DETECTING SPEECH FROM BODY MOVEMENTS:

A LOOK INTO THE NATURE OF SPEECH BASED ON NEURAL NETWORKS AND MULTI-SOURCE DOMAIN ADAPTATION

Thesis
submitted in partial fulfilment of the
requirements for the degree of
MASTER OF SCIENCE
in
Embedded System

Author:	BSc. X. Ni
Student ID:	4714431
Supervisor:	Dr. H. Hung
	Dr. E. Gedik

September, 2019

Intelligence System Department,
Faculty of Electrical Engineering, Mathematics & Computer Science,
Delft University of Technology,
Delft, The Netherlands.

CONTENTS

Summary v

1 Introduction 1
- 1.1 Research Motivation . . . 2
- 1.2 Research Questions . . . 3

2 Related Work 5
- 2.1 Human activity recognition . . . 6
- 2.2 Human activity recognition with video data . . . 6
- 2.3 Human activity recognition with Wearable sensors . . . 6
- 2.4 Speech detection with accelerometers . . . 8
- 2.5 Person specific nature of speech . . . 8
- 2.6 Transfer learning . . . 9
  - 2.6.1 Data shift . . . 9
  - 2.6.2 Deep Transfer Learning . . . 11

3 Dataset 13
- 3.1 Data Set . . . 14

4 Methodology 17
- 4.1 Data Preprocessing . . . 18
- 4.2 Hand-crafted feature extraction . . . 19
- 4.3 Neural Network for speech detection: CNN+RNN . . . 19
  - 4.3.1 Convolution Layer . . . 19
  - 4.3.2 Gated Recurrent Unit Layer . . . 20
  - 4.3.3 Model Structure . . . 21
- 4.4 Visualization for the features extracted by neural network . . . 21
- 4.5 Multi-source Domain Adaption . . . 22
  - 4.5.1 Strategy 1: Transductive Parameter Transfer . . . 22
  - 4.5.2 Kernel Function . . . 24
  - 4.5.3 Strategy 2: Two stage Sample Re-weighting . . . 25

5 Result Analysis and Discussion 27
- 5.1 Logistic Regression VS TPT . . . 28
- 5.2 Logistic Regression vs Neural network . . . 29
- 5.3 Neural Network based TPT . . . 30
- 5.4 Influence of Data Size . . . 31
- 5.5 Possible Upper bound for neutral network based TPT . . . 32
- 5.6 Two-stage sample re-weighting . . . 32
- 5.7 The existence of person-specificity . . . 33

6 Conclusion and Future Work **37**
6.1 Conclusion 38
6.2 Future work 40

Acknowledgements **43**
References 44

SUMMARY

Our research focuses on speech detection from body movements using wearable accelerometer data collected in an in-the-wild mingling event. We aim to explore the nature of the connection between speech and body movements. More specifically, we stress on the person-specificity of speech. Many studies have shown that speech always comes along with unconscious body behaviours [1]. There is a strong correlation and synchrony between speech and body movements [2]. Previous research [3, 4] has proved that human behaviour is highly person-specific. In other words, in our experiment set-up, the accelerometer data distributions collected from different persons are different. Based on the two considerations discussed above, our work contains two phases. In the first phase, we investigate utilizing convolutional and recurrent neural networks for learning informative representations from raw body acceleration readings. The model we proposed outperforms the state-of-the-art approach presented in [5] by 6 % (Area Under the Curve) with the same data. In the next stage, we visualize the features extracted by the proposed model. The results show that distributions of data obtained from different individuals can differ (also known as person-specificity of the problem). We adopt two approaches of multi-source domain adaption [6] based on the features extracted by our model, aiming to form a personalized speech detection model for each person in our dataset. The first approach is called transductive parameter transfer (TPT) [5]. It deduces the personalized model of the target domain from the known well-trained models of several source domains based on the assumption that distributions of individuals with similar marginal distributions should also have similar decision boundaries. The second strategy is a sample re-weighting based method where the training samples from different persons are re-weighted with respect to the similarities of their conditional and marginal distributions to the target person. We use those re-weighted samples to train a personalized model for each target person. The approaches we adopted only achieved a relative performance increase compared to the general neural network model trained on all the data. We then discuss the possible reasons why these two methods did not bring significant improvement and what can be the alternative solution in the future.

1

INTRODUCTION

1

1.1. RESEARCH MOTIVATION

Our work focuses on detecting speech from body movements in an in-the-wild mingling event, investigating in the connection between speech and body movements. Speech is a type of social behaviour [5] through which we communicate with each other, share opinions, express attitudes. Many works have been done about detecting speech directly from audio data [7–9], however, these research purely focused on the detection of speech audio frame from the audio recording. Our work is different from their works since we attempt to explore the connection between speech and body movements, focusing on the nature of body movements when people are speaking and the upper bound for inferring speech from body movements.

The detection of speech from body movements is closely linked with many works in Social Signal Processing (SSP) [1, 10], which focuses on enabling machines with the ability of recognizing social signals (e.g. people's mood, attitude, activeness during a social event) [11]. One of the core research areas of SSP is to find out if it is possible to automatically infer social signals from body movements or nonverbal behavioral cues (e.g. gestures, postures) [11]. Speech is a vital unit of social behaviour, the foundation of communication. If the connection between speech and body movement patterns can be established, then, such connection and corresponding algorithms can also be applied to the detection of other social signals. Furthermore, since speech is the basis of interpersonal communication, many SSP related studies are based on the detection of speech. For example, works [12, 13] used speaking activity (e.g. speaking time, interruptions, number of times) to detect who is in the dominance during a meeting and work [14] used speech related features (e.g. mean of pitch , energy, fidgeting) to classify people' s personality. For many tasks like role recognition and social attitude detection, speaking activity related features have been shown to be the most effective ones[1]. If a comprehensive approach for speech detection can be built, then many advanced, further studies mentioned above can be better executed.

Research in Human Activities Recognition (HAR) [15–17] has shown that many human actions (e.g. running, walking, jumping) can be detected from body movement information collected by accelerometers. However, unlike traditional human activities, people's behaviour performed during speech is unconscious. Many studies [2, 18] in social psychology have shown that there is a strong correlation between speech and body movements. Several works [19, 20] have also proved the possibility of detecting speech based on a single worn accelerometer. However, these works simply treated speech detection as a classification task and did not consider the huge variation in ways in which people behave when talking. Furthermore, these works used hand-crafted features which are less representative than our neural network model based features.

In our research, we use a single accelerometer worn on people's chest to collect body movement information. Compared with audio or video data source, using accelerometer naturally has the following two advantages:

1. Our experiment is conducted in a crowded mingling event. For audio data, such

a scenario will bring high non-stationary background noise. Raw audio data requires sophisticated reprocessing before being used [7–9]. For video data, due to the large number of participants during the experiment, the occlusion of people in the video is inevitable. These facts bring difficulties when we want to observe people's behaviour during speech. Meanwhile, for accelerometer data, either of the problems needs to be concerned. Each individual's data is natural separated and not contaminated by the others' data.

2. Audio and video record people's conversation directly. Such recordings usually cause privacy concerns especially in the real life. Compared with audio and video, the accelerometer does not record the content of conversations and is less affected by privacy concerns. Due to this concern, accelerometers based speech detection approaches are more possible to be applied to our daily life.

Another challenge we need to pay attention to is the person-specificity phenomenon. Research in HAR [3, 4, 21, 22] has shown that human behaviours vary significantly between persons, thus the data extracted from different subjects may follow different distributions. [5] also proved this observation in their research about speech detection. They found that a personalized individual model trained with data collected from single person performs much better than a general model trained with data collected from the other subjects. This indicated that data collected from different subjects have different distributions. A general model for all the subjects may eradicate person-specificity and could achieve a less satisfying performance. Based on the discussions above, how to gain a personalized model without labeled data of the target person (subjects in the test set) is one of our emphasis in our research.

Neural networks are a set of algorithms that can recognize underlying relationships of raw data automatically. Neural networks have been widely applied in HAR for their ability to learn representative features directly from the raw time-series data. When it comes to speech detection from accelerometers, currently all the previous research focused on hand-crafted features. No research has applied neural networks for speech detection. In our research, we will use the neural network to extract features from the raw accelerometer data. Meanwhile, as discussed in the former paragraph, to overcome the challenge of person-specificity, we will try to apply two approaches of multi-source domain adaption (transductive parameter transfer (TPT) [5] and two-stage sample re-weighting [23]) to the features extracted by the neural network, aiming to find the optimized personalized model for each target person. Our work is a combination of neural networks and multi-source adaption for speech detection from accelerometer readings.

1.2. RESEARCH QUESTIONS

1. Currently, for speech detection based on accelerometers, the state of art approach (hand-crafted features based TPT) [5] is based on a small data set with 18 subjects, each with 10 minutes of recording. For our research, we have a larger data set (50 subjects, 30 minutes recording for each person). Our first research question is, how will the TPT perform on a larger data set?

2. Compared to the hand-crafted features based approach in [5], whether we could find a neural network model that could generate representative features from the raw accelerometer and outperform the state of the art [5]?

3. Whether the person-specificity assumption still holds for the features extracted by a neural network (different distributions)? If so, can we combine the neural network and TPT (applying TPT to the features extracted by the neural network) to get a better performance per person?

4. TPT holds the assumption that the optimized personalized decision boundary (conditional distribution) is determined by the marginal distribution. Is this assumption true and what is the possible way to adapt TPT to get a better performance? What is the upper performance bound for TPT?

5. In most cases, neural networks are data-hungry. How much data is enough for a well-trained model for speech detection? Will we get a better performance if more data is provided?

6. Compared to the TPT, the second approach we proposed (two-stage sample re-weighting) does not hold any assumption about the data distributions. It re-weights the samples from source domain persons based on their conditional and marginal distribution similarity to the target person (subjects in the test set). Will combing the neural network and the two-stage sample re-weighting bring better performance?

7. Does the assumption of person-specificity really hold in speech detection in our experimental set-up? Do the personalized neural network model (trained with data from the target person) really outperform the general model that trained on the data from all the other subjects?

2

Related Work

2.1. HUMAN ACTIVITY RECOGNITION

Human activity recognition (HAR) shares many similarities with our research. There are two main categories in terms of HAR: HAR with video data and HAR with wearable sensors. Here we first briefly discuss video based HAR.

2.2. HUMAN ACTIVITY RECOGNITION WITH VIDEO DATA

Four types of hand-crafted features are usually used in video based HAR, namely, space-time volume, frequency transform, local descriptors and body modeling [24]. The space-time volume (STV) feature is formed by stacking the consecutive silhouette of objects along the time axis. Work [25, 26] used this feature to detect simple human actions like jumping, walking and bending. Discrete Fourier transform (DFT) can also be used for detecting human activities from the video [27]. It has been widely adapted to present the geometric structure of the objects in the video. However, both STV and DFT are global features that represent the entire image. STV and DFT could perform badly when people are occluded in the image. Many studies [28, 29] also used local descriptors (scale-invariant feature transform, histogram of oriented gradients). Those descriptors capture the characteristic of different parts of an image. Compared to STV and DFT, they are invariant in terms of occlusions, rotations and scale. [28] used the PCA-HOG descriptor to track and recognize people's behaviours during sports events. All the features mentioned above focused on the representation of an image, not the human body. Thus, many other works investigated building 2D or 3D body models and transferring the models into more discriminate feature representations like geometric relational features [30] and Boolean features [31].

2.3. HUMAN ACTIVITY RECOGNITION WITH WEARABLE SENSORS

Many HAR works focused on detecting human actions from body movement information collected by wearable sensors. Those approaches applied in HAR can also be applied to our research. We now discuss works about HAR with wearable sensors. Also in the following parts of our thesis, when we refer to HAR, it always means HAR with wearable sensors.

HAND-CRAFTED FEATURES

HAR with wearable sensors mainly focuses on detecting daily activities (e.g. walking, running and jumping) based on sensor data of different modalities (e.g. gyroscope, accelerometer). Early works [3, 17, 32] of HAR focused on various combinations of hand-crafted features (e.g. time domain, frequency domain), modalities and classifiers (e.g. SVM,KNN). Work [3] investigated in the detection of eight common human activities (e.g. standing, walking, running, brushing teeth...). They placed a tri-axial accelerometer around the pelvic region to capture body movement information. The data was collected from two subjects in multiple rounds over different days. The collected raw time series data was then transformed by Fast Fourier Transform (FFT) to form spectral features. The final feature they used was a mixed set of time and frequency domain features (mean, standard deviation, energy, correlation). Their result showed that a plurality voting model consists of several basic classifiers (Decision Tables, Decision Tree,

k-nearest Neighbor, Naive Bayes) achieved the best performance with an average accuracy higher than 90 %. [17] made an effort to find out what types of features are best for HAR. They placed accelerometers at people's ankles and thighs for data collection. Then they compared the accuracy of different combinations of statistical and spectral features using KNN as the base classifier. The result showed that the feature consists of the magnitude of the first five components of FFT analysis achieved the highest accuracy. They announced that FFT related features are the most efficient features for HAR. Research [32] made a further step by applying HAR under a less-constraint environment. In their experiment, they placed 5 accelerometers on different positions of subjects' body. The data was collected from 20 subjects without researcher supervision or observation. Their result was quite similar with [17], suggesting that FFT-based features are the most suitable features for HAR.

REPRESENTATION LEARNING

Neural networks are universally applied in HAR since it could generate representative features automatically from the raw data [4]. Work [15] has shown that features learned by some automatic methods (e.g. Principle Component Analysis and Restricted Boltzmann Machine) could achieve an equal or higher performance compared to hand-crafted features in HAR. A great conclusion has been done in [33] in the area of applying neural networks to HAR. This paper also pointed out that shallow features (e.g. mean, variance, frequency) extracted based on human expertise can only be used to recognize low-level activities (e.g. walking, cycling, vacuuming). However, they are insufficient for detecting high-level or contextual activities like talking.

An early work presented in [34] applied the convolutional neural network (CNN) to HAR. It treated each dimension of the accelerometer data as a channel. Subsequently, it performed 1D convolution separately on each channel. There are also other automatic feature learning methods like Principle Component Analysis(PCA), Restricted Boltzmann Machine (RBM) mentioned in [15]. [34] compared their CNN model with those methods, and their model achieved the highest accuracy on all the datasets (Opportunity [35], Skoda [36]) they used for testing. Afterward, the long-short term memory network (LSTM) [37] has also been introduced in HAR in [16]. Later works [38–41] came up with different variations of the CNN-based or RNN-based model, aiming to bring a better HAR performance. Most of the neural networks based HAR research made use of public HAR datasets like Opportunity, PAMP2 [42]. These datasets are collected under a sensor rich environment. For example, dataset Opportunity contains data collected from multi modal sensors including accelerometer, gyroscope and magnetometer. Those sensors are placed at different positions of the human body. The final raw data dimension of research [43], which used dataset Opportunity, is 113. However, for our research, only one accelerometer is available, the dimension for our raw data is just 3. Work [40] needs to be emphasized. It established a CNN-based model and fed it with the data from a single accelerometer. Their model achieves an accuracy around 95 % on detecting actions like falling, jumping and running. Their work proved that for the neural network model, data collected from a single accelerometer is enough for recognizing many basic actions (falling, jumping and running).

2.4. SPEECH DETECTION WITH ACCELEROMETERS

Speech detection based on accelerometers is rarely explored. Several early works have been done in [5, 44]. Work[5] focused on establishing a personalized model for speech detection using transductive parameter transfer(TPT)[45] based on the discovery that the distributions of hand-crafted features extracted from different subjects are different. Our research is an extension of their work. Compared to their work, we use data from a larger range of participants (50 people vs 18 people) and for each person, a longer recording time is adopted (30 mins vs 10 mins). Also, instead of using predefined hand-crafted features, a neural network model is applied for feature extraction.

2.5. PERSON SPECIFIC NATURE OF SPEECH

As has been discussed in 1, for HAR and speech detection, person-specificity could be a challenge when the training data comes from multiple subjects. Since the data distributions of different persons are different, training a model with the data from multiple subjects may lead to a less satisfying result. Work [3] proved the existence of person-specificity in HAR. In their experiments, they compared several classifiers' (Decision Tables, Decision Tree, k-nearest Neighbor, Naive Bayes) performance under the person-specific set-UP and person-independent set-UP. With the "person-specific" setup, train and test sets come from the same person. For the "Person-independent" setup, train and test sets come from different subjects. The accuracy performance with the person-specific set-up is around 30 % higher than the performance with the person-independent set-up (90% vs 60%). This phenomenon indicated that the feature distributions of different subjects are different, in other words, human activities are person-specific. Person-specificity also has been proved in [4] with the neural network model. Their model combined CNN and LSTM, using CNN for learning temporal relationships between time steps while RNN for learning long-term activity information. In their experiments, they trained and tested the model on the data that came from different subjects. After model training, they fine-tune the model using the data from the subjects in the test set. The result showed that the fine-tuning model achieved a higher F1-score compared to the original model when tested on the subjects in the test set. Their discovery complied with the results in [3], indicating that human activities are intended to be highly person-specific and this property hinders the learning process of classification models. Since our research focuses on detecting speech from body movements just like HAR, we believe this discovery can also be applied to speech detection. Work [5] showed the person-specificity in terms of speech. In their experiment, for each subject, they trained a Logistic Regression (LR) classifier with data come from the same subject and a LR with the data from the other subjects separately. The classifier (LR) trained with individual data performed much better than the one with the data from other subjects (10 % higher in terms of Area Under the Curve).

2.6. TRANSFER LEARNING

2.6.1. DATA SHIFT

As discussed in the former sections, person-specificity is a serious obstacle when we want to improve the model performance in terms of HAR and speech detection. In the following discussion, we will define the concept of person-specificity mathematically. One of the basic assumptions of machine learning is that the distribution of train and test sets are independently identical distributions. But what if the distributions of the test and train set are different? In other words, there is a data shift exists between the train and test set. A great conclusion work has been done in [46]. According to their definition, there are three basic types of data shift: prior shift, covariate shift and concept shift. In the following discussions, we use s to denote the source domain (train set) and t (test set) for the target domain. Before the discussion, we first give the definitions of the marginal and conditional distribution.

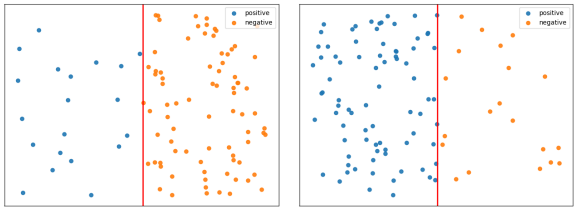
1. Marginal Distribution: A domain D is composed of a feature space $\mathscr{X}$ and marginal probability distribution $p(x)$, $D = \{\mathscr{X}, p(x)\}$, where $x \in \mathscr{X}$

2. Conditional Distribution: Given domain $D = \{\mathscr{X}, p(x)\}$, Suppose y is the corresponding label for x, we have a function such that $y = f(x)$, $f(x) = p(y|x)$ can be interpreted as the conditional probability distribution [47]

3. Joint Distribution: Given the marginal distribution $p(x)$ and $p(y|x)$, we could form the joint distribution as $p(x,y) = p(y|x)p(x)$. In the following discussion, we use $p_s(x,y) = p_s(y|x)p(x)$ to denote the data distribution of the source domain (from where the model are trained) while $p_t(x,y) = p_t(y|x)p(x)$ for the target domain.

PRIOR SHIFT

For prior shift, the prior probabilities of the classes are different, i.e. $p_s(y) \neq p_t(y)$, while the conditional distributions are the same, $p_s(x|y) = p_t(x|y)$. For example, in the train set (source domain), the prior probability for the positive and negative class are respectably 1/5, 4/5, while in the train set (target domain), this ratio might be changed to 4/5, 1/5. However, under this situation, the classifier decision boundary should be the same in the source and target domain. As shown in the above figure, although the



Source Domain Distribution Target Domain Distribution

Figure 2.1: Demonstration of Prior Shift

prior probability distributions of the source and target domain are different, the decision

boundary remains the same. However, when the class distribution is extremely imbalanced, it is possible that during the training process, the classifier simply only considers the dominant class [46]. In this case, we will balance the class weights inversely proportional to class frequencies in the input data when training the basic classifier.

2

COVARIATE SHIFT

Another type of shift is called covariate shift. Under this situation, the target and source domain share the same distribution model $p(y|x)p(x)$. However, due to problems like biased sampling, $p_s(x)$ and $p_t(x)$ can be different. Also $p_s(y|x)$ and $p_t(y|x)$ are different [48]. In other words, target domain and source domain lie at different places of a universal distribution. As shown in 2.2, the target domain and source domain are in two

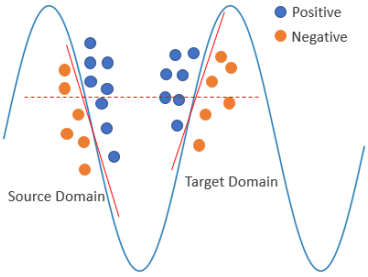
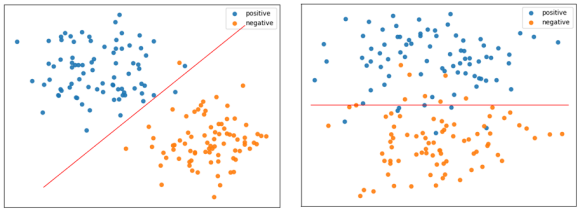


Figure 2.2: Demonstration of Covariate Shift

different areas of the entire distribution. If a linear classifier like logistic regression is used, the decision boundary of the target and source domain would be different. If we train a linear classifier considering both the source and target domain, just like the horizontal line in 2.2, then this classifier would be too general to have a good individually optimized performance.

CONCEPT SHIFT

In this case $p_s(x) = p_t(x)$, while $p_s(y|x) \neq p_t(y|x)$. Their decision boundaries violate with each other. In most of the cases, concept shift can not be easily solved without knowing the target domain label.



Source Domain Distribution Target Domain Distribution

Figure 2.3: Demonstration of Concept Shift

MARGINAL AND CONDITIONAL DISTRIBUTION

In the real world, the data shifts discussed above can happen at the same time [47]. Many studies directly uses marginal distribution and conditional distribution [47, 49] when discussing data shift. Many problems in real life are the combination of both marginal and conditional distribution differences($p_s(x) \neq p_t(x)$ and $p_s(y|x) \neq p_t(y|x)$). This kind of situation is less strict than covariate shift. For covariate shift, target domain and source domain share the same universal distribution. For the simplicity of our discussion, we will use the terminology conditional distribution and marginal distribution in the following chapters.

2.6.2. DEEP TRANSFER LEARNING

As has been discussed in the above chapter, the 'person-specific' issue seems to be a factor that constrains the performance of the neural network models in HAR. Models learned from one subject's data do not generalize well on the target person. So how to transfer knowledge between different subjects? Several works [21, 22] tried to apply transfer learning to neural networks to overcome this issue. Here we refer the person included in the train set as source domain while person for the test as the target domain. As shown in figure 2.4, in work [21], a Maximum Mean Discrepancy(MMD) loss function is added at the end of the neural network. The neural network can be regarded as a kind of feature extractor. By minimizing the MMD distance between the feature distributions of the source domain and target domain, this model is forced to map the raw data from different domains to the same feature space.

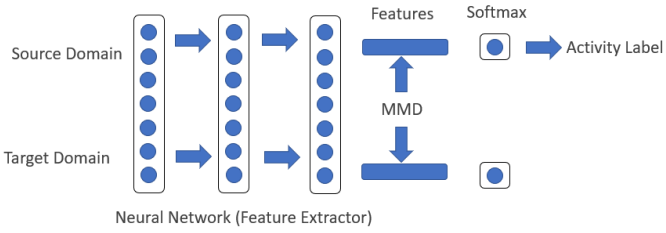


Figure 2.4: The structure of Transfer Neural Network for Activity Recognition

Another structure is so-called Domain-adversarial Neural Network [22]. As shown in figure 2.5, features extracted by the neural network will be sent to an adversarial layer. This layer tries to classify whether the features come from the source domain or the target domain. The worse this layer performs, the better the features extracted by the network are, since there is little difference between the source features and target features.

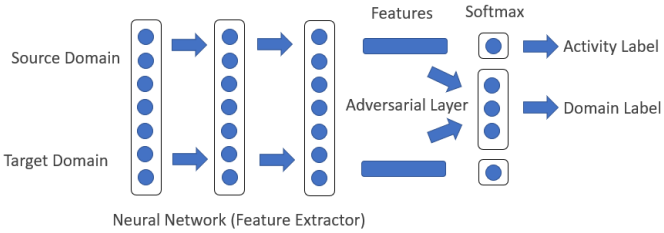


Figure 2.5: The structure of Adversarial-based Transfer Neural Network for Activity Recognition

However, the method we discussed above only focused on the transfer from one domain to another. Things would be more difficult in our experimental set-up since we have 50 subjects (multiple-source domains) in our dataset, and each person only has 875 training frames which are not sufficient to train a neural network.

3

DATASET

3.1. DATA SET

Our work is based on the dataset *MatchNMingle* [50], which is specially designed for SSP research purposes. It is a multimodal dataset for the analysis of free-standing conversational groups and speed-dates in the wild. Table 3.1 shows the main contents of *MatchNMingle*.

Sensor/Input	Modality/Survey	Details
Questionnaires	HEXACO	Score and sub-score for each trait
	SOI SCS	
	Data Responses	All dates in the event
Hormone baseline	Cortisol Testosterone	Collected using hair sample
Cameras	Video	9 overhead cameras recording both the speed dates and mingle
	Audio	General audio from the event
	Frontal Photos	Face (neutural/ smile) + full body
Wearable Sensors	Acceleration	Triaxial at 20Hz for entire event
	Proximity	Binary values at 1 Hz for entire event
Manual Annotations	Positions Social Actions	30 mins at 20 FPS for the mingle
	F- Formations	10 mins at 1 FPS for the mingle

Table 3.1: Summary of all the elements included in *MatchNMingle*

MatchNMingle was collected in an indoor in-the-wild scenario, during 3 real speed date events, each followed by a cocktail party. The social events are held in 3 days. 92 participants have joined the experiment, each of them attended one of the 3 speed date events in a public pub followed by a mingle cocktail party (60 minutes). Wearable sensors and cameras are used to collected multimodal data (accelerator proximity and video) during the event.

Figure 3.1: Snapshot of *MatchNmingle* session

Among the 92 participants, there were 46 women (age: 19-27) and 46 men (age: 18-30). In our research, we make use of the accelerometer data collected during the mingle cocktail session. *MatchNMingle* used a single wearable tri-axial sensor worn around the chest to collect the body movement information of each participant. The sample rate

of the sensor is 20 Hz. Over 30 minutes of recording is manually labeled per day. There are 8 social actions annotated during the 30 minutes labeled segmentation: 1) Walking, 2) Stepping, 3) Drinking), 4) Speaking, 5) Hand Gestures, 6) Head Gestures, 7) Laugh and 8)Hair Touching. For our research purpose, we will use the annotations for speech. Compared with other commonly used HAR datasets (e.g. OPPORTUNITY, PAMAP2), our dataset has several properties:

- Single Modality: *MatchNmingle* only has one accelerometer for each person. The dimension of the raw data is small (data from three axes). However, for other datasets like OPPORTUNITY [35], 23 body-worn sensors are used for each subject. On the one hand, using only one accelerometer, the information we collected could be limited. On the other hand, such a set-up may minimize the effect of measurement equipment, so people can behave in a more natural way.
- Large subjects set: our dataset contains data from 92 participants. Although we only make use of 50 out of all of them, compared with other sets (OPPORTUNITY: 4, PAMAP2:9), we still have a relatively large data set source, this gives us a chance to observe how people's speech behaviour varies from different subjects.

4

METHODOLOGY

4.1. DATA PREPROCESSING

MatchNmingle contains about 30 minutes labeled multi-modal (e.g. video, accelerometer) recordings of 92 people in a crowded mingling scenario. Here we use the accelerometer data to detect speech status (video recordings are used for manual annotation). It collects the movement data from participants' chest from 3 axes with a sample rate of 20 Hz. Thus, there will be 36000 samples for each person. For each channel, we apply the Z-score standardization.

$$z = \frac{x - \mu}{\sigma} \tag{4.1}$$

where μ is the mean of all samples and σ is the standard deviation of all samples.
Then, the raw time-series data is segmented with a sliding window of 3 seconds and an overlap of 1.5s. Each interval (frame) was labeled by majority voting. Most of the HAR

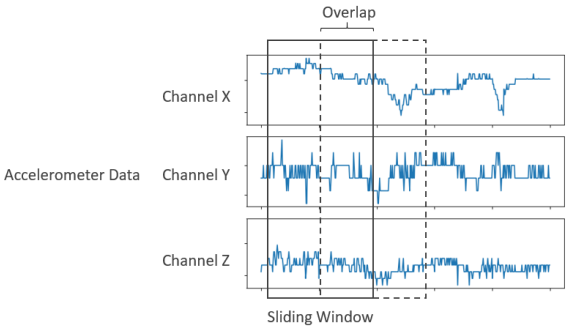


Figure 4.1: Sliding Window Demo

research based on time-series data adopts a similar segmentation (the selection of the length of the sliding window can be different) and labeling method described above. Several works also used the label of the frame's last sample as its label [16]

However, with the majority voting label method, an interval with 49 % speaking samples will be labeled as "non-speech" while the one with 50 % speaking samples will be labeled as "speech". Those intervals in the gray area unnecessarily increase the difficulty of classification. Thus in our research, for each subject's data, we will only retain intervals where samples are purely labeled as "speech" or "non-speech". Although we also use AUC as the evaluation standard, we still expect the positive-negative frame distribution to be less imbalanced, so we select subjects whose speech period is longer than 15% (4.5 minutes) into our data set. Finally, we get a data set containing 50 subjects with averagely 875 data frames per person. The shape of the data frame is a 3 x 60 matrix.

We perform a leave-10-subjects-out cross-validation for all the experiments we did in our project. For each fold, we select 10 persons' data out of the 50 subjects as the test set, while the others (40) remain as the train set. In the meantime, we also want to ensure that the "speech" and "non-speech" frame ratio in the test sets and train sets are approximately the same. Since the ratio of "speech" and "non-speech" varies significantly

between different subjects, it brings difficulties in the data division. We want every person included in our experiments to appear at least once in the test set while the distribution balance needs is guaranteed. We number the person we have selected from 0 to 49. Table 4.1 shows the components of each fold's test set while the other subjects are

	Test set
Fold0	0 1 2 3 4 5 6 7 31 45
Fold1	16 17 18 19 20 21 22 23 8 13
Fold2	24 25 26 27 28 29 30 31 14 48
Fold3	32 33 34 35 36 37 38 39 9 13
Fold4	40 41 42 43 44 45 46 47 48 49

Table 4.1: Test Set for Each Fold

correspondingly assigned in the train set. The overall speech frame ratio of both train and test set for all folds is approximately 25%. In total, there are 47 subjects appearing in the test sets. Subjects 13, 31, 45 appear twice in different folds. For these people, we will take the average AUC performance in the following result analysis.

4.2. HAND-CRAFTED FEATURE EXTRACTION

To answer the first research question, whether hand-crafted features based TPT is still valid when a larger dataset is given, we first reimplement the approach in [5], the hand-crafted features we used here is exactly the same as in [5]. AS has discussed in the former section, each channel of the acceleration is first standardized by Z-score standardization. The features we adapted come from time and frequency domains. For time-domain features, we calculate the mean and variance of the raw acceleration, absolute value of the acceleration and magnitude of the acceleration. For frequency domain, we first apply FFT to the raw acceleration, the absolute value of the acceleration and the magnitude of the acceleration separately, then calculate the corresponding power spectral density (PSD) by logarithmically binning the FFT components in a range of 0-8 Hz. The final length of the feature per window we get is 70.

4.3. NEURAL NETWORK FOR SPEECH DETECTION: CNN+RNN

4.3.1. CONVOLUTION LAYER

The convolution layer has been widely used in the area of HAR to learn features in time series. Instead of using traditional 2D CNN for tasks like image classification, 1D CNN is used in our structure. The demo of 1D CNN is shown below in figure 4.2
We choose RELU as the activation function

$$f(x) = max(0, x) \tag{4.2}$$

Also, we use xavier normal initialization [51] for the kernel initialization while assigning "0" for the bias initialization. For xavier normal initialization, suppose for each neural in a layer, the number of input units is N_{in} and the number of output units is N_{out}, then the initialization weight is drawn from a a truncated normal distribution centered on 0 with a standard deviation of $\sigma = \sqrt{2/(N_{in} + N_{out})}$.

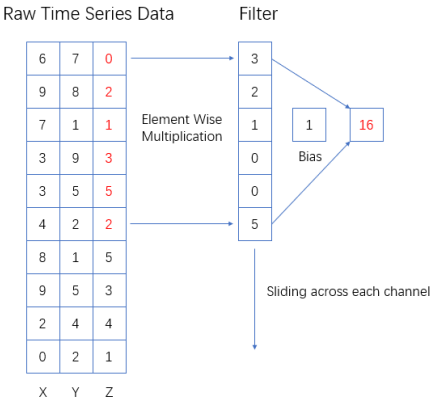


Figure 4.2: 1D CNN for Feature Extraction

4.3.2. GATED RECURRENT UNIT LAYER

Instead of using LSTM, we selected Gated Recurrent Unit (GRU) [52] in our structure. Since GRU has less parameters than LSTM, it is less likely to be over-fitted and takes fewer epochs to converge. Suppose for each time step t the input is x_t, while the output of $t-1$ is h_{t-1}. The structure of GRU is shown below:

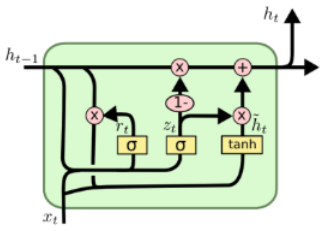


Figure 4.3: Structure of Gated Recurrent Unit

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{4.3}$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{4.4}$$

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t * h_{t-1}) + b_h) \tag{4.5}$$

$$h_t = z_t * h_{t-1} + (1 - z_t) * \tilde{h}_t \tag{4.6}$$

Here , $*$ denotes element-wise multiplication. Xavier normal initialization is used for kernel initialization, and "0" is assigned for the bias initialization. Suppose the number of units of the GRU layer is n , and the shape of the input tensor is $a \times b$, where b is the number of time steps. Then, x_t is a vector of shape $a \times 1$. W_r, W_z and W_h are matrices of shape $n \times a$. U_r, U_z and U_h are matrices of shape $n \times n$. b_r, b_z and b_h are bias vectors of shape $n \times 1$.

4.3.3. MODEL STRUCTURE

In our project, we set up a CNN+GRU model for speech detection using accelerometer data. Research has shown that hybrid models usually achieve good performance in HAR [33]. For our work, we use CNN to learn the local relationships between neighboring time steps. And GRU is applied for learning longer dependency within each frame. The detail for our model is shown below:

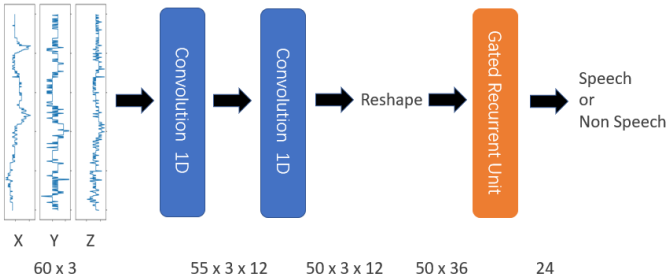


Figure 4.4: Neural Network for Speech Detection

Number of Filters of CNN	12
Output Dimension of GRU(Unit) each time step	24
Optimizer	RMSprop[53]
Batch Size	512
Training Epoch	200
Recurrent Dropout rate for GRU	0.1

Table 4.2: Details about the Neural Network Model

4.4. VISUALIZATION FOR THE FEATURES EXTRACTED BY NEURAL NETWORK

Now we visualize the features extracted from the last GRU layer and check how the data distributions of different subjects look like.

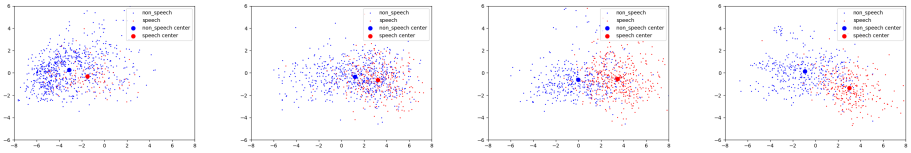


Figure 4.5: PCA Analysis for person 0,1,2,3

The above figures show the data distributions of 4 different persons in fold 0. As we can see, their feature distributions are different from each other. For the neural network

model, the last layer of the neural network is simply a logistic regression. An universal logistic regression might not be adapted well for all the people. Following the same idea in [5], we use multi-source domain adaption to deduce the optimized individual logistic regression for each person's data.

4.5. MULTI-SOURCE DOMAIN ADAPTION

Suppose that we collect data from n subjects in our train set. For each person, we extract features of length 24 from the raw accelerometer based on the neural network model in 2. Each subject's data represents a source domain. Then we have $D^s = \{D_i^s\}_{i=1}^n$ and $X^s = \{x_i^s\}_{i=1}^n$. Each subject's marginal distributions is different from the others, $p(x_i^s) \neq p(x_i^s)$. Their conditional distributions are also different, $p(y_i^s|x_i^s) \neq p(y_j^s|x_j^s)$. For each subject, we assume there will be a individually optimized logistic regression classifier $f^s = \{f_i^s\}_{i=1}^n$. Given a target distribution x^t, $p(x^t)$, we aim to find its optimized decision boundary. We try to solve this challenge with two possible strategies. For strategy 1, since we know the labels of the source domain, for each subjects i, we could get the optimized individual f_i^s. We directly deduce f^t from $f^s = \{f_i^s\}_{i=1}^n$, based on the similarity between $p(x^t)$ and each $p(x_i^s)$. For strategy 2, we reweight the samples of source domains based on their similarity to the target domain in terms of conditional and marginal distribution. Then those re-weighted samples are used to train a personalized model for the target person.

For all the following discussions, we will use Logistic Regression as the basic classifier (LR). Suppose for known domain (x_i, y_i), where $(x_i, y_i) = \{(x_{ij}, y_{ij})\}_{j=1}^m$, we aim to find the function $f_i = wx + c$ such that:

$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{j=1}^{m} \log(\exp(-y_{ij}(x_{ij}^T w + c)) + 1) \tag{4.7}$$

The hyper parameter C is found by k-fold cross validation and the logistic regression classifier(LR) is trained on the entire dataset with the optimal C.

4.5.1. STRATEGY 1: TRANSDUCTIVE PARAMETER TRANSFER

Here we assume that for domain x_i, the conditional distribution $p(y_i|x_i)$ is bounded with $p(x_i)$. In other words, the optimized f_i is decided by $p(x_i)$. Following the idea in [5], as we have 40 subjects in our train set, we want to learn a mapping function from the distribution $p(x_i)$ to f_i, $\hat{f} : p(x_i) \rightarrow f_i$. Then given any target distribution $p(x^t)$, we could derive the corresponding f^t by $\hat{f}$.

Algorithm 1 Transductive Parameter Transfer

Input: Source sets D_1^s,...,D_n^s with labels and target set x^t
  Compute $\{f_i = (w_i, c_i)\}$ using 4.7
  for each parameter z_{ij} of f_i **do**
    Creating training set $\tau = \{x_i^s, z_{ij}\}_{i=1}^n$
    Compute the kernel matrix K where $K_{ij} = k(x_i^s, x_j^s)$ using 4.19
    Given K and τ, compute f_j^t by solving 4.16 and 4.17.
  end for
Output: Classifier for the target set x^t, $f^t = [z_1^t, ..., z_{24}^t]$

Let us first assume that, for each distribution $\{x_i^s\}_{i=1}$ in the train set $X^s = \{x_i^s\}_{i=1}^n$, we have a $r \times 1$ vector v_i^s to uniquely represent $p(x_i^s)$. v_i^s can be obtained by a function $v_i^s = T(x_i^s)$. As we have discussed in 2, the last layer of our neural network can be regarded as a logistic regression classifier, and the output of the GRU is the encoded features. The length of the feature is 24, thus our logistic regression f_i^s has 25 parameters (including intercept). Suppose that z_{ij}^s is the jth parameter of f_i^s, and $z_j = [z_{1j}^s, z_{2j}^s, .., z_{nj}^s]^T$. Also, we define $V = [v_1^{sT}, ..., v_n^{sT}]$ and V is the matrix composed by v. Here we deploy Kernel Ridge Regression (KRR) [54]

$$\min((z_j - Vw_j)^T(z_j - Vw_j) + \lambda||w_j||^2) \tag{4.8}$$

where w_j is the corresponding mapping(vector of shape $r \times 1$) from the given distribution $p(x_i)$ to the j th parameter of the final logistic regression classifier f_i. The solution for 4.8 is given by

$$w_j = (V^T V + \lambda I_R)^{-1} + V^T z_j = (\sum v_i^s v_i^{sT} + \lambda I_R)^{-1} V^T z_j \tag{4.9}$$

Where I_R is diagonal matrix of shape $r \times r$. Using the matrix inversion lemma, we rewrite 4.9 as

$$w_j = V^T(VV^T + \lambda I_N)^{-1} z_j \tag{4.10}$$

and

$$VV^T = \begin{bmatrix} v_1^{sT}v_1^s & v_1^{sT}v_2^s & \cdots & v_1^{sT}v_n^s \\ v_2^{st}v_1^s & v_2^{sT}v_2^s & \cdots & v_2^{sT}v_n^s \\ \vdots & \vdots & \ddots & \vdots \\ v_n^{sT}v_1^s & v_n^{sT}v_2^s & \cdots & v_n^{sT}v_n^s \end{bmatrix} \tag{4.11}$$

$$v_i^T v_j = k(x_i, x_j) \tag{4.12}$$

For any pair $v_i^{sT}v_j^s$, we replace it by a kernel function $k(x_i^s, x_j^s)$. We will discuss this kernel function in the next section.Then

$$VV^T = \begin{bmatrix} k(x_1^s, x_1^s) & k(x_1^s, x_2^s) & \cdots & k(x_1^s, v_n^s) \\ k(x_2^s, x_1^s) & k(x_2^s, x_2^s) & \cdots & k(x_2^s, v_n^s) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n^s, x_1^s) & k(x_n^s, x_2^s) & \cdots & k(x_n^s, x_n^s) \end{bmatrix} = K \tag{4.13}$$

4.10 now can be written as

$$w_j = V^T(K + \lambda I_N)^{-1} z_j \tag{4.14}$$

Here we define

$$\alpha \equiv (K + \lambda I_N)^{-1} z_j \tag{4.15}$$

then

$$w_j = V^T \alpha = \sum_i^N \alpha_i v_i^s \tag{4.16}$$

Now given any target marginal distribution $p(x_t)$, suppose its representation is v_t, then the corresponding j th parameter of its optimized logistic regression classifier is:

$$f_j^t(p(x_t)) = w_j^T v_t = \sum_i^N \alpha_i v_i^{sT} v_t = \sum_i^N \alpha_i k(x_i^s, x_t) \tag{4.17}$$

Since for f^t there are 25 parameters, we need to perform the above process for each parameter in f^t. It is simply a multivariate regression. Finally, we could get f^t.

4.5.2. KERNEL FUNCTION

The key to solve 4.11 and 4.17 is to find a kernel function $k(x_i, x_j)$ to replace $v_i v_j$. Here we apply a Earth Mover's Distance (EMD) based kernel used in [45, 55, 56]. Given two distributions x_i and x_j, suppose that x_i contains several signatures $\{(c_1^i, w_1^i), ..., (c_Q^i, w_Q^i)\}$, while x_j contains $\{(c_1^j, w_1^j), ..., (c_Q^j, w_Q^j)\}$, where c_q^i and c_q^j are the cluster centers of x_i and x_j respectively, and w_q^i, w_q^j are the corresponding cluster weights[45]. In our implementation, we use K-means++ [57] to find the cluster centers for each distribution. w_q is the cardinality of the cluster c_q. Now the EMD distance between x_i and x_j can be defined as:

$$\begin{aligned} D(x_i, x_j) = D_{EMD}(x_i, x_j) = \min_{f_{pq} \geq 0} \sum_{p,q=1}^{Q} d_{pq} f_{pq} \\ s.t. \quad \sum_{p=1}^{Q} f_{pq} = w_q^i \quad \sum_{q=1}^{Q} f_{pq} = w_p^j \end{aligned} \tag{4.18}$$

Here, f_{pq} is a flow variable[45], and d_{pq} is the euclidean distance between cluster center c_p^i and c_q^j defined as $d_{pq} = \|c_p^i - c_q^j\|^2$. Then we could use the kernel function:

$$k_{EMD}(x_i, x_j) = e^{-\lambda D_{EMD}(x_i, x_j)} \tag{4.19}$$

where λ is a user defined parameter and we follow [5] to set it to the average distance between all possible pairs of distributions.

4.5.3. STRATEGY 2: TWO STAGE SAMPLE RE-WEIGHTING

TPT holds the assumption that $p(y|x)$ is determined by $p(x)$, this assumption might be too strong. In the following discussion, we adopt a two-stage sample re-weighting approach for sample re-weighting [49]. This method first reweights the samples of each source domain based on their similarity to the target domain in terms of conditional distribution and marginal distribution. Second, it uses those re-weighted samples to train a classifier for the target domain.

Algorithm 2 Two-stage Sample Re-weighting

Input: Source sets D_1^s,...,D_n^s with labels and target set x^t
  for each D_i^s in D^s **do**
    Compute α_i^s by solving 4.22
    Lean a hypothesis f_i^s on the α_i^s weighted source data using 4.7
  end for
  Form the $l \times n$ prediction matrix F by apply $\{f_i\}_{i=1}^n$ to x^t
  Compute matrices W, L, D using x_t
  Compute β by solving 4.25
  Learn the classifier f^t by solving 4.26
Output: f^t

SAMPLE RE-WEIGHTING BASED ON MARGINAL DISTRIBUTION

We use Maximum Mean Discrepancy (MMD) distance to evaluate the marginal distribution distance between the source and target domain [58, 59]. Given a target domain x^t and source domain x_i^s, the distance between x^t and x_i^s is defined by:

$$\| \frac{1}{n_s} \sum_{j=1}^{n_s} \phi(x_{ij}^s) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(x_{ij}^t) \|_H^2 \tag{4.20}$$

Where n_s and n_t are the number of samples in the target and source domain. $\phi(x)$ is a feature map onto a reproducing kernel Hilbert Space H[59]. For each x_{ij}, we give it a weight α_{ij}^s to minimize 4.20. We want to solve the following optimization problem:

$$\min_{\alpha_i^s} \| \frac{1}{n_s} \sum_{j=1}^{n_s} \alpha_{ij}^s \phi(x_{ij}^s) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(x_j^t) \|_H^2 \tag{4.21}$$

Where α_i^s is a $n_s \times 1$ vector composed of the weight of each sample in the source domain. According to [60], 4.21 can be written as:

$$\begin{aligned} &\min \frac{1}{2} {\alpha_i^s}^T K \alpha_i^s - \kappa^T \alpha_i^s \\ &s.t. \quad \alpha_{ij}^s \in [0, B] and | \sum_{j=1}^{n_s} a_{ij}^s - n_s | \leq n_s \epsilon \end{aligned} \tag{4.22}$$

where K is a $n_s \times n_s$ matrix, $K_{pq} = k(x_{ip}^s, x_{iq}^s)$. And κ is a $n_s \times 1$ vector, $\kappa_p = \frac{n_s}{n_t} \sum_{q=1}^{n_t} k(x_{ip}^s, x_q^t)$,
The kernel function is defined as: $k(x_p, x_q) = e^{\frac{-\|x_p - x_q\|^2}{2\sigma^2}}$. In our experiment, we set $B = 3$

and $\epsilon = \frac{9}{\sqrt{n_s}}$, which bring the best AUC.

SAMPLE RE-WEIGHTING BASED ON CONDITIONAL DISTRIBUTION

Without knowing the labels of the target domain, computing the conditional distribution difference of target and source domain is impossible. However, Given a target domain, we hold the assumption that nearby points in the marginal distribution should have similar class labels (conditional probability).
Given a target domain, suppose there are n source domains. We define $F_j = [f_{1j}, f_{2j}, .., f_{nj}]$ as the $1 \times n$ vector of predicted labels (probability) of given by n source model f_i for the j-th sample of target domain. And $\beta = [\beta_1, \beta_2, ..., \beta_n]^T$ is a $n \times 1$ weight vector, where β_i is the weight for f_i. we want to solve the following problem:

$$\min_{\beta \geq 0, \sum \beta_i = 1} \sum (F_j \beta - F_k \beta) W_{jk} \tag{4.23}$$

where $F_j\beta$ and $F_k\beta$ are the prediction of j-th and k-th samples of target domain, while W_{jk} is the weight between sample j and k given by

$$W_{jk} = e^{\frac{-\|x_j - x_k\|^2}{2\sigma^2}} \tag{4.24}$$

we can rewrite 4.24 as:

$$\min_{\beta \geq 0, \sum \beta_i = 1} \beta^T (F)^T L_u F \beta \tag{4.25}$$

where F is an $l \times n$ matrix (l is the number of samples in the target domain). Each row of F is F_j. L_u is given by $L_u = I - D^{-0.5} W D^{-0.5}$ and D is the diagonal matrix given by $D_{jj} = \sum_{k=1}^{l} W_{jk}$.

Now for a domain $\{x_i, y_i\}$ in train set, each sample $\{x_{ij}, y_{ij}\}$ has its weight for conditional distribution β_i and weight for marginal distribution α_{ij}. The loss function 4.7 of our personalized model change to:

$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{j=1}^{m} \log \beta_i \alpha_{ij} (\exp(-y_{ij}(x_{ij}^T w + c)) + 1) \tag{4.26}$$

5

Result Analysis and Discussion

5.1. LOGISTIC REGRESSION VS TPT

In this section, we compare the performance of the simple logistic regression and TPT (stat of the art). We apply a leave-10-subjects-out cross-validation for performance evaluation as mentioned in 2. The hand-crafted features we adopted are the same as [5].

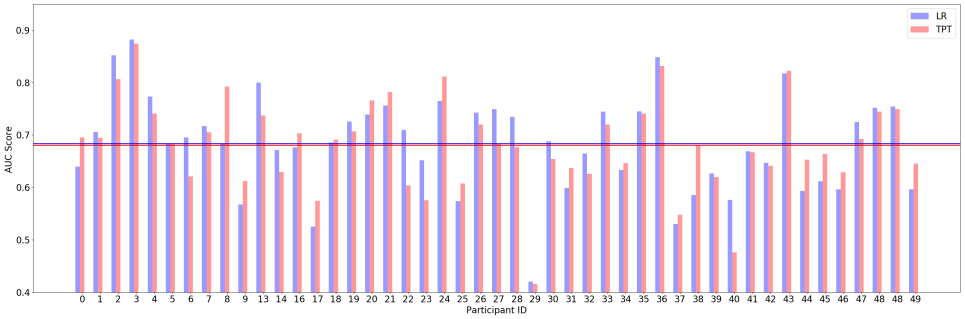


Figure 5.1: LR vs TPT

	AUC Performance
LR	0.682 (std = 0.091)
TPT	0.680 (std = 0.087)

Table 5.1: LR vs TPT

As shown in the above tables, the performance of TPT is almost the same as the performance of the simple logistic regression. Unlike the result in [5], TPT lost its power when a longer recording time is adopted (30 mins vs 10 mins) and more participants are included (50 vs 18). The following figures show the PCA analysis of the first 4 people in fold 0.

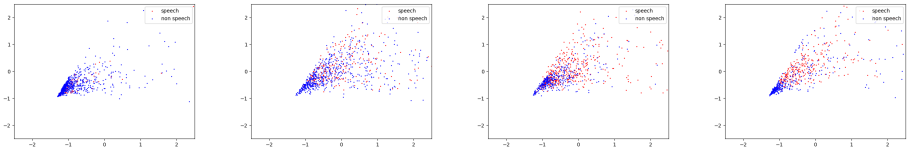


Figure 5.2: PCA Analysis for person 0,1,2,3 (hand-crafted feature)

As shown in figure 5.2, the distributions of different subjects are quite similar. The basic assumption of TPT is that, the distributions of different source domains are different. However, with a longer recording time and more participants, the marginal distribution difference seems to be disappeared. When the recording time is limited, for example, 10 minutes as mentioned in [5], the data we collected for each person might just be distributed at a subarea of the entire distribution. This type of sampling bias causes the

data distribution difference between different subjects, thus when data is collected in a small range, TPT is valid. When a longer recording time is time is adopted and more subjects are included, the data we collect from each person is enough to represent the entire distribution. The distribution difference between people thus disappears and TPT no longer works. However, this situation only happens with hand-crafted features. From 5.4 we could see the feature distribution difference. For the features extracted by our neural network model, the person-specificity still exists.

5.2. LOGISTIC REGRESSION VS NEURAL NETWORK

Since the TPT and LR achieved the same AUC performance in the above discussion, we still use LR as the baseline to compare with our neural network model.

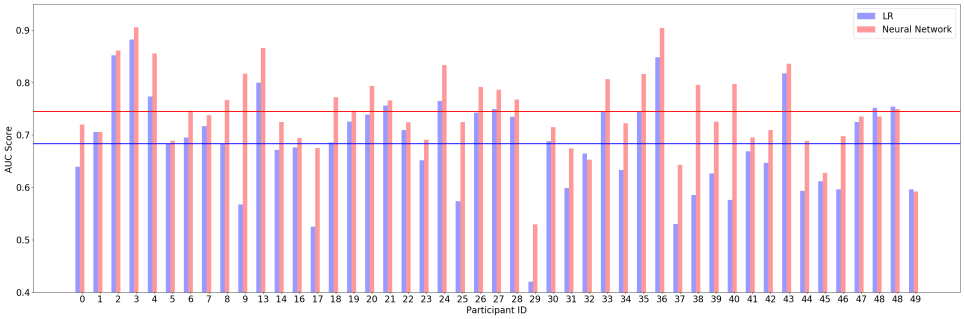


Figure 5.3: LR vs Neural Network

	AUC Performance
LR	0.682 (std = 0.091)
Neural Network	0.743 (std = 0.075)

Table 5.2: LR vs Neural Network

Compared with hand-crafted features based approaches, our neural network model brings an AUC improvement for 6%. As shown in 5.4, the AUC varies between different subjects. For person 3, the AUC can achieve 90%, while for person 29, the AUC is even less than 60 %.

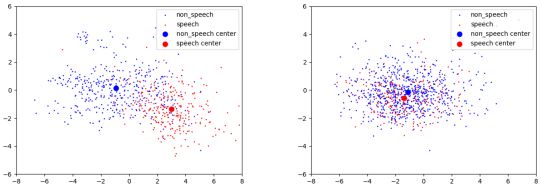


Figure 5.4: PCA Analysis for person 3, 29

We draw the PCA analysis for persons 3 and 29. For person 3, the distributions of speech and non-speech are separable, while for person 29, the distributions of speech and non-speech are mixed up. This indicates that the connection between speech and body movement changes between different subjects. For some people, this relation is strong and speech can be easily detected through the accelerometer data, while for the other person, such connection is rather implicit.

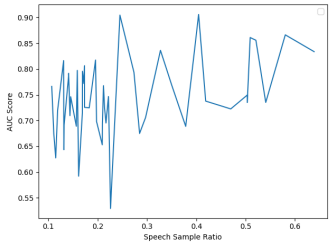


Figure 5.5: AUC Score with Speech Ratio

We also draw the speech sample Ratio-AUC curve 5.5. People with a higher speech sample ratio tend to have a higher AUC. However, this connection is weak, the person with quite a low speech sample ratio can still achieve an acceptable AUC.

5.3. NEURAL NETWORK BASED TPT

Now we combine TPT with the features extracted by the neural work. We first extracted features from the raw data using the neural network, then we applied TPT on the new neural network based features. Figure 5.6 shows the performance for neural network based TPT per person.

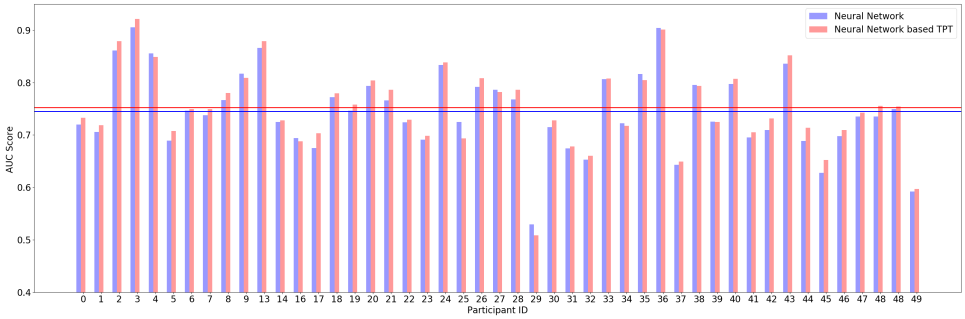


Figure 5.6: Neural Network vs Neural Network based TPT

By applying TPT with the neural network, we slightly improve the average AUC performance by 1%. We also did a related t-test between the performance of these two methods. Usually, a p-value smaller than 0.05 indicates a distribution difference (performance improvement). In our case, we got a p-value smaller than 0.01, which means TPT does improve the overall performance.

	AUC Performance
Neural Network	0.743 (std = 0.075)
Neural Network based TPT	0.752 (std = 0.076)

Table 5.3: Neural Network vs Neural Network based TPT

5.4. INFLUENCE OF DATA SIZE

We are also interested in how the size of our dataset will influence the performance of our neural network. We want to know whether the size of the data set we currently have is enough for a well-trained model. As has discussed in 2, averagely, each person will have 875 raw frames, each raw frame lasts for 3 seconds. In this experiment, we still follow the leave-10-subjects out cross-validation set up. We first only made of 2% of each person's data in the train set in the first turn, then for every following round, added another 2% of each person's data into the former round train set, trained the model again and test on the person in the test set. 5.7 shows the learning curve of our neural network and neural

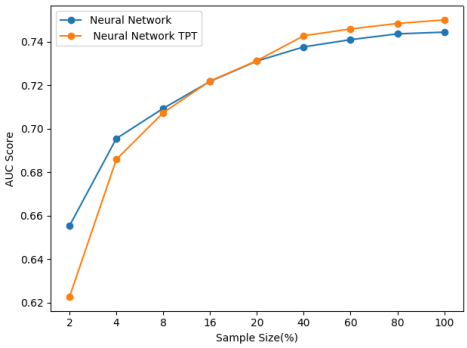


Figure 5.7: Learning Curve of Neural Network and Neural Network based TPT

network based TPT approach. When we only make use of 8 %(70 frames, 105 seconds) of data per person, both models can already achieve an AUC performance for nearly 71 %. Also, when we don't have enough for each person (less than 8 %), the TPT method declines the performance of the original neural network. With the sample size grows, the performance of the neural network TPT gradually outperforms the neural network model.

When each individual's data is limited, for example, in the first round, only 2 % of each person's data is used. In this case, when applying TPT, we only have 17 frames to train an individual LR for each participant in the train set and LR can be easily over-fitted. A TPT algorithm based on those over-fitted models would definitely decline the performance. When the size of the data from each subject grows (e.g. 16 %, 140 frames each). The data we have is enough for training individual model and the map between the data distribution and the decision boundary each can be precisely established. This is the reason why when more data per person is used, TPT outperforms single neural networks.

Furthermore, the rising trend of the AUC curve is slowing down when more data is added. The curve has almost achieved the saturation point when 100 % of data is used. In other words, if we used the data from all the participants, the data we currently have is enough to train a general model.

5.5. POSSIBLE UPPER BOUND FOR NEUTRAL NETWORK BASED TPT

TPT holds the assumption that the decision boundary of each person's data is determined by its marginal distribution. However, even two data distribution are the same, their decision boundaries can still be different. In other words, we may need to consider both conditional and marginal distributions. However, without knowing the labels of the target person, we can never get access to the conditional distribution of data. To check the upper bound of TPT, in this experiment, we make use of the label of the test set. Instead of computing the distance between two persons:

$$D(x_i, x_j) = D_{EMD}(x_i, x_j) \tag{5.1}$$

Here we use:

$$D(x_i, x_j) = D_{EMD}(x_{ip}, x_{jp}) + D_{EMD}(x_{in}, x_{jn}) \tag{5.2}$$

where x_{ip}, x_{jp} are the positive (speech) sample distributions and x_{in}, x_{jn} are the negative (non-speech) sample distributions. By computing the distance between speech and non-speech distributions separately, we approximately take both conditional and marginal distribution into consideration.

	AUC Performance
Neural Network	0.743 (std = 0.075)
Neural Network based TPT	0.752 (std = 0.076)
Neural Network based TPT(known test set label)	0.757 (std =0.075)

Table 5.4: Performance in terms of AUC

As shown in the above table, even with the known label and the new distance metric, our TPT approach can only bring an AUC improvement of 1.5%.

5.6. TWO-STAGE SAMPLE RE-WEIGHTING

Next, we changed our strategy by re-weighting the samples in the train set based on their conditional and marginal similarity to the target person in the test set. the result is shown in figure 5.8:

Compared with TPT, our two-stage sample re-weighting performs slightly worse at than the original neural network model. We also did a related t-test here and the corresponding p-value was 0.15 which was much higher than 0.05. This indicated that the sample

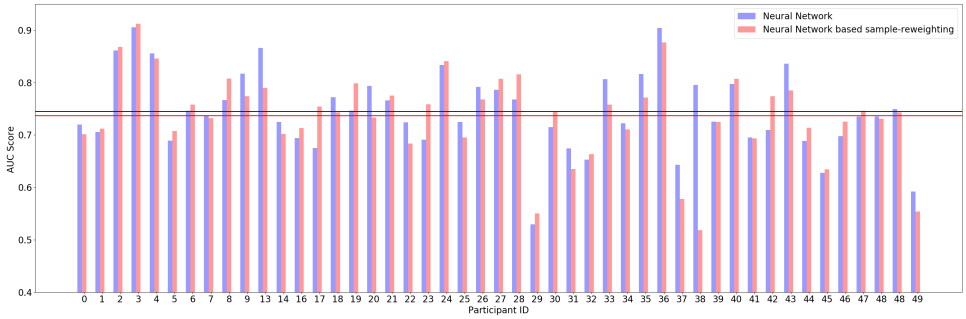


Figure 5.8: Neural Network vs Neural Network based Sample Re-weighting

	AUC Performance
Neural Network	0.743 (std = 0.075)
Neural Network based Sample Re-weighting	0.736 (std = 0.081)

Table 5.5: Neural Network vs Neural Network based Sample Re-weighting

re-weighting approach did not bring a performance improvement. The two-stage sample re-weighting approach holds the assumption that for any distribution, two nearby samples should have the similar prediction value. However, for most of the subjects included in our dataset, their speech and non-speech sample distributions are highly overlapped.

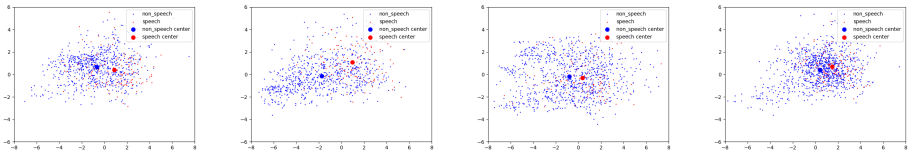


Figure 5.9: PCA Analysis for person 5, 6, 8, 45 in fold0

The figure 5.9 shows the sample distributions of 4 subjects in fold 0. Their speech and non-speech sample distributions are highly overlapped. Two close by samples usually have different labels. Such situation violate the basic assumptions(close points have the same label) of our re-weighting approach. This is main reason why this method does not work.

5.7. THE EXISTENCE OF PERSON-SPECIFICITY

All our effort trying to get a personalized model did not achieve a satisfying result. Although we have observed from the PCA analysis that data distributions of different subjects are different. However, this type of difference is still not significant enough for EMD or MMD to precisely recognize it. Furthermore, at the beginning of our research, we hold

the assumption that each subject's data distribution is different. However, when we are training the neural network model, we fed it with the data collected from multiple subjects. During the training process, the model was forced to learn some individual invariant features and lose person specificity.

In this section, we prove the existence of person specificity. We made use of a simple GRU model (8 units). We selected the subjects in fold 0 as the test set and the remain as the train set. First, we trained our model on the data in the train set. Second, for each subject in the test set, we divided the data into fine-tuning train set and validation set(50% : 50 %), checking whether fine-tuning the model with personal data can improve our model performance for each person. However, according to [61], the train set can be contaminated if neighbor sliding window frames are contained in train and test set separately. To avoid this problem, we implemented the meta-segmentation[61] algorithm to divide each person's data into the train and validation set this time. The detail for meta-segmentation is shown below:

Algorithm 3 Meta-segmentation[61]

Input: frameLabels, numFolds, seglength
classes = **unique**(frameLabels)
numSegments = **framLabels**/seglength
segDist = **zeros**(numSegments, **length**(classes))
for i = 1 to numSegments **do**
 # get distribution of labels within each segment
segDist(i,:) = **getLabelDist**(frameLabels ∈ segment $_i$)
 # add noise for randomness
segDist(i,:) = segDist(i,:) + **randn** *0.1
end for
 # get sorted list of indices (lexicographic sort of distributions)
indices = **lexisort**(segDist)
 # assign fold to each segment
foldIds(indices) = 1 + **mod**(1...numSegments,numFolds)
for i = 1 to numSegments **do**
 # assign each frame within segment to fold
frameFoldIds ∈ segment $_i$ = foldIds(i)
end for
Output: frameFoldIds

In this way, we avoided the data contamination problem. We repeated the experiment for 10 times, making sure the performance is trustworthy. The result are shown below:

	AUC Performance
Before fine tuning	0.723 (std = 0.089)
After fine tuning	0.773 (std = 0.076)

Table 5.6: Performance in terms of AUC

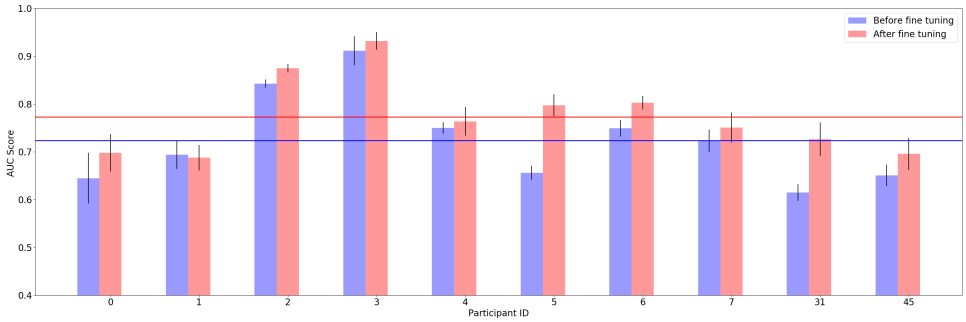


Figure 5.10: The affect of fine tuning

As shown in table 5.6, after fine-tuning, the average AUC increases by 5%. Since for each person, we only have around 400 frames for fine-tuning, we believe the performance of the fine-tuned model can even be better if more data is given.

6

Conclusion and Future Work

6.1. CONCLUSION

Our research focused on detecting speech from body movements. We used a single tri-axial accelerometer worn around the chest to collect the body movement information. Through the entire research process, we held the assumption that speech is highly personal-specific, which means the data distributions of different subjects are different. Compared with previous research focusing on speech detection from accelerometers [5, 44], instead of using hand-crafted features, our works used neural networks to extract features from the raw accelerometer automatically. The CNN+GRU model applied in this research brings a 6 % AUC improvement compared to the state of art [5]. After that, we visualized the features extracted by the neural network, then found out that, for each person, their data distributions were different. This discovery complied with our initial hypothesis of person-specificity. As follow-up research, we adopted two approaches of multi-source domain adaption based on the features extracted from the neural network, aiming to get a personalized model for each individual. We first applied TPT and brought an AUC improvement by 1 % (74% vs 75 %). Another strategy based on sample re-weighting is also tested, however, this method did not bring a satisfying result.

Based on the experiments we conducted, here, we answer the research questions we proposed at the beginning:

1. How will the TPT perform on a larger data set(50 subjects, 30 minutes recording for each person)?

   For hand-crafted features, when a larger data set is adopted, the feature distribution difference between subjects no longer exists, TPT is not valid in this case. When the recording time is limited, the data we collected for each person might just be distributed at a subarea of the entire distribution. This type of sampling bias causes the data distribution difference between different subjects, thus when data is collected in a small range, TPT is valid.

2. Whether the person-specificity assumption still holds for the features extracted by a neural network (different distributions)? If so, can we combine the neural network and TPT (applying TPT to the features extracted by the neural network) to get a better performance per person?

   Yes, by applying neural network, we improve the AUC performance by 6 %. Also, through the PCA analysis, we find that the feature distributions of different persons are different. By applying TPT, we further improve the performance by 1%.

3. TPT holds the assumption that the optimized personalized decision boundary (conditional distribution) is determined by the marginal distributions, is this assumption true and what is the possible way to adapt TPT to get a better performance? What is the upper performance bound for TPT?

   TPT only measures the distance between two distributions based on their marginal distributions. However in experiment 5.5, when we compute the distance between

the speech and non-speech samples separately, we could get a better result (0.752 for TPT, 0.757 for TPT with known test set label). This phenomenon indicated that when applying TPT, if some samples' labels in the test are known, those samples can be used to calculate the distance between the source and target domain as mentioned in 5.2. In this way, TPT can get better performance.

4. In most cases, neural networks are data-hungry. How much data is enough for a well-trained model for speech detection? Will we get better performance if more data is provided?

   According to 5.7, when more data is added, the learning curve for our CNN + GRU model gradually reaches to the saturation point. This indicates that, if we want to get a well trained general model that trained with data from multiple subjects, the data amount we have for now is enough. Thus, if we want to improve the performance, instead of adopting a longer recording time, a better choice would be using more accelerometers. Accelerometers can also be placed at body positions like forearms and waists to get more body movement information. However, if we want to get a pure personalized model, we still need more data per person.

5. Compared to the TPT, the second approach we proposed (two-stage sample re-weighting) does not hold any assumption about data distributions. It re-weights the samples from source domain persons based on their conditional and marginal distribution similarity to the target person (subjects in the test set). Will combing the neural network with two-stage sample re-weighting bring a better performance?

   The re-weighting method we adopted does not achieve a good result. The basic assumption of this approach is that: two close-by samples need to have the same labels. However, for our research, for most of the persons, the speech and non-speech samples are heavily overlapped. Two close-by points usually have different labels. The basic assumption of the re-weighting approach does not hold there.

6. Does the assumption of person-specificity really hold in speech detection in our experimental set-up? Do the personalized neural network model (trained with data from the target person) really outperform the general model that trained on the data from all the other subjects?

   Yes, according to 5.10, the model fine-tuned with individual data performs better than a general model (0.77 after fine-tuning : 0.72 before fine-tuning). Furthermore, for each person, we only have around 400 frames for fine-tuning, if we have more personal data, the difference between the personalized model and the general model can be larger. However, when we combine TPT with neural networks, we only get 1 % improvement, this is too small. A possible explanation is that, when using accelerometer data from multiple subjects to train our model, the model is forced to learn some person-invariant feature and thus the person-specificity is lost. A possible solution to this issue will be discussed in the next section.

6.2. FUTURE WORK

As have discussed in the former section, training a neural network model with data collected from multiple subjects might force the model to learn some individual invariant features and lose person-specificity. A possible structure is shown below:

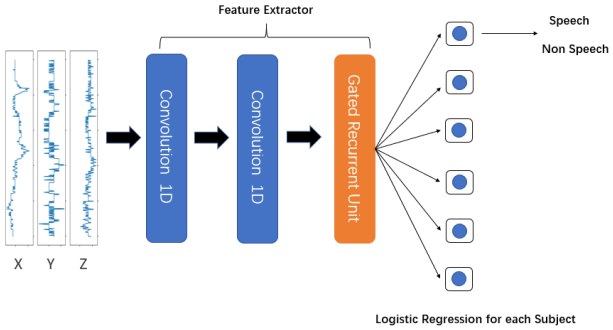


Figure 6.1: A Possible Structure for Feature Extraction

In this structure, instead of using a universal logistic regression layer, we give each person in the train set an individual logistic regression. In other words, we treat the classifications of different persons' data as different tasks (multi-task learning) [62]. In this way, we compress the original raw data to 1 d vectors while the final distributions of different subjects don't necessarily to be the same. Then we could combine this structure with TPT or the sample re-weighting method to get a personalized model.

Another possible direction to improve the model performance might be source domain selection. As shown in 5.2, for some persons like person 3, there is a strong connection between the speech and body movement, the speech and non-speech samples can be easily separated. However for person 29, this connections is fairly implicit (with an AUC of 0.54). The data from such a person might have a negative effect when training the model (negative transferring)[63]. If we could delete such a person's data from the train set, our model might get better performance. Furthermore, for now, our research purely focuses on how to process the data we have. Since the participants come from different backgrounds. Some external information might be helpful when we select the source domains. For example, given a person's data, we could select the source domains that have the same age, gender and nationality. People who have the similar background might share similar behavioural habits, thus have similar data distributions.

Based on the learning curve figure 5.7, we could draw the conclusion that, under current the experiment set-up, the data amount we have is enough to train a well-trained neural network model using data come from multiple subjects. However, the performance of our model is still quite low(74%). For now, since the learning curve of our model has already reached a saturation point. If we want to get a general model with better performance based on the data from multiple subjects, it's not wise to adopt a longer recording

time.

Through our research, we have proven that there is a connection between speech and body movements. Then, a further question would be: how strong this connection could be? What is the upper bound performance for detecting speech from body movements? With our neutral network + TPT model, we get an AUC of 75 %. If we could use more sensors placed at different positions of each person, then we can capture the body movements more precisely. A model trained with multi-modality information may easily break the current upper bound.

We have proven the existence of person-specificity through our experiment of fine-tuning model. However, the fine-tuned model only gets an AUC improvement for 5 % averagely. For our data set, for each person, we only have 875 data frames. The data we have per person is not enough for training a neural network. For now, we can not figure out what is the upper bound of a personalized model trained purely with each individual's data. In future research, we need a much longer recording time for each individual (e.g 3 or 4 hours). Only in this way, we can get a well-trained neural network model and compare the difference between the personalized model and the general model.

ACKNOWLEDGEMENTS

Throughout the writing of this dissertation I have received a great deal of support and assistance.

I would like to thank my supervisor Prof. H. Hung and Dr. E. Gedik, whose expertise was invaluable in the formulating of the research topic and methodology in particular. Thanks for your great patience and supportive suggestions.

REFERENCES

[1] Alessandro Vinciarelli, Maja Pantic, Dirk Heylen, Catherine Pelachaud, Isabella Poggi, Francesca D'Errico, and Marc Schröder. Bridging the gap between social animal and unsocial machine: A survey of social signal processing, 2012.

[2] Tanya L. Chartrand and John A. Bargh. The chameleon effect: The perception-behavior link and social interaction. *Journal of Personality and Social Psychology*, 1999.

[3] Nikhil Dandekar Nishkam Ravi Preetham Mysore, Michael L. Littman. Activity recognition from accelerometer data. *The Seventeenth Innovative Applications of Artificial Intelligence Conference on Artificial Intelligence (IAAI-05)*, 2005.

[4] Francisco Javier Ordóñez Morales and Daniel Roggen. Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations. 2016.

[5] Ekin Gedik and Hayley Hung. Personalised models for speech detection from body movements using transductive parameter transfer. *Personal and Ubiquitous Computing*, 21(4):723–737, 8 2017.

[6] Shiliang Sun, Honglei Shi, and Yuanbin Wu. A survey of multi-source domain adaptation. *Information Fusion*, 2015.

[7] J. Ramirez, J. M., and J. C. Voice Activity Detection. Fundamentals and Speech Recognition System Robustness. In *Robust Speech Recognition and Understanding*. 2012.

[8] Deepti Singh and Frank Boland. Voice activity detection. *Crossroads*, 2009.

[9] Alex Graves. Generating Sequences With Recurrent Neural Networks. 2013.

[10] A. Vinciarelli, H. Salamin, and M. Pantic. Social signal processing: Understanding social interactions through nonverbal behavior analysis. In *2009 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009*, 2009.

[11] Alessandro Vinciarelli, Maja Pantic, and Hervé Bourlard. Social signal processing: Survey of an emerging domain. *Image and Vision Computing*, 2009.

[12] Rutger Rienks and Dirk Heylen. Dominance detection in meetings using easily obtainable features. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2006.

[13] Rutger Rienks, Dong Zhang, Daniel Gatica-Perez, and Wilfried Post. Detection and application of influence rankings in small group meetings. 2007.

[14] Oded Ghitza. Modern Methods of Speech Processing. *Speech and Audio Processing, IEEE Transactions on*, 2:115–132, 1994.

[15] Thomas Plötz, Nils Y. Hammerla, and Patrick Olivier. Feature learning for activity recognition in ubiquitous computing. In *IJCAI International Joint Conference on Artificial Intelligence*, 2011.

[16] Nils Y. Hammerla, Shane Halloran, and Thomas Plötz. Deep, convolutional, and recurrent models for human activity recognition using wearables. In *IJCAI International Joint Conference on Artificial Intelligence*, 2016.

[17] Stephen J. Preece, John Yannis Goulermas, Laurence P J Kenney, and David Howard. A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. *IEEE Transactions on Biomedical Engineering*, 2009.

[18] Sherman Wilcox and Sherman Wilcox. Language and Gesture. In *Ten Lectures on Cognitive Linguistics and the Unification of Spoken and Signed Languages*. 2017.

[19] Hayley Hung, Ekin Gedik, and Laura Cabrera Quiros. Complex conversational scene analysis using wearable sensors. In *Multimodal Behavior Analysis in the Wild*. 2018.

[20] Hayley Hung, Gwenn Englebienne, and Jeroen Kools. Classifying social actions with a single accelerometer. 2013.

[21] Jindong Wang, Vincent W Zheng, Yiqiang Chen, and Meiyu Huang. Deep Transfer Learning for Cross-domain Activity Recognition. In *Proceedings of the 3rd International Conference on Crowd Science and Engineering*, ICCSE'18, pages 16:1–16:8, New York, NY, USA, 2018. ACM.

[22] Renjie Ding, Xue Li, Lanshun Nie, Jiazhen Li, Xiandong Si, Dianhui Chu, Guozhong Liu, and Dechen Zhan. Empirical study and improvement on deep transfer learning for human activity recognition. *Sensors (Switzerland)*, 2019.

[23] Qian Sun, Rita Chattopadhyay, Sethuraman Panchanathan, and Jieping Ye. A Two-Stage Weighting Framework for Multi-Source Domain Adaptation. *NIPS*, 2011.

[24] Shian-Ru Ke, Hoang Thuc, Yong-Jin Lee, Jenq-Neng Hwang, Jang-Hee Yoo, and Kyoung-Ho Choi. A Review on Video-Based Human Activity Recognition. *Computers*, 2013.

[25] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.

[26] Yan Ke, Rahul Sukthankar, and Martial Hebert. Spatio-temporal shape and flow correlation for action recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[27] Sonal Kumari and Suman K. Mitra. Human action recognition using DFT. In *Proceedings - 2011 3rd National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics, NCVPRIPG 2011*, 2011.

[28] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. 2007.

[29] Wei Lwun Lu and James J. Little. Simultaneous tracking and action recognition using the PCA-HOG descriptor. In *Third Canadian Conference on Computer and Robot Vision, CRV 2006*, 2006.

[30] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3D skeletons as points in a lie group. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014.

[31] Meinard Müller, Tido Röder, and Michael Clausen. Efficient content-based retrieval of motion capture data. *ACM Transactions on Graphics*, 2005.

[32] Ling Bao and Stephen S. Intille. Activity Recognition from User-Annotated Acceleration Data. 2004.

[33] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, 2019.

[34] Ming Zeng, Le T. Nguyen, Bo Yu, Ole J. Mengshoel, Jiang Zhu, Pang Wu, and Joy Zhang. Convolutional Neural Networks for Human Activity Recognition using Mobile Sensors. 2014.

[35] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, Jakob Doppler, Clemens Holzmann, Marc Kurz, Gerald Holl, Ricardo Chavarriaga, Hesam Sagha, Hamidreza Bayati, Marco Creatura, and Jose Del R. Millàn. Collecting complex activity datasets in highly rich networked sensor environments. In *INSS 2010 - 7th International Conference on Networked Sensing Systems*, 2010.

[36] Piero Zappi, Clemens Lombriser, Thomas Stiefmeier, Elisabetta Farella, Daniel Roggen, Luca Benini, and Gerhard Tröster. Activity Recognition from On-Body Sensors: Accuracy-Power Trade-Off by Dynamic Sensor Selection. In Roberto Verdone, editor, *Wireless Sensor Networks*, pages 17–33, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[37] Felix Gers. Long short-term memory in recurrent neural networks. *Neural Computation*, 2001.

[38] Yu Guan and Thomas Plötz. Ensembles of Deep LSTM Learners for Activity Recognition using Wearables. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2017.

[39] Jian Bo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiao Li Li, and Shonali Krishnaswamy. Deep convolutional neural networks on multichannel time series for human activity recognition. In *IJCAI International Joint Conference on Artificial Intelligence*, 2015.

[40] Yuqing Chen and Yang Xue. A Deep Learning Approach to Human Activity Recognition Based on Single Accelerometer. In *Proceedings - 2015 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015*, 2016.

[41] Abdulmajid Murad and Jae Young Pyun. Deep recurrent neural networks for human activity recognition. *Sensors (Switzerland)*, 2017.

[42] Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. In *Proceedings - International Symposium on Wearable Computers, ISWC*, 2012.

[43] Francisco Javier Ordóñez and Daniel Roggen. Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. *Sensors (Switzerland)*, 2016.

[44] Aleksandar Matic, Venet Osmani, and Oscar Mayora. Speech activity detection using accelerometer. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2012.

[45] Enver Sangineto, Gloria Zen, Elisa Ricci, and Nicu Sebe. We are not All Equal: Personalizing Models for Facial Expression Analysis with Transductive Parameter Transfer. In *ACM Multimedia*, 2014.

[46] Wouter M Kouw. An introduction to domain adaptation and transfer learning. *CoRR*, abs/1812.1, 2018.

[47] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S. Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[48] Niall Adams. Dataset Shift in Machine Learning. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 2009.

[49] Rita Chattopadhyay, Qian Sun, Wei Fan, Ian Davidson, Sethuraman Panchanathan, and Jieping Ye. Multisource domain adaptation and its application to early detection of fatigue. *ACM Transactions on Knowledge Discovery from Data*, 2013.

[50] Laura Cabrera-Quiros, Andrew Demetriou, Ekin Gedik, Leander van der Meij, and Hayley Hung. The MatchNMingle dataset: a novel multi-sensor resource for the analysis of social interactions and group dynamics in-the-wild during free-standing conversations and speed dates. *IEEE Transactions on Affective Computing*, 2018.

[51] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference On Artificial Intelligence and Statistics*, 2010.

[52] Yoshua Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *Journal of Biological Chemistry*, 2006.

[53] Sebastian Ruder. An overview of gradient descent optimization algorithms. *CoRR*, abs/1609.0, 2016.

[54] Christian Robert. Machine Learning, a Probabilistic Perspective . *CHANCE*, 2015.

[55] Elisa Ricci, Gloria Zen, Nicu Sebe, and Stefano Messelodi. A prototype learning framework using EMD: Application to complex scenes analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[56] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. Earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 2000.

[57] David Arthur and Sergei Vassilvitskii. K-Means++: the Advantages of Careful Seeding. In *Proc ACM-SIAM symposium on discrete algorithms.*, 2007.

[58] Karsten M. Borgwardt, Arthur Gretton, Malte J. Rasch, Hans Peter Kriegel, Bernhard Schölkopf, and Alex J. Smola. Integrating structured biological data by Kernel Maximum Mean Discrepancy. In *Bioinformatics*, 2006.

[59] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander J Smola. A Kernel Method for the Two-Sample Problem. *CoRR*, abs/0805.2, 2008.

[60] Correcting Sample Selection Bias by Unlabeled Data. In *Advances in Neural Information Processing Systems 19*. 2018.

[61] Nils Y. Hammerla and Thomas Ploetz. Let's (not) Stick Together: Pairwise Similarity Biases Cross-Validation in Activity Recognition. In *Proc. Ubicomp*, 2015.

[62] Azad Naik and Huzefa Rangwala. Multi-task Learning. In *SpringerBriefs in Computer Science*. 2018.

[63] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning, 2010.