sharing architecture and showed how our proposed soft parameter sharing approach overcome these limitations. Furthermore, we showed that squared MMD adaptation loss gives better results compared to *DC* loss for time series data. The effectiveness of DASH approach was examined under comprehensive set of experiments on two benchmark AR datasets collected in the wild, Heterogeneity Activity Recognition (HAR) and Position Activity Recognition (PAR). The proposed method showed significant improvements in three DA scenarios compared to previous state-of-the-art approaches [29, 73]. We showed an increase of up to 8% in the weighted F1 score on average on both benchmark datasets with improvements of 3.5x on average in model robustness.

Future work, includes Examining DASH approach to other domain adaptation

problems for time series data like speech recognition and epilepsy where we need robustness to both noise in the signal and robustness to shifts in data distribution.

Appendix A

Appendix

A.1 Derivation of Non-linear modeling

To illustrate the intuition behind our approach we first assume a simple system of only one parameter and a non linear activation "S" for both source and target domains as shown in Fig.A.1. The θ_k is the trainable weight of the neuron at iteration 'k'. The inputs to the source and target neurons are X^s and X^t where $X^t = f(X^s)$ and $f()$ any linear or nonlinear function. We initialize the weights of the two neurons with the same values $\theta_0^s = \theta_0^t$. The update equations of the

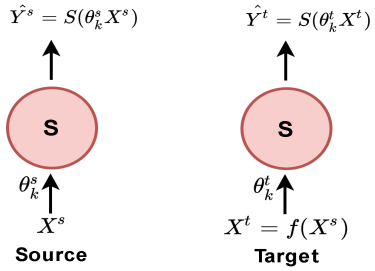


Figure A.1: Simplified system for both source and target domains of one neuron with a nonlinear activation 'S'.

two parameters (θ_k^s ,θ_k^t) during the training are as follows:

$$\theta_{k+1}^s = \theta_k^s - \eta \nabla_{\boldsymbol{\theta_k^s}} J(\hat{Y}_1, Y_1) \tag{A.1}$$

$$\theta_{k+1}^t = \theta_k^t - \eta \nabla_{\boldsymbol{\theta_k^t}} J(\hat{Y}_2, Y_2) \tag{A.2}$$

The relationship between the updated weights θ_{k+1}^s and θ_{k+1}^t can be expressed as follows:

$$\theta_{k+1}^s - \theta_{k+1}^t = \theta_k^s - \theta_k^t + \eta(\nabla_{\boldsymbol{\theta_k^t}} J(\hat{Y}^s, Y^s) - \nabla_{\boldsymbol{\theta_k^s}} J(\hat{Y}^t, Y^t)) \tag{A.3}$$

And hence the relation between the parameters at each iteration 'k' becomes as follows:

$$\theta_k^t = \theta_k^s + \eta(g(\theta_k^s, \theta_k^t)) \tag{A.4}$$

Where $g()$ is a non-linear function in θ_s, θ_t and involves gradients of nonlinear activations. Since the two domains are related, and since we initialized the parameters to be equal, we expect that θ_k^s to be related to θ_k^t, hence $g(\theta_s, \theta_t)$ could be written as $g(\theta_s, F(\theta_t))$ which could be written in terms of θ_s only $r(\theta_s)$, where $F()$ could be any linear or nonlinear mapping function and $r()$ is a non linear function. Now Eq.A.5 becomes as follows :

$$\theta_k^t = \theta_k^s + \eta r(\theta_k^s) \tag{A.5}$$

For deep neural network the exact expression of $r()$ as a function of θ_s is very complicated. However the function $r()$ can be approximated in general using a nonlinear function $\sigma()$ with learnable weights. Since at the convergence, the gradients become very small and since η is usually chosen to much less than 1, $\eta r(\theta_s)$ is expected to be strictly in the range (-1, 1). We chose a non-linear function $tanh()$ because it is in the range of (-1,1) and also easy to update during the back-propagation. The final expression that we propose to model the relation between the parameters of source and target domains is shown in Eq.A.6. The parameters $'a'$ and $'b'$ are trainable weights to learn the nonlinear relation between source model parameters and target model parameters.

$$\theta_t = \theta_s + (tanh(a\theta_s + b)) \tag{A.6}$$

Bibliography

[1] N. D. Truong, A. D. Nguyen, L. Kuhlmann, M. R. Bonyadi, J. Yang, S. Ippolito, and O. Kavehei, "Convolutional neural networks for seizure prediction using intracranial and scalp electroencephalogram," *Neural Networks*, vol. 105, pp. 104–111, 2018.

[2] J. Dahmen, A. La Fleur, G. Sprint, D. Cook, and D. L. Weeks, "Using wrist-worn sensors to measure and compare physical activity changes for patients undergoing rehabilitation," pp. 667–672, 2017.

[3] J. Wen, J. Indulska, and M. Zhong, "Adaptive activity learning with dynamically available context," pp. 1–11, 2016.

[4] M. Swan, "Sensor mania! the internet of things, wearable computing, objective metrics, and the quantified self 2.0," *Journal of Sensor and Actuator networks*, vol. 1, no. 3, pp. 217–253, 2012.

[5] A. R. Dargazany, P. Stegagno, and K. Mankodiya, "Wearabledl: Wearable internet-of-things and deep learning for big data analytics—concept, literature, and future," *Mobile Information Systems*, vol. 2018, 2018.

[6] S. Liu, "Connected wearable devices worldwide 2016-2022," 2019.

[7] P. Mirowski, D. Madhavan, Y. LeCun, and R. Kuzniecky, "Classification of patterns of eeg synchronization for seizure prediction," *Clinical neurophysiology*, vol. 120, no. 11, pp. 1927–1940, 2009.

[8] L. D. Iasemidis, "Seizure prediction and its applications," *Neurosurgery Clinics*, vol. 22, no. 4, pp. 489–506, 2011.

[9] S. M. Usman, M. Usman, and S. Fong, "Epileptic seizures prediction using machine learning methods," *Computational and mathematical methods in medicine*, vol. 2017, 2017.

[10] A. H. Shoeb and J. V. Guttag, "Application of machine learning to epileptic seizure detection," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 975–982, 2010.

[11] J. Rasekhi, M. R. K. Mollaei, M. Bandarabadi, C. A. Teixeira, and A. Dourado, "Preprocessing effects of 22 linear univariate features on the performance of seizure prediction methods," *Journal of neuroscience methods*, vol. 217, no. 1-2, pp. 9–16, 2013.

[12] I. Kiral-Kornek, S. Roy, E. Nurse, B. Mashford, P. Karoly, T. Carroll, D. Payne, S. Saha, S. Baldassano, T. O'Brien, *et al.*, "Epileptic seizure prediction using big data and deep learning: toward a mobile system," *EBioMedicine*, vol. 27, pp. 103–111, 2018.

[13] U. of Freiburg, "Seizure prediction project freiburg," 2019.

[14] M. Sugiyama and M. Kawanabe, *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press, 2012.

[15] D. Trabelsi, S. Mohammed, F. Chamroukhi, L. Oukhellou, and Y. Amirat, "An unsupervised approach for automatic activity recognition based on hidden markov model regression," *IEEE Transactions on automation science and engineering*, vol. 10, no. 3, pp. 829–835, 2013.

[16] S. Ramasamy Ramamurthy and N. Roy, "Recent trends in machine learning for human activity recognition , survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, p. e1254, 2018.

[17] T. Maekawa, D. Nakai, K. Ohara, and Y. Namioka, "Toward practical factory activity recognition: unsupervised understanding of repetitive assembly work in a factory," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 1088–1099, ACM, 2016.

[18] T. Fritz, E. M. Huang, G. C. Murphy, and T. Zimmermann, "Persuasive technology in the real world: a study of long-term use of activity sensing devices for fitness," in *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 487–496, ACM, 2014.

[19] J. K. Lee, S. N. Robinovitch, and E. J. Park, "Inertial sensing-based pre-impact detection of falls involving near-fall scenarios," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 23, no. 2, pp. 258–266, 2014.

[20] D. Cook, K. D. Feuz, and N. C. Krishnan, "Transfer learning for activity recognition: A survey," *Knowledge and information systems*, vol. 36, no. 3, pp. 537–556, 2013.

[21] A. Stisen, H. Blunck, S. Bhattacharya, T. S. Prentow, M. B. Kjærgaard, A. Dey, T. Sonne, and M. M. Jensen, "Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition," in

Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, pp. 127–140, ACM, 2015.

[22] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[23] R. Chavarriaga, H. Bayati, and J. D. Millán, "Unsupervised adaptation for acceleration-based activity recognition: robustness to sensor displacement and rotation," *Personal and Ubiquitous Computing*, vol. 17, no. 3, pp. 479–490, 2013.

[24] M. A. A. H. Khan and N. Roy, "Transact: Transfer learning enabled activity recognition," pp. 545–550, 2017.

[25] R. Ding, X. Li, L. Nie, J. Li, X. Si, D. Chu, G. Liu, and D. Zhan, "Empirical study and improvement on deep transfer learning for human activity recognition," *Sensors*, vol. 19, no. 1, p. 57, 2019.

[26] J. Wang, Y. Chen, L. Hu, X. Peng, and S. Y. Philip, "Stratified transfer learning for cross-domain activity recognition," pp. 1–10, 2018.

[27] J. Wang, V. W. Zheng, Y. Chen, and M. Huang, "Deep transfer learning for cross-domain activity recognition," in *Proceedings of the 3rd International Conference on Crowd Science and Engineering*, p. 16, ACM, 2018.

[28] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *Advances in neural information processing systems*, pp. 343–351, 2016.

[29] A. Rozantsev, M. Salzmann, and P. Fua, "Beyond sharing weights for deep domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[30] F. Duman, N. Özdemir, and E. Yildirim, "Patient specific seizure prediction algorithm using hilbert-huang transform," in *Proceedings of 2012 IEEE-EMBS International Conference on Biomedical and Health Informatics*, pp. 705–708, IEEE, 2012.

[31] H. Qu and J. Gotman, "A patient-specific algorithm for the detection of seizure onset in long-term eeg monitoring: possible use as a warning device," *IEEE transactions on biomedical engineering*, vol. 44, no. 2, pp. 115–122, 1997.

[32] A. B. Das, M. I. H. Bhuiyan, and S. S. Alam, "A statistical method for automatic detection of seizure and epilepsy in the dual tree complex wavelet transform domain," in *2014 International Conference on Informatics, Electronics & Vision (ICIEV)*, pp. 1–6, IEEE, 2014.

[33] A. Van Esbroeck, L. Smith, Z. Syed, S. Singh, and Z. Karam, "Multi-task seizure detection: addressing intra-patient variation in seizure morphologies," *Machine Learning*, vol. 102, no. 3, pp. 309–321, 2016.

[34] K. Gadhoumi, J.-M. Lina, F. Mormann, and J. Gotman, "Seizure prediction for therapeutic devices: A review," *Journal of neuroscience methods*, vol. 260, pp. 270–282, 2016.

[35] B. Direito, C. A. Teixeira, F. Sales, M. Castelo-Branco, and A. Dourado, "A realistic seizure prediction study based on multiclass svm," *International journal of neural systems*, vol. 27, no. 03, p. 1750006, 2017.

[36] M. Bandarabadi, C. A. Teixeira, J. Rasekhi, and A. Dourado, "Epileptic seizure prediction using relative spectral power features," *Clinical Neurophysiology*, vol. 126, no. 2, pp. 237–248, 2015.

[37] E. B. Assi, D. K. Nguyen, S. Rihana, and M. Sawan, "Towards accurate prediction of epileptic seizures: A review," *Biomedical Signal Processing and Control*, vol. 34, pp. 144–157, 2017.

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

[39] F. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[40] R. Hussein, H. Palangi, R. Ward, and Z. J. Wang, "Epileptic seizure detection: A deep learning approach," *arXiv preprint arXiv:1803.09848*, 2018.

[41] M. Golmohammadi, S. Ziyabari, V. Shah, S. L. de Diego, I. Obeid, and J. Picone, "Deep architectures for automated seizure detection in scalp eegs," *arXiv preprint arXiv:1712.09776*, 2017.

[42] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, and H. Adeli, "Deep convolutional neural network for the automated detection and diagnosis of seizure using eeg signals," *Computers in biology and medicine*, vol. 100, pp. 270–278, 2018.

[43] H. Khan, L. Marcuse, M. Fields, K. Swann, and B. Yener, "Focal onset seizure prediction using convolutional networks," *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 9, pp. 2109–2118, 2017.

[44] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, IEEE, 2017.

[45] K. Roth, A. Lucchi, S. Nowozin, and T. Hofmann, "Adversarially robust training through structured gradient regularization," *arXiv preprint arXiv:1805.08736*, 2018.

[46] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "The space of transferable adversarial examples," *arXiv preprint arXiv:1704.03453*, 2017.

[47] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 427–436, 2015.

[48] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. McDaniel, "On the (statistical) detection of adversarial examples," *arXiv preprint arXiv:1702.06280*, 2017.

[49] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[50] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[51] S. Sun, C.-F. Yeh, M. Ostendorf, M.-Y. Hwang, and L. Xie, "Training augmentation with adversarial examples for robust speech recognition," *arXiv preprint arXiv:1806.02782*, 2018.

[52] D. A. Dickey and W. A. Fuller, "Distribution of the estimators for autoregressive time series with a unit root," *Journal of the American statistical association*, vol. 74, no. 366a, pp. 427–431, 1979.

[53] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[54] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," *arXiv preprint arXiv:1605.05101*, 2016.

[55] B. Romera-Paredes, A. Argyriou, N. Berthouze, and M. Pontil, "Exploiting unrelated tasks in multi-task learning," in *International conference on artificial intelligence and statistics*, pp. 951–959, 2012.

[56] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 109–117, ACM, 2004.

[57] Q. Xu, S. J. Pan, H. H. Xue, and Q. Yang, "Multitask learning for protein subcellular location prediction," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 8, no. 3, pp. 748–759, 2011.

[58] X. Sun, H. Kashima, and N. Ueda, "Large-scale personalized human activity recognition using online multitask learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2551–2563, 2013.

[59] R.-G. Cirstea, D.-V. Micu, G.-M. Muresan, C. Guo, and B. Yang, "Correlated time series forecasting using multi-task deep neural networks," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 1527–1530, ACM, 2018.

[60] M. Rehman, C. Liew, T. Wah, J. Shuja, B. Daghighi, *et al.*, "Mining personal data using smartphones and wearable devices: A survey," *Sensors*, vol. 15, no. 2, pp. 4430–4469, 2015.

[61] D. Wang, E. Candinegara, J. Hou, A.-H. Tan, and C. Miao, "Robust human activity recognition using lesser number of wearable sensors," in *2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, pp. 290–295, IEEE, 2017.

[62] L. Cao, Z. Liu, and T. S. Huang, "Cross-dataset action detection," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1998–2005, IEEE, 2010.

[63] R. Chavarriaga, H. Bayati, and J. D. Millán, "Unsupervised adaptation for acceleration-based activity recognition: robustness to sensor displacement and rotation," *Personal and Ubiquitous Computing*, vol. 17, no. 3, pp. 479–490, 2013.

[64] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[65] C. Chu and R. Wang, "A survey of domain adaptation for neural machine translation," *arXiv preprint arXiv:1806.00258*, 2018.

[66] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[67] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[68] M. Long, Y. Cao, Z. Cao, J. Wang, and M. I. Jordan, "Transferable representation learning with deep adaptation networks," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[69] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, no. Mar, pp. 723–773, 2012.

[70] A. Rozantsev, M. Salzmann, and P. Fua, "Beyond sharing weights for deep domain adaptation," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[71] R. Ding, X. Li, L. Nie, J. Li, X. Si, D. Chu, G. Liu, and D. Zhan, "Empirical study and improvement on deep transfer learning for human activity recognition," *Sensors*, vol. 19, no. 1, p. 57, 2019.

[72] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," *arXiv preprint arXiv:1707.01217*, 2017.

[73] M. A. A. H. Khan, N. Roy, and A. Misra, "Scaling human activity recognition via deep learning-based domain adaptation," in *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 1–9, IEEE, 2018.

[74] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *International Conference on Machine Learning*, pp. 2342–2350, 2015.

[75] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[76] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *2010 IEEE Computer Society Conference on computer vision and pattern recognition*, pp. 2528–2535, IEEE, 2010.

[77] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in neural information processing systems*, pp. 2366–2374, 2014.

[78] S. Ruder, J. Bingel, I. Augenstein, and A. Søgaard, "Learning what to share between loosely related tasks," *arXiv preprint arXiv:1705.08142*, 2017.

[79] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3994–4003, 2016.

[80] T. Sztyler and H. Stuckenschmidt, "On-body localization of wearable devices: An investigation of position-aware activity recognition," in *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 1–9, IEEE, 2016.

[81] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Advances in neural information processing systems*, pp. 137–144, 2007.

[82] C. Nadeau and Y. Bengio, "Inference for the generalization error," in *Advances in neural information processing systems*, pp. 307–313, 2000.