A Thesis

entitled

Unsupervised Motion Artifact Detection in Wrist-Measured Electrodermal Activity Data

by

Yuning Zhang

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the Masters of Science Degree in Electrical Engineering

Dr. Kevin S. Xu, Committee Chair

Dr. Mansoor Alam, Committee Member

Dr. Scott Pappada, Committee Member

Dr. Amanda Bryant-Friedrich, Dean
College of Graduate Studies

The University of Toledo
August 2017

An Abstract of

Unsupervised Motion Artifact Detection in Wrist-Measured Electrodermal Activity Data

by

Yuning Zhang

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the Masters of Science Degree in Electrical Engineering

The University of Toledo
August 2017

One of the main benefits of a wrist-worn computer compared to other computing platforms is its ability to collect a variety of physiological data in a minimally intrusive manner. Among these physiological data, electrodermal activity (EDA) is readily collected and provides a window into a person's emotional and sympathetic responses. Unfortunately, EDA data collected using a wearable wristband are easily influenced by motion artifacts (MAs) that may significantly distort the data and degrade the quality of analyses performed on the data if not identified and removed. Prior work has demonstrated that MAs can be successfully detected using supervised machine learning algorithms on a small data set collected in a lab setting. In this thesis, we demonstrate that unsupervised learning algorithms perform competitively and sometimes *even better* than supervised algorithms for detecting MAs on EDA data collected in both a lab-based and a real-world data set comprising about 23 hours of data. We also find, somewhat surprisingly, that accelerometer data do not appear to be very useful in detecting MAs in EDA, incorporating accelerometer data as well as EDA improves detection accuracy only slightly for supervised algorithms and significantly degrades the accuracy of unsupervised algorithms

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my advisor, Dr. Kevin S. Xu, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my master's studies, and I would not possibly complete this thesis and publish the paper without his continuous support!

I am also grateful to the other two members of my committee, Dr. Mansoor Alam and Dr. Scott Pappada, for their support in overcoming numerous obstacles I have been facing through my master's studies and for the advises on modifying this thesis.

I thank my fellow lab-mates in IDEAS lab Rehan, Ruthwik, Abhishek, Brian, Maysam, and Makan, for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last two and half years!

I would also like to thank the two labeling experts, Abhishek and Maysam, who were involved in labeling the data for the ISWC paper and my thesis. Without their passionate participation and input, the research and the paper could not have been successfully conducted.

Last but not the least, I would like to thank my parents: my mom, Ying Xu, and my dad, Xiaojie Zhang, for supporting me spiritually throughout writing this thesis and my life in general!

Contents

List of Tables

List of Figures

List of Abbreviations

1-class SVMs	One-class Support Vector Machines
ACC	Acceleration
ANOVA	Analysis of variance
AUC	Area Under the Curve
AWW	Alan Walks Wales data
CV	Cross-validation
ECG	Electrocardiogram
EDA	Electrodermal activity
EDR	Electrodermal response
EEG	Electroencephalography
EMG	Electromyography
FFT	Fast Fourier Transform
FWER	Family-wise error rate
GSR	Galvanic skin response
JOSS	JOint Sparse Spectrum reconstruction algorithm
kNN	k-nearest neighbor
MAs	Motion artifacts
MLP	Multi-layer Perceptron
NLMS	Normalized Least Mean Square
PGR	Psychogalvanic reflex
PPG	Photoplethysmogram

ROC	Receiver Operator Characteristic
SC	Skin Conductance
SCL	Skin conductance level
SCRs	Skin conductance responses
SSR	Sympathetic skin response
SVMs	Support vector machines
SWT	Stationary wavelet transform
TROIKA	signal decomposiTion, sparse signal RecOnstructIon, and spectral peaK trAcking algorithm
UTD	UT Dallas stress data

List of Symbols

μ	the mean of the population
σ	the standard deviation of the population
μS	the unit of the skin conductance, micro Siemens
Mag	the magnitude of the acceleration
z	the standard score, which is the signed number of standard deviations by which the value of an observation or data point is above the mean value of what is being observed or measured
a_x	the x axis of the acceleration
a_y	the y axis of the acceleration
a_z	the z axis of the acceleration
α	the L2 penalty parameter of the multi-layer perceptron
γ	the kernel width of the support vector machines
ν	an upper bound on the fraction of margin errors and a lower bound of the fraction of support vectors relative to the total number of training examples in the one-class support vector machines

Chapter 1

Introduction

With the increasing popularity of wearable computers on the wrist, including fitness bands and smart watches, there has been tremendous interest in analyzing data collected from these wearables, particularly physiological data. The physiological data of a person, such as electrocardiogram(ECG), electromyography(EMG), electroencephalography(EEG), or electrodermal activity(EDA), provides a huge source of implicit information that can be used to monitor the overall condition of a person, or infer a person's reactions to various circumstances. One of these physiological data that is readily measured by a wearable wristband and reflects the emotional and sympathetic responses of a person is the electrodermal activity(EDA) [1]. EDA has been used in many applications including content valence classification [2], stress detection [3, 4], and classifying autonomic nervous system activity [5]. Silveira et al. use the EDA responses of viewers watching video content to accurately predict and classify the explicit feedback of the viewer to the feature films [2]. Hernandez et al. apply two methods to automatically discriminate the stressful/non-stressful calls by analyzing the skin conductance (SC) of nine call center employees during their regular work[3]. Natarajan, Xu, and Eriksson successfully distinguish between sympathetic and parasympathetic nervous system responses using both photoplethysmogram(PPG) data and EDA collected from wearables [5].

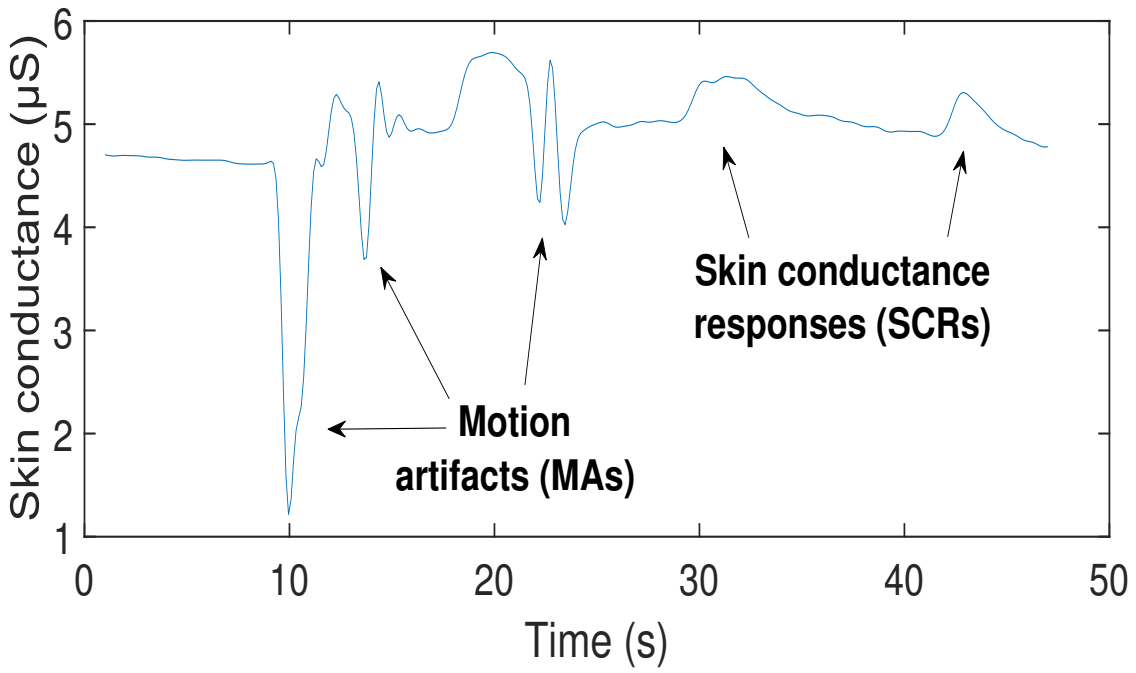

Figure 1-1: Examples of SCRs compared to MAs. A sudden increase in SC may be indicative of the start of an SCR or an MA.

EDA is commonly measured via the skin conductance (SC). When a person is under stress or at a high level of emotion, the sympathetic nervous system is activated and causes the person to sweat, increasing the SC in a series of skin conductance responses (SCRs) where the SC rapidly increases then gradually decays.

EDA data has traditionally been collected using stationary equipment in a laboratory setting, such as the Biosemi ActiveTwo used to collect the DEAP data set [6]; however, recent wearable wristbands, such as the Affectiva Q sensor [7], offer the ability to non-invasively measure EDA in real-world environments. The Affectiva Q Sensor was the world's first comfortably wearable sensor that can scientifically and accurately measures the EDA, it captures the electrical conductance across the skin by passing a miniature amount of current between two electrodes in contact with the human skin. The Affectiva Q Sensor also measures skin temperature and collects 3-axis accelerometer data [8]. A sample look of Affectiva Q Sensor is presented in

Figure 1-2: The wrist worn Affectiva Q Sensor.

Figure 1-2.

One of the main challenges when analyzing EDA data collected from such wearables is the presence of motion artifacts (MAs) in EDA data. Such artifacts may result from changes in the amount of pressure on the sensor or movements or rotations of the wrist that affects the amount of contact between the electrodes and the skin. Some examples of SCRs and MAs are shown in Figure 1-1.

Many analyses of EDA data consider features such as the mean and standard deviation of SC over a time window [5], as well as the number of peaks within the time window. If MAs are present during this time window, such features can be significantly affected by the MAs and lead to erroneous results. Thus, it is important to automatically detect segments of EDA where MAs are present.

To the best of our knowledge, prior work on suppressing and detecting MAs in EDA has not taken advantage of data collected from an accelerometer, which is also typically present on wearable wristbands. Accelerometer data have been shown to be extremely valuable for suppressing MAs in photoplethysmogram (PPG) data for heart rate estimation [9].

In this thesis, we apply eight different machine learning algorithms, five supervised and three unsupervised, to two EDA data sets comprising about 23 hours of data in both lab and real-world settings, to automatically detect MAs in EDA. We also evaluate the usefulness of accelerometer data for improving MA detection. Our main findings are as follows:

- The accuracy of unsupervised learning algorithms is competitive with that of the supervised algorithms for out-of-sample prediction (when training and testing on different data sets), and the accuracy is *even higher* than that of the supervised learning algorithms for in-sample prediction (within a particular data set).

- Inclusion of the accelerometer data only slightly improves the accuracy of the supervised learning algorithms and significantly degrades the accuracy of the unsupervised algorithms.

The comparatively strong performance of unsupervised algorithms is very promising because they potentially enable MA detection on a large scale without significant human effort in labeling training data, which addresses a significant problem in analyzing EDA data collected using wearables. Some of the results from this thesis are presented in [10].

Outline

The rest of the thesis is organized as follows:

Chapter 2 provides the background information and details on electrodermal activity, as long as the importance of detecting the MAs in EDA data. It then gives a literature review about the MA suppression and MA detection on EDA data along with other psycho-physiological data. Chapter 3 describes the two data sets that we use for our research. Chapter 4 explains the methods we use for the experiment, from how to construct the features to what machine learning algorithms we use in our experiment. Chapter 5 gives the details on the experiment set-up, including the criterion the three experts follow while labeling the data, the way to generate the ground truth labels, and the cross validation method that is used in the in-sample and out-of-sample prediction. Chapter 6 and 7 analysis and discusses the experiment results, then summarizes the thesis and provides the future directions of the research.

Chapter 2

Background and Related Work

2.1 Electrodermal Activity

EDA refers to the electrical properties of the surface of the skin. When people are under stress or at a high level of emotion or activity, the sympathetic nervous system is highly aroused, then sweat gland activity increases, the skin gets sweaty which in turn changes the electrical potential of the skin, and this change is measured by the wearable devices and known as the skin conductance response (SCR).[11] [1] Based on the description in [1], a typical SCR starts with a steep onset, followed by an exponential decay and lasts between 1-5 seconds, the minimum amplitude of an SCR is .01 μs. An example of a typical SCR is shown in Figure 2-1.

2.2 Motion Artifacts in EDA and PPG data

While measuring the EDA data using wearable devices, such as the wristband or the smart watch, the EDA signal can be easily affected by the motion artifacts. The MAs usually caused by the change of the contact between the skin and the two recording electrodes, which is generated by pressure or excessive movements during the daily activities. Correctly analyzing the EDA signal with MA involved is quite challenging, since MA might be misidentified as the SCR.[12] For example, since the

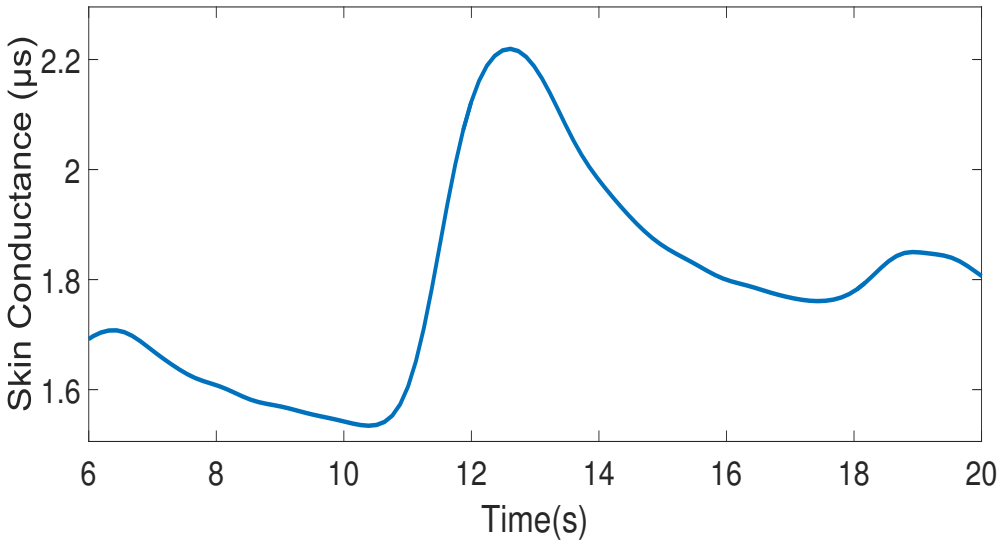

Figure 2-1: An example of a typical SCR.

SCR typically starts with a steep onset and has a decay, some researchers use a peak detection algorithm to identify the SCR,[13] and many MAs can also generate a peak in EDA, thus the MAs might be misclassified as SCRs and influence the analysis. Hence, the identification of the EDA portions that contain MA becomes extremely important in order to obtain the clear EDA data.

As we know that EDA is one of the psycho-physiological signals that can be collected by wearable devices, and many previous types of research have been done on detecting and correcting the motion artifacts in these different psycho-physiological signals. Photoplethysmography (PPG) signal, which is associated with the heart rate estimation, is one of the most important psycho-physiological signals other than EDA that attracts lots of attentions from the researchers. PPG is the change of blood's volume in the capillary vessel and it is often used for heart rate measurement.[14] The changes of blood flow in the micro-vascular vessel influence the light absorption rate and this change can be detected by photo-diode and shown as the PPG.[15] And similar to EDA, the waveform of PPG can also be easily distorted by MA, especially

when the subject is running.

2.2.1 Algorithms to deal with the Motion Artifacts in PPG data

Many signal processing and noise-reduction techniques have been proposed to identify and remove MA in the PPG. Han and Kim [16] developed a motion artifact reduction algorithm which is primarily based on the Normalized Least Mean Square (NLMS) adaptive filter to compensate the distorted signals using the three-axis accelerometer data. Fukushima et al.[17] developed a spectrum subtraction algorithm to remove the spectrum of acceleration data from that of a PPG signal. After the Fast Fourier Transform (FFT), the frequency of the PPG and accelerometer data are both shown as peaks in spectrums, thus the MA can be removed by simply subtracting the MA peaks from PPG in spectrums. Lin et al.[18] chose an adaptive filter with a synthetic reference to eliminate the noise outside the heart rate band and rebuild the MA free PPG signal with the use of the accelerometer data. Their heart rate detection algorithm is able to locate the heart rate peak in the spectrum that is mixed with MA. The TROIKA (signal decomposiTion, sparse signal RecOnstructIon, and spectral peaK trAcking) and JOSS (JOint Sparse Spectrum reconstruction) algorithms are similar and both developed by Zhang [19][9], the two algorithms are frequency domain based algorithms that identified and removed spectral peaks of MA in PPG spectrums with the help of accelerometer data, the two algorithms have high estimation accuracy and are robust to strong motion artifacts. Even though some of the above mentioned algorithms can successfully identify and remove the MAs in the PPG signal, they would not work well with the EDA data. As EDA is a non-periodic signal compared to the PPG, the EDA won't have significant and clear spectrum peaks after the FFT, thus these frequency domain algorithms would only contaminate the EDA

data and make the identification of the MAs more challenging. However, since the use of the accelerometer data played an important role in detecting and removing the MAs from the PPG data, there is a high chance that the accelerometer data would also be an important factor in dealing with the MAs in EDA data, and the evaluation of its usefulness in the EDA data is noteworthy.

2.2.2 Algorithms to deal with the Motion Artifacts in EDA data

MAs in EDA data are dealt with in two very similar ways: MA suppression and MA detection. However, unlike PPG and other psycho-physiological signals, not much work has been done on dealing with the MAs in EDA data.

On the MA suppression side, many researchers attempt to clean the portions of data with MAs by passing it through some type of smoothing filter [20, 3, 21, 22] or low-pass filter [23][24], which can only deal with small magnitude MA and unable to remove the obvious high-intensity artifacts. Other researchers used some heuristic methods to identify MAs by looking for abnormal signal variations. Storm et al.[25] set thresholds to the amplitude and width of the SCRs and only count the peaks that fulfill the criteria as SCRs; Kocielnik et al.[22] defined the maximal possible increase and decrease of the EDA data based on their experimental results, and eliminate the samples that do not meet those criteria. However, the thresholds and criteria they established only suited for particular experiments, and were only verified through visual inspection. Therefore, these heuristic techniques are not guaranteed to generalized to other researches or experiments. Chen et al.[20] proposed a method that can remove the large magnitude motion artifacts from EDA data by using a stationary wavelet transform (SWT), but for the artifacts that have a similar intensity with the SCR, it may not work well. In general, the main downsides to MA suppression is that it either

distorts the EDA signal that includes the informative SCRs, or it couldn't establish a universal criteria that can work in different settings.

The MA detection, on the other hand, aims to identify portions of the data with MAs so they can be removed from further analysis. Hedman[26] used two independent EDA sensors, and if there is a rapid increase or decrease in only one sensor, he treated that as an MA. However, this method is completely lab-based and not practical for general use at all. Taylor et al. [12] formulated MA detection as a supervised machine learning problem and demonstrated that supervised learning algorithms can automatically detect MAs on a small EDA data set collected in a lab environment. The downside to supervised algorithms is that they require lots of labeled data to train, which requires significant human effort. Also, the data used in these former researches are all collected in a limited lab environment, and they only use EDA data to create features that fed into the machine learning algorithms. We apply supervised as well as the unsupervised learning algorithms on a real life dataset using both EDA and acceleration features to evaluate their performance in a real life environment.

Chapter 3

Data Description

We use two publicly available data sets with EDA and 3-axis accelerometer data, both collected using an Affectiva Q sensor [7] worn on the wrist, totaling about 23 hours.

3.1 UT Dallas Stress (UTD) Data

This data set was collected at the University of Texas at Dallas [27]. A total of 20 college students (14 males and 6 females) were asked to perform a sequence of tasks subjecting them to three types of stress: physical stress (standing, walking, and jogging), cognitive stress (mental arithmetic and the Stroop test), and emotional stress (watching a horror movie clip). Each task was performed for 5 minutes, and tasks were separated by 5 minute relaxation periods. The EDA and 3-axis accelerometer data were collected using an Affectiva Q sensor worn on the wrist of the subjects during the experiment. Altogether, about 13 hours worth of data was collected. The detail information of the subjects is shown in Table 3.1.

Over all 20 subjects, 3.8% of the data was determined by three human experts to contain MAs (see Expert Labeling section for details). On the low end, three subject's data contained no MAs as determined by either expert, while on the high end, one subject's data contained 14% MAs.

Table 3.1: Information about the 20 subjects in UT Dallas Stress (UTD) Data

Subjects Information				
Subject ID	Age	Gender	Height [cm]	Weight [kg]
1	30	M	177	94
2	28	M	172	68
3	28	M	177	91
4	22	M	167	58
5	30	M	182	82
6	30	F	167	58
7	33	F	157	90
8	27	M	182	64
9	25	M	177	68
10	23	M	180	64
11	26	M	170	71
12	32	F	162	53
13	29	F	167	64
14	19	F	160	50
15	23	M	165	64
16	24	M	180	54
17	23	M	167	57
18	23	M	177	64
19	22	M	167	64
20	24	F	160	44

Table 3.2: Descriptions of 10 hours of labeled time segments in Alan Walks Wales data

Date	Time Segment	Segment Length	% Artifacts	Description
		Walking data: 5 hours total		
2013/4/24	4:47 AM–5:47 AM	1 hour	56%	Walking and chatting with friend
2013/5/06	11:38 AM–12:38 PM	1 hour	42%	Walking and hiking
2013/5/28	6:43 PM–7:43 PM	1 hour	6%	Walking around
2013/6/04	4:03 PM–5:03 PM	1 hour	19%	Walking around
2013/6/10	10:46 AM–11:46 AM	1 hour	42%	Walking around
		Resting data: 5 hours total		
2013/4/24	5:47 AM–6:47 AM	1 hour	43%	Drinking and chatting in a pub
2013/5/14	2:16 PM–2:56 PM	40 minutes	15%	Having lunch at a restaurant
2013/5/19	3:20 PM–4:00 PM	40 minutes	4%	Having lunch and drinking beer
2013/6/03	1:00 PM–2:00 PM	1 hour	5%	Having lunch and chatting with friend
2013/6/11	11:38 AM–12:18 PM	40 minutes	10%	Having lunch and reading newspaper
2013/7/10	6:08 PM–7:08 PM	1 hour	8%	Having dinner at a restaurant

3.2 Alan Walks Wales (AWW) Data

This data set was collected by Alan Dix while he walked around Wales from mid-April to July 2013 [28]. He collected 64 days of data and also wore a GPS and kept a diary of his activities. In order to evaluate the experiment in this daily life environment, We extracted segments of data over 10 different days resulting in 10 hours of data in total with a variety of daily activities. We split the segments into two categories of activities: walking and resting. The walking data contain 5 hours of data collected as Alan was walking or hiking, and the resting data contain 5 hours of data collected when he was resting, eating, reading, or interacting with others. The reason we divided the data in this way is that the walking data contain more physical movements, which in turn have more MAs, while the resting data contain less physical movements (and less MAs) but more cognitive and emotional activities. Details of the data that we extracted are shown in Table 3.2. We used both the EDA and the three-axis accelerometer data from the Alan Walks Wales dataset for our research.

For the walking data, 33% of the data was determined to contain MAs by all three human experts. On the low end, one segment's data contained 6% MAs, while on the

high end, one segment's data contained 56% of MAs. For the resting data, 15% of the data was determined to contain MAs. On the low end, one segment's data contained 3.7% MAs, and on the high end, one segment's data contained 43% of MAs.

Chapter 4

Methods

4.1 Feature Construction

Following the analysis in [12], we divide the data into 5-second time windows. We construct statistical features on both the EDA and simultaneously collected 3-axis accelerometer data. Overall, we construct 120 features including 24 EDA features and 96 acceleration features.

For the EDA data, we consider 6 different signals: the SC amplitude, its first and second derivatives, and the coefficients of a Discrete Wavelet Transform (DWT) with the Haar wavelet applied to the SC at 3 different time scales: 4 Hz, 2 Hz, and 1 Hz. Wavelet transforms are able to capture both frequency and time information, and the Haar wavelet is excellent at detecting sudden changes in signals, which frequently occur during MAs. The 6 signals we consider were found to be informative for MA detection in EDA by Taylor et al. [12]. For each of the 6 signals, we construct 4 statistical features: the mean, standard deviation, maximum, and minimum over the 5-second windows, resulting in 24 total EDA features.

To evaluate the value of the accelerometer data in detecting MAs, we construct the same set of features as for EDA on each of the 3 axes of accelerometer data, as well as on the acceleration magnitude (root-mean-square). The acceleration magnitude is

calculated as:

$$Mag = \sqrt{a_x^2 + a_y^2 + a_z^2} \tag{4.1}$$

where Mag is the magnitude, and a_x, a_y and a_z stand for the 3 different axis of the accelerometer data. This results in 24 features for each of the 3 axes and 24 features for the magnitude for a total of 96 accelerometer features. The detail description of the 120 constructed feature can be found in table A.1. We arrived at this set of features after examining a significant amount of prior work on classification using EDA [3, 21, 12, 5] and accelerometer [29, 30] data. The final selection of features is admittedly somewhat ad-hoc; however, we believe it is a fair representation of commonly used features in the literature.

4.2 Data Preprocessing

The preprocessing of the dataset is very important, the normalization and standardization of the data are commonly applied before fed it into any algorithms in machine learning field. Since the range of values of raw data might vary widely, the objective functions of many machine learning algorithms will not work properly without normalization. For example, in the k-nearest neighbor (kNN) classification algorithm, the distance between two points can be calculated by the Euclidean distance, and if one of the features has a wide range of values, this particular feature will have a huge impact on the distance. Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance.[31]

In our experiment, after the feature construction is done, we normalized the feature matrix by calculating the standard score of each feature and use the standard scores as the final feature matrix. Standard scores are also called z-values, z-scores,

normal scores, or standardized variables, it indicates how many standard deviations an element is from the mean.[32] The z-score can be calculated from the following formula:

$$z = (X - \mu)/\sigma \tag{4.2}$$

where z is the z-score, X is the value of the element (in this case the value of each feature), μ is the sample mean, and σ is the standard deviation.

4.3 Feature Selection

We perform the feature selection to the 120 features we constructed. Feature selection is also known as variable selection, attribute selection or variable subset selection in the statistics and machine learning fields, it is the process of selecting a subset of relevant features that have the most contributions to the prediction variable or output. Feature selection techniques are used for three reasons before modeling the data:

- Improve the performance: the modeling accuracy could be improved by removing less valuable and irrelevant features.
- Reduce overfitting: removing the redundant features can enhance the generalization, the noise components won't be able to affect the final output easily.
- Reduce the training time: the dimensionality of the feature space can be reduced after the feature selection and thus can speed up the training process.

If the data contains many features that are either redundant or irrelevant and if the removal of these features won't cause much information loss, which is exactly our situation, then the feature selection process becomes essential.[33, 34] We only have

24 EDA features, however, the number of acceleration features we constructed is 4 times of the EDA one. Besides, the acceleration features contain both acceleration magnitude features and the 3 individual axis's features, which are highly likely to contain redundant features. So the feature selection step is very necessary in our case.

We apply the univariate feature selection technique in our research to select the valuable features, the feature selection is based on the family-wise error rate that calculated by doing the analysis of variance (ANOVA) F-test between the features and corresponding labels. In general, univariate feature selection works by selecting the best features based on univariate statistical tests. The ANOVA F-test is known to be nearly optimal in the sense of minimizing false negative errors for a fixed rate of false positive errors, and the family-wise error rate (FWER) is the probability of making one or more false discoveries, when performing multiple hypotheses tests.[34]

4.4 Machine Learning Algorithms

We examine 5 widely used supervised algorithms: support vector machines (SVMs), k-nearest neighbor (kNN) classifiers, random forests, logistic regression, and multi-layer Perceptron (MLP). We also formulate the problem of predicting MAs as an unsupervised anomaly detection problem. Among the different unsupervised anomaly detection approaches, we examine unsupervised variants of the first 3 supervised algorithms: One-class Support Vector Machines (1-class SVMs), kNN distances, and isolation forests, respectively.

The supervised algorithms are used for binary classification, where the two classes are MA and clean. The unsupervised algorithms are used for anomaly detection, where it is assumed that the training data consists of mostly clean data. We interpret the time windows predicted by the unsupervised algorithms as anomalies to be

predicted MAs.

4.4.1 Support Vector Machines

Support Vector Machine (SVM) is a classifier that constructs a high-dimensional hyper-plane to perform the classification.[35] The SVM training algorithm builds a model, or a high-dimensional hyper-plane, base on the given set of training examples with each marked as belonging to one or the other of two groups. New examples are then mapped into that model and predicted to belong to a group based on which side of the hyper-plane they are located. SVM also has different types of kernel functions to transform testing examples into a higher dimensional feature space and to easier classify the data.

4.4.2 K-nearest Neighbor Classification

The k-nearest neighbors algorithm is a non-parametric method used for both classification and regression.[36] For the k-nearest neighbors classifier, the input has k nearest training examples in the feature space, and the output is a class that generated by a majority vote of the testing example's neighbors, the testing example is assigned to the class most common among its k nearest neighbors.

4.4.3 Random Forests

Random forests or random decision forests[37] is an ensemble learning method for classification or regression. When used for classification, it constructs a multitude of decision trees while training and outputting the class that is the mode of the classes of the individual trees. Random forests averages multiple deep decision trees, trained on different random parts of the same training dataset, and eventually reducing the variance.[38] Because of this randomness, the bias of the forest usually increases

slightly, however due to averaging, its variance decreases, hence it generally boosts the performance in the final model.[39]

4.4.4 Logistic Regression

Logistic regression was developed by statistician David Cox in 1958.[40] The binary logistic regression is used to predict the probability of a binary response based on one or more predictor variables or features. For example, by using logistic regression, people could find out that the probability of a given outcome could be increased by a specific percentage with the existence of a risk factor. In the machine learning field, despite its name, logistic regression is a method for classification, not regression.

4.4.5 Multi-layer Perceptron

A Multi-layer perceptron (MLP) is a class of feed-forward artificial neural network. A MLP consists of at least three layers of nodes, in other words, between the input and the output layer, one or more non-linear layers could exist, and these non-linear layers are called hidden layers. Each node in the MLP is a neuron that uses a non-linear activation function except for the input nodes. MLP applies a supervised learning method called back-propagation for training. MLP distinguishes itself from the linear perceptron by its non-linear activation function and multiple layers. It can learn non-linear models in order to classify the data that is not linearly separable.[41, 42, 43]

4.4.6 One-class Support Vector Machines

The one-class Support Vector Machines are usually used for anomaly detection. In one-class SVM, the support vector model is trained on data with only one class, the normal class. It deduces the properties of normal class to define a frontier, and then if the examples fall outside the frontier, it unlike belong to the normal class and

will be predicted as anomaly.[44] This is useful for anomaly detection because most of the times the network intrusion, fraud, or other anomalous behavior are very few in the everyday life.

4.4.7 K-nearest Neighbor Distances

The K-nearest Neighbor distance algorithm is different than the kNN classifier in that it uses the distance between a test sample and its k-th nearest training sample as its test statistic, whereas the kNN classifier performs a majority vote over the labels of the k nearest training samples.

4.4.8 Isolation Forests

The isolation forests first select a feature randomly, and then randomly select a split value between the maximum and minimum values of the selected feature to 'isolates' the observations. As the recursive partitioning can be represented by a tree structure, the number of splittings required to 'isolate' a sample and the path length from the root node to the terminating node are equivalent to each other. This path length is averaged over a forest of random trees to measure the normality. Random partitioning can create much shorter paths for anomalies. Hence, if a forest of random trees all create shorter path lengths for a particular sample, this sample is most likely to be anomaly.[45]

4.5 Parameter Tuning

We optimize the parameters of each algorithm by using a grid search with exponential grid and retain the parameters with the highest cross-validation accuracy (see Results section for details). The grid search is currently the most widely used method for parameter tuning, since it tries out every parameter in a given range and picks

the best one, Which is obviously guaranteed to get the global optimum. We use the Gaussian kernel for the SVM, which was found to be most accurate in [12], and ReLU activations for the MLPs with up to 2 hidden layers. For the unsupervised algorithms, we also use the grid search cross-validation approach to select parameters in order to provide a fair comparison to the supervised algorithms. Since this approach is likely not possible in practice with unlabeled data, we also experiment with different choices of parameters to determine the sensitivity of results to the parameter choices.

For SVMs, the kernel width and Penalty parameter C are optimized; similarly, the kernel width and the ν of the 1-class SVMs are optimized. For the kNN classifier and kNN distance, the number of neighbor k is tuned. The number of estimators in the random forests and isolation forests is optimized. The inverse of regularization strength, C, is tuned for the logistic regression. The L2 penalty (regularization term) parameter α and the hidden layer size of the multi-layer perceptron is also determined in the parameter optimization process by the gird search.

Chapter 5

Experiment Set-up

The experiments involve evaluating the predictions of the machine learning algorithms compared to hand-labeled MAs by two EDA experts. Code to reproduce the experiments is available at `https://github.com/IdeasLabUT/EDA-Artifact-Detection`.

5.1 Expert Labeling

We have three EDA experts hand label each 5-second time window as clean or containing a MA using the EDA Explorer software [46, 12]. The EDA Explorer software is an on-line tool for visualization and analysis of Electrodermal Activity data, which is designed and hosted by the Affective Computing Group from the Massachusetts Institute of Technology. The EDA Explorer can detect noise within the EDA signal, detect SCRs, visualize the results, compute features which users can download, and help researchers label their own signal data. We only use the software for the EDA data labeling in our research. While labeling, All experts used a common set of criteria to define an MA in the SC:

- A peak that does not have an exponential decay, except in the case where two peaks are very close to each other in a short time period so that the decay of the first peak is interrupted by the second peak;

- A sudden change in SC correlated with motion;
- A sudden drop of more than 0.1 μS in SC.

The first two criteria were used in [12]; we added the third criterion based on the physiology of EDA. SC can increase suddenly due to sweat glands releasing sweat, but there is no physiological mechanism for SC to decrease suddenly [1].

We combined the 3 sets of labels from the three experts into a single label set by the majority vote, which means if the majority of the three experts (which is at least 2 experts in our case) agree on one label, we will assign this exact label as the final label for that corresponding time window.

The labels from all three experts were in agreement for 95% and 87% of the time windows for the UTD and AWW data, respectively. The only two possibilities for disagreement are two experts labeling as MA and one as clean or vice-versa. When there was disagreement, 2 MA/1 clean occurred 38% and 44% of the time in the UTD and AWW data, respectively.

5.2 In-sample Prediction

For the UT Dallas dataset, we evaluate the in-sample prediction accuracy for each learning algorithm using a leave-one-group-out cross-validation (CV) approach, which was found to be preferable to k-fold CV for time series data due to the dependence of time windows [30]. Each training set thus consists of all the samples except the ones that are in a specific group, and after the training, the model will be tested on that left out group of data samples. Each subject in the UT Dallas dataset is considered to be one group, and thus there are 20 groups for the UT Dallas dataset. For the AWW data, we have only one single subject, but we have 10 hours of labeled time segments spaced out across 10 different days in the data trace with a total of 11 different time

segments, so we consider each of these time segments as one group and end up with 11 groups of data for the Alan Walks Wales dataset.

We also split the Alan Walks Wales data into two separate datasets containing only resting data and only walking data prior to performing the CV to evaluate the prediction accuracy for both categories of activities, resulting in Alan Walks Wales resting dataset with 6 groups and Alan Walks Wales walking dataset with 5 groups.

For both data sets, we use the Area Under the Curve (AUC) of the Receiver Operator Characteristic (ROC) as the accuracy metric. To test the value of the accelerometer data, we test the learning algorithms on three different feature sets: ACC only, containing only the 96 features constructed from the accelerometer; EDA only, containing the 24 features constructed from EDA; and all 120 features.

In the feature selection process, we select the features for the three datasets individually, and within each dataset, we select the features for EDA only feature set and acceleration feature set respectively, and then add up the selected features from these 2 feature sets to form the All feature set. For the Alan Walks Wales resting dataset, we end up with 61 acceleration features, 16 EDA features, and a total of 77 features in the All feature set; for the Alan Walks Wales walking dataset, there are 65 acceleration features, 16 EDA features, and add up to 81 features for the All feature set; and for the UT Dallas dataset, the number of the selected features are: 64 acceleration features, 21 EDA features, and 85 features in All feature set. We only use the selected features for the 5 supervised learning algorithms, and we use the original feature sets for the 3 unsupervised learning algorithms. We optimize the parameters of all the eight algorithms for the three different feature sets individually using the grid search method.

5.3 Out-of-sample Prediction

For out-of-sample prediction, we train each learning algorithm on all of the time windows in one of the two data sets, then test the algorithm's predictions on all of the time windows in the other data set. In this experiment, we treat the Alan Walks Wales data as a single data set by combining the resting and walking sets together so that the Alan Walks Wales data and the UT Dallas dataset are of roughly the same size (10 hours of data vs. 13 hours of data). This should be a tougher prediction task than in-sample prediction because the two data sets contain different activities and were collected in very different settings (lab-based setting vs. real-world setting) with different test subjects.

For the feature selection, same as the in-sample prediction task, we select the features for the two datasets individually, and only use the selected features for the 5 supervised learning algorithms. The selected features of the UTD data maintain the same, which is 64 acceleration features, 21 EDA features, and 85 features in total. Since the AWW walking and resting datasets are combined together, we performed the univariate feature selection on this "new" dataset and thus 15 features are selected as the EDA only features, 64 features are selected as the ACC only features, and by combining the EDA only features and ACC only features, we get a total of 79 features for the all features set. For all eight algorithms, we use the same parameter tuning method on the training data set as in the in-sample prediction task. We tested the same three feature sets as in the in-sample prediction task.

Chapter 6

Results

6.1 In-sample Prediction

The AUC results of the in-sample prediction task for Alan walks Wales resting, Alan walks Wales walking, and UT Dallas dataset are shown in Table A.2,A.3, and A.4, respectively. The receiver operating characteristic (ROC) curves of the in-sample prediction task using EDA features only for Alan walks Wales resting is shown in Figure 6.1, and the rest of the ROC curves, including the ROC curves for the acceleration only feature sets and all feature sets are shown in Appendix A.2: The ROC curves for the in-sample prediction task.

The first observation from all three data sets is that using only the accelerometer-derived features (ACC only) provides a significantly worse predictor than the other two feature sets. Besides, in all three data sets, we do not observe a significant benefit in using all of the features rather than just the EDA-derived features (EDA only). The inclusion of accelerometer-derived features appears to have minimal effect on the supervised learning algorithms, which only improve by 0.4% on average, and it reduces the AUC of the unsupervised algorithms by 4.3% on average. Additionally, when comparing the AUCs and the ROC curves of the supervised and unsupervised algorithms on EDA features only, we notice that the unsupervised algorithms (on

Table 6.1: Alan Walks Wales resting data's in-sample prediction AUC using leave-one-subject-out cross-validation. The top five algorithms are supervised, while the bottom three are unsupervised. Highest value for the data set is shown in bold.

Alan Walks Wales resting data			
Algorithm	**All features**	**ACC only**	**EDA only**
Logistic regression	0.843	0.665	0.715
Multi-layer Perceptron	0.681	0.414	0.694
SVM	0.685	0.426	0.684
kNN classification	0.670	0.595	0.701
Random forest	0.750	0.601	0.705
1-class SVM	0.830	0.780	0.767
kNN distance	0.803	0.719	**0.891**
Isolation forest	0.818	0.697	0.800

Table 6.2: Alan Walks Wales walking data's in-sample prediction AUC using leave-one-subject-out cross-validation. The top five algorithms are supervised, while the bottom three are unsupervised. Highest value for the data set is shown in bold.

Alan Walks Wales walking data			
Algorithm	**All features**	**ACC only**	**EDA only**
Logistic regression	0.812	0.652	0.798
Multi-layer Perceptron	0.791	0.655	0.778
SVM	0.801	0.641	0.782
kNN classification	0.770	0.650	0.766
Random forest	0.811	0.658	0.784
1-class SVM	0.579	0.586	0.759
kNN distance	0.776	0.691	**0.847**
Isolation forest	0.673	0.564	0.781

Table 6.3: UT Dallas data's in-sample prediction AUC using leave-one-subject-out cross-validation. The top five algorithms are supervised, while the bottom three are unsupervised. Highest value for the data set is shown in bold..

	UT Dallas data		
Algorithm	**All features**	**ACC only**	**EDA only**
Logistic regression	**0.942**	0.851	0.935
Multi-layer Perceptron	0.927	0.835	0.926
SVM	0.917	0.843	0.864
kNN classification	0.922	0.868	0.926
Random forest	0.936	0.880	0.926
1-class SVM	0.880	0.882	0.908
kNN distance	0.923	0.885	0.935
Isolation forest	0.910	0.895	0.900

EDA features only) perform very competitively with and sometimes even better than the supervised algorithms. From the ROC curve of the Alan walks Wales resting with EDA features only dataset, which is shown in Figure 6.1, we can clearly see that the three unsupervised learning algorithms outperform all of the supervised learning algorithms greatly, with the highest AUC which is 0.89 for kNN distance, second highest AUC 0.80 for isolation forests, and third highest AUC 0.77 for 1-class SVMs. We expand on these points in the Conclusion and Future Works section.

Also, by applying the feature selection for the supervised learning algorithms, the AUC is only slightly improved for most of the cases, the original results shown in the ISWC paper that without the feature selection are included in the Appendix A.4: The AUC tables shown in ISWC paper without feature selection.

6.2 Out-of-sample Prediction

The results of the out-of-sample prediction task are shown in Table A.5 and Table 6.5 for the AUC, and one of the receiver operating characteristic (ROC) curves for the out-of-sample prediction task, which is training on Alan Walks Wales data and

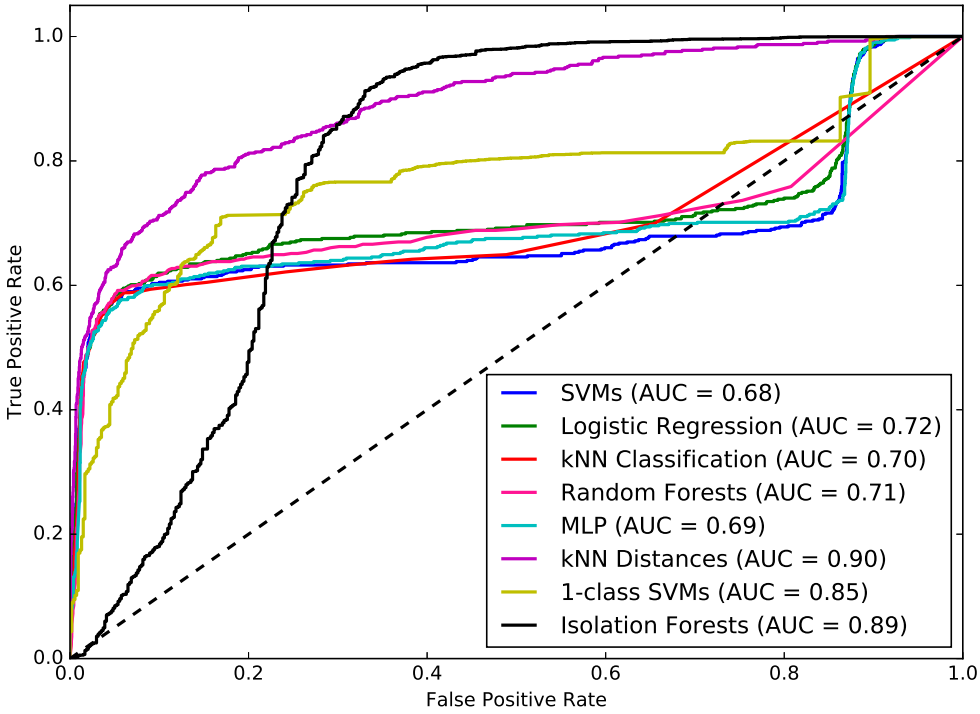

Figure 6-1: The ROC curves for Alan walks Wales resting data using only EDA features.

testing on UT Dallas data using ACC features only, is shown in Figure 6-2. The rest of the ROC curves for the out-of-sample prediction task are shown in Appendix A.0.3: The ROC curves for the out-of-sample prediction task.

In this task, from Table A.5 and Table 6.5, we observe the very similar results as in the in-sample prediction task, that using the EDA only features resulted in better performance than using all features or ACC only features for the majority of the algorithms in both data sets, especially for the three unsupervised learning algorithms. Besides this, notice that the results are much worse for all the eight algorithms when training on the lab-based UTD data and testing on the real-world AWW data, as one might expect when attempting to generalize from data collected in a lab setting.

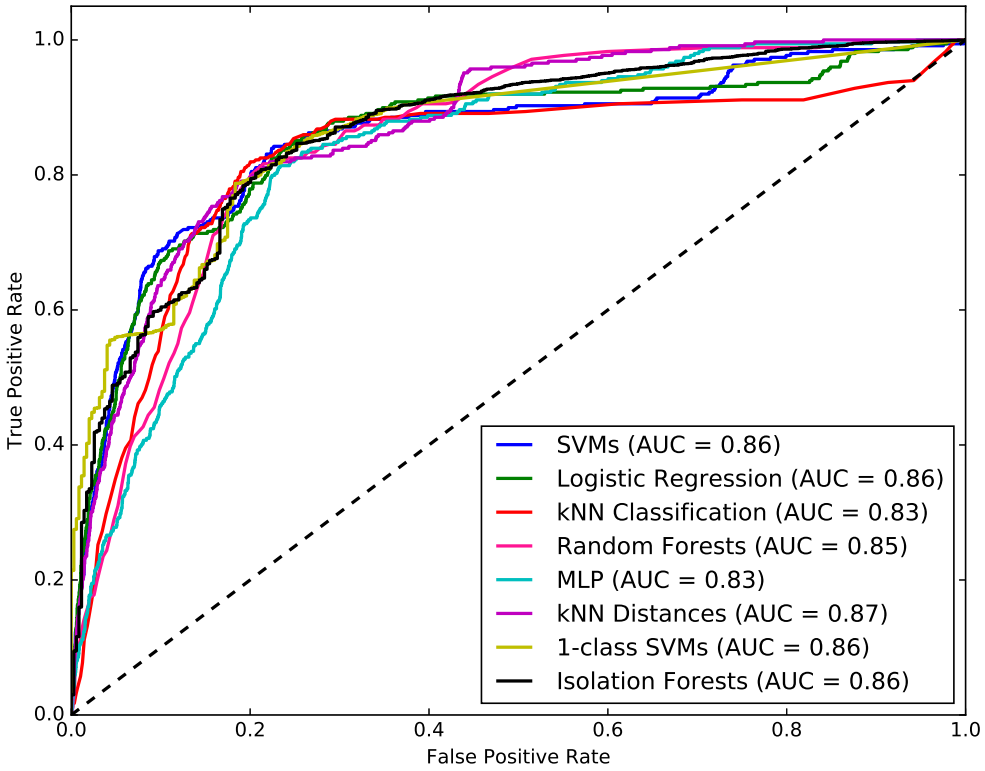

Figure 6-2: The ROC curves for training on Alan Walks Wales data and testing on UT Dallas data using only ACC features.

Notice also that the unsupervised algorithms are again highly competitive with the supervised ones. In the case that training on the AWW data and testing on the UTD data with ACC features only, as we can see from both Table A.5 and Figure 6-2, the three unsupervised algorithms achieve the top 3 highest AUC, and the kNN distances algorithm has the highest, which is 0.870.

In practice, choosing parameters for the unsupervised algorithms is very difficult without labeled data, in which case CV is not possible. We do find that the kNN distance algorithm does not appear to be very sensitive to the choice of the number of neighbors. By sweeping the number of neighbors from 1 to 30, the AUC remains

Table 6.4: Out-of-sample prediction AUC while train on Alan Walks Wales dataset and test on UT Dallas dataset. Highest value for the data set is shown in bold.

Train on AWW, test on UTD			
Algorithm	**All features**	**ACC features**	**EDA features**
Logistic regression	0.942	0.857	0.940
Multi-layer Perceptron	0.943	0.829	0.934
SVM	0.942	0.850	0.940
kNN classification	0.918	0.844	**0.946**
Random forest	0.922	0.860	0.937
1-class SVM	0.879	0.862	0.894
kNN distance	0.906	0.870	0.905
Isolation forest	0.902	0.864	0.905

Table 6.5: Out-of-sample prediction AUC while train on UT Dallas dataset and test on Alan Walks Wales dataset. Highest value for the data set is shown in bold.

Train on UTD, test on AWW			
Algorithm	**All features**	**ACC features**	**EDA features**
Logistic regression	0.828	0.695	0.846
Multi-layer Perceptron	0.836	0.691	0.849
SVM	0.844	0.690	0.822
kNN classification	0.763	0.670	0.827
Random forest	0.749	0.391	**0.850**
1-class SVM	0.731	0.658	0.814
kNN distance	0.727	0.647	0.835
Isolation forest	0.741	0.670	0.759

between 0.936 and 0.938 when training the kNN distance algorithm on the AWW data and testing on the UTD data. However, isolation forests are slightly more sensitive to the parameters, the AUC ranging from 0.86 to 0.91 for the number of base estimators changing between 1 to 40. And 1 class SVMs are the most sensitive ones, the AUC can be varied up to 0.2 difference by choosing different kernel width γ and ν value.

Chapter 7

Conclusion and Future Works

In all of our experiments, we found, somewhat surprisingly, that the accelerometer-derived features added little value to supervised learning algorithms (0.4% improvement on average). This does not necessary imply that the accelerometer data itself has little value—it could be that the features we adopted, which are commonly used for activity recognition, are not ideal for MA detection. However, the experts noticed that, on some occasions, even though the acceleration changes, the EDA doesn't get affected at all. This phenomenon has been noticed quite frequently by all three experts especially when they are labeling the Alan Walks Wales data. This is somehow reasonable, since the Alan Walks Wales data contains much more physical movements than the UT Dallas data, it contains 5 hours of walking and hiking data in the walking subset, which results in tons of acceleration changes, however, such movements may not cause any change in the contact between the EDA electrodes and the skin and thus don't affect the EDA signal. An example of such a time window is shown in the Figure 7-1. From this figure, we can clearly see that one of the axis of the acceleration, which is the red lines in the figure, has a significant change, however, the EDA data hasn't been affected and doesn't change at all. Conversely, the Figure 7-2 shows an example of a time window where the accelerometer data is helpful. In the Figure 7-2, if we only concentrate on the EDA data, we might think it is the decay

of an SCR, which has no MAs at all, but after we examine the acceleration data, we can clearly see the correlation between the 2 data, and find out that the decay is actually associate with the acceleration change, thus it is actually a motion artifact in the EDA data. Since the accelerometer data only slightly improve the supervised algorithms, it is reasonable to expect that they would degrade the unsupervised algorithms, which cannot distinguish between important and irrelevant features without labeled data. This aligns with our experimental findings.

We also observed that the overall performance of the unsupervised learning algorithms can be very competitive with the supervised ones, especially for the real-world data that are collected in the everyday life instead of from a lab, which contains more MAs and with a much more complex scenario. We believe this finding has profound consequences, enabling automatic MA detection on a large scale without the need for significant human effort in labeling data!

Some future works could be focused on the feature construction and feature selection. The features we construct are the most commonly used ones, but they only consist and based on 4 main statistic values, a wider variety of the features, especially for the acceleration data, should be collected in order to have a more comprehensive and convincing result. The feature selection process should also be modified with some other widely used and reliable feature selection techniques, such as wrapper feature selection, recursive feature elimination or neighborhood components analysis, in order to have a more valuable subset of features that are closely related to each algorithm. The evaluation of algorithms for other more complex machine learning settings that lie in between supervised and unsupervised settings, especially semi-supervised learning and transfer learning, would also be of tremendous value for EDA motion artifact detection, as would evaluation of algorithms such as deep neural networks capable of automatically learning features directly from the raw data.

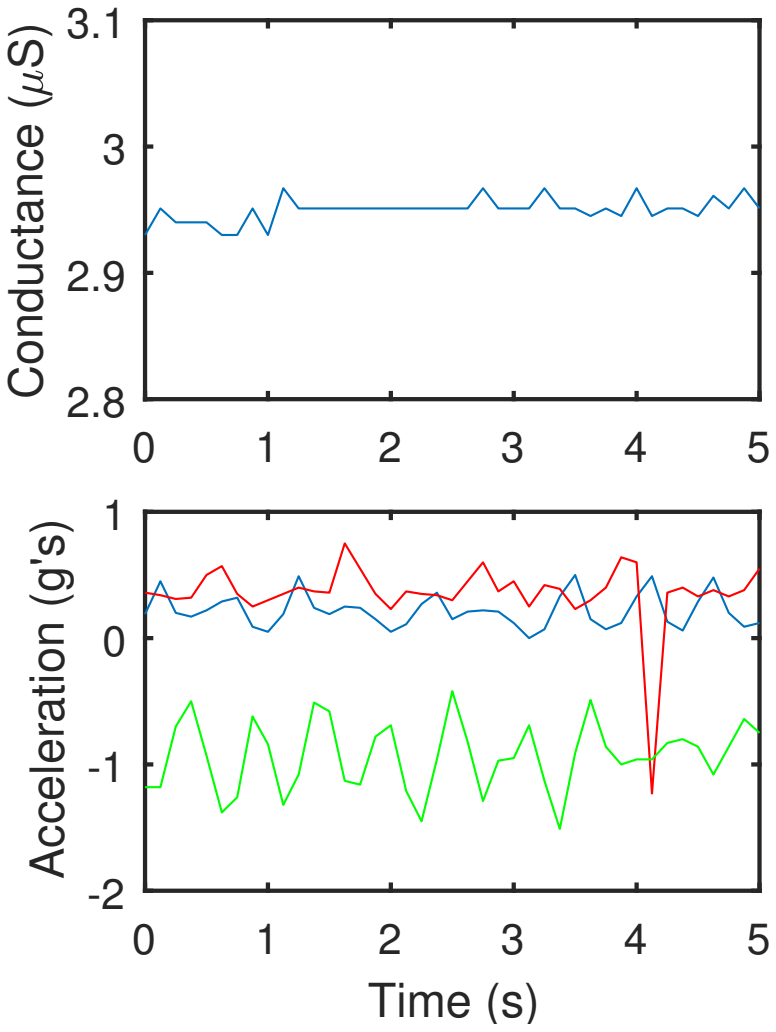

Figure 7-1: Examples of time windows where kNN classifier using all features fails but using EDA only succeeds.

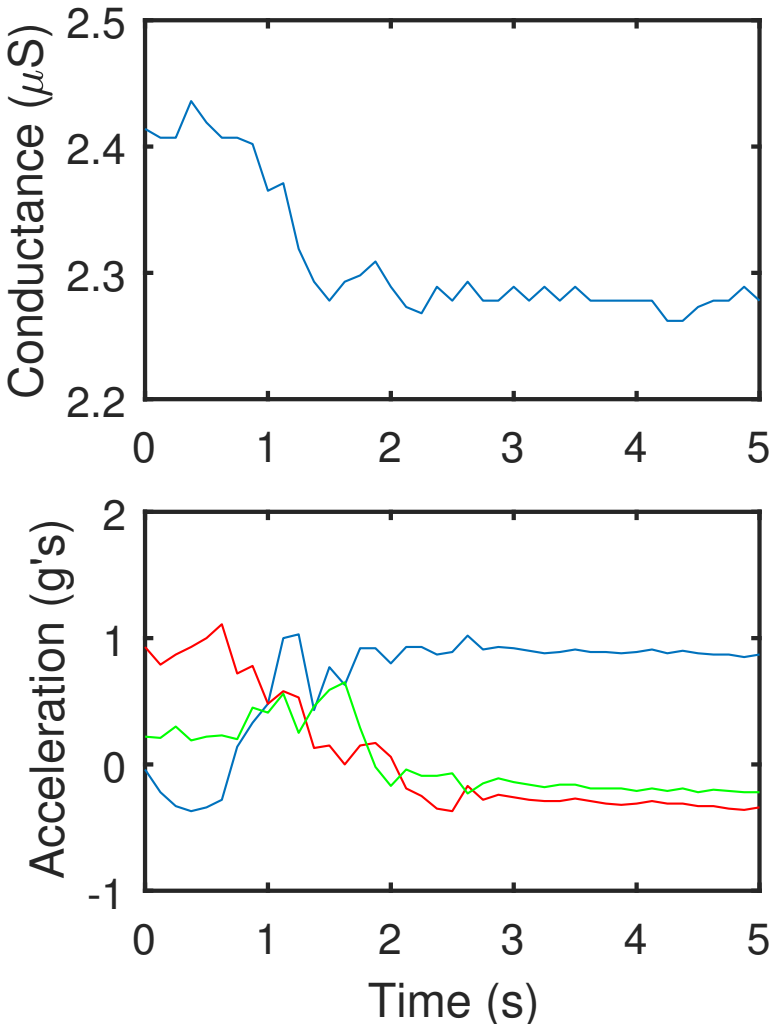

Figure 7-2: Examples of time windows where kNN classifier using all features succeeds but using EDA only fails.

References

[1] W. Boucsein. *Electrodermal Activity*. The Springer Series in Behavioral Psychophysiology and Medicine. Springer US, 2013.

[2] Fernando Silveira, Brian Eriksson, Anmol Sheth, and Adam Sheppard. Predicting audience responses to movie content from electro-dermal activity signals. In *Proc. ACM Int. Jt. Conf. Pervasive Ubiquitous Comput.*, pages 707–716, 2013.

[3] Javier Hernandez, Rob Morris, and Rosalind Picard. Call center stress recognition with person-specific models. *Proc. Int. Conf. Affect. Comput. Intell. Interact.*, pages 125–134, 2011.

[4] Hong Lu, Denise Frauendorfer, Mashfiqui Rabbi, Marianne Schmid Mast, Gokul T. Chittaranjan, Andrew T Campbell, Daniel Gatica-Perez, and Tanzeem Choudhury. StressSense: detecting stress in unconstrained acoustic environments using smartphones. In *Proc. ACM Conf. Ubiquitous Comput.*, pages 351–360, 2012.

[5] Annamalai Natarajan, Kevin S. Xu, and Brian Eriksson. Detecting divisions of the autonomic nervous system using wearables. In *Proc. 38th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, pages 5761–5764, 2016.

[6] S Koelstra, C Muhl, M Soleymani, Jong-Seok Lee, A Yazdani, T Ebrahimi, T Pun, A Nijholt, and I Patras. DEAP: A Database for Emotion Analysis Using Physiological Signals. *IEEE Transactions on Affective Computing*, 3(1):18–31, 2012.

[7] Liberate yourself from the lab: Q Sensor measures EDA in the wild. White paper, Affectiva Inc., 2012.

[8] Affectiva inc. Affectiva Q sensor, 2017. `http://qsensor-support.affectiva.com/`.

[9] Zhilin Zhang. Photoplethysmography-based heart rate monitoring in physical activities via joint sparse spectrum reconstruction. *IEEE Trans. Biomed. Eng.*, 62(8):1902–1910, 2015.

[10] Yuning Zhang, Maysam Haghdan, and Kevin Xu. Unsupervised motion artifact detection in wrist-measure electrodermal activity data. In *International Symposium on Wearable Computers*, 2017.

[11] Neil Carlson. *Physiology of behavior*. Pearson, Boston, 2013.

[12] Sara Taylor, Natasha Jaques, Weixuan Chen, Szymon Fedor, Akane Sano, and Rosalind Picard. Automatic identification of artifacts in electrodermal activity data. In *Proc. 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, pages 1934–1937, 2015.

[13] Sara Ann Taylor. *Characterizing electrodermal responses during sleep in a 30-day ambulatory study*. PhD thesis, Massachusetts Institute of Technology, 2016.

[14] Rasoul Yousefi, Mehrdad Nourani, Sarah Ostadabbas, and Issa Panahi. A motion-tolerant adaptive algorithm for wearable photoplethysmographic biosensors. *IEEE journal of biomedical and health informatics*, 18(2):670–681, 2014.

[15] Andrew Reisner, Phillip A Shaltis, Devin McCombie, and H Harry Asada. Utility of the photoplethysmogram in circulatory monitoring. *The Journal of the American Society of Anesthesiologists*, 108(5):950–958, 2008.

[16] Hyonyoung Han, Min-Joon Kim, and Jung Kim. Development of real-time motion artifact reduction algorithm for a wearable photoplethysmography. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 1538–1541. IEEE, 2007.

[17] Hayato Fukushima, Haruki Kawanaka, Md Shoaib Bhuiyan, and Koji Oguri. Estimating heart rate using wrist-type photoplethysmography and acceleration sensor while running. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pages 2901–2904. IEEE, 2012.

[18] Zhe Lin, Jin Zhang, Yanjiao Chen, and Qian Zhang. Heart rate estimation using wrist-acquired photoplethysmography under different types of daily life motion artifact. In *Communications (ICC), 2015 IEEE International Conference on*, pages 489–494. IEEE, 2015.

[19] Zhilin Zhang, Zhouyue Pi, and Benyuan Liu. Troika: A general framework for heart rate monitoring using wrist-type photoplethysmographic signals during intensive physical exercise. *IEEE Transactions on Biomedical Engineering*, 62(2):522–531, 2015.

[20] Weixuan Chen, Natasha Jaques, Sara Taylor, Akane Sano, Szymon Fedor, and Rosalind W Picard. Wavelet-based motion artifact removal for electrodermal activity. In *Proc. 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, pages 6223–6226, 2015.

[21] Javier Hernandez, Ivan Riobo, Agata Rozga, Gregory D. Abowd, and Rosalind W. Picard. Using electrodermal activity to recognize ease of engagement in children during social interactions. In *Proc. ACM Int. Jt. Conf. Pervasive Ubiquitous Comput.*, pages 307–317, 2014.

[22] Rafal Kocielnik, Natalia Sidorova, Fabrizio Maria Maggi, Martin Ouwerkerk, and Joyce HDM Westerink. Smart technologies for long-term stress monitoring at work. In *Computer-Based Medical Systems (CBMS), 2013 IEEE 26th International Symposium on*, pages 53–58. IEEE, 2013.

[23] Ming-Zher Poh, Tobias Loddenkemper, Nicholas C Swenson, Shubhi Goyal, Joseph R Madsen, and Rosalind W Picard. Continuous monitoring of electrodermal activity during epileptic seizures using a wearable sensor. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 4415–4418. IEEE, 2010.

[24] Akane Sano and Rosalind W Picard. Stress recognition using wearable sensors and mobile phones. In *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, pages 671–676. IEEE, 2013.

[25] H Storm, A Fremming, S Oedegaard, Ø G Martinsen, and L Moerkrid. The development of a software program for analyzing spontaneous and externally elicited skin conductance changes in infants and adults. *Clinical Neurophysiology*, 111(10):1889–1898, 2000.

[26] Elliott B Hedman. *In-situ measurement of electrodermal activity during occupational therapy*. PhD thesis, Massachusetts Institute of Technology, 2010.

[27] Javad Birjandtalab, Diana Cogan, Maziyar Baran Pouyan, and Mehrdad Nourani. A non-EEG biosignals dataset for assessment and visualization of neurological status. In *Proc. IEEE Int. Workshop Signal Process. Sys.*, pages 110–114, 2016.

[28] Alan Dix. Alan Walks Wales data, 2017. `http://alanwalks.wales/data/`.

[29] Óscar D. Lara and Miguel A. Labrador. A survey on human activity recognition

using wearable sensors. *IEEE Communications Surveys & Tutorials*, 15(3):1192–1209, 2013.

[30] Nils Y Hammerla and Thomas Plötz. Let's (not) stick together: Pairwise similarity biases cross-validation in activity recognition. In *Proc. ACM Int. Jt. Conf. Pervasive Ubiquitous Comput.*, pages 1041–1051, 2015.

[31] Selim Aksoy and Robert M Haralick. Feature normalization and likelihood-based similarity measures for image retrieval. *Pattern recognition letters*, 22(5):563–582, 2001.

[32] Wilfrid Joseph Dixon, Frank Jones Massey, et al. *Introduction to statistical analysis*, volume 344. McGraw-Hill New York, 1969.

[33] Mairead L Bermingham, Ricardo Pong-Wong, Athina Spiliopoulou, Caroline Hayward, Igor Rudan, Harry Campbell, Alan F Wright, James F Wilson, Felix Agakov, Pau Navarro, et al. Application of high-dimensional feature selection: evaluation for genomic prediction in man. *Scientific reports*, 5, 2015.

[34] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[35] Christopher JC Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[36] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[37] Tin Kam Ho. Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE, 1995.

[38] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[39] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[40] Strother H Walker and David B Duncan. Estimation of the probability of an event as a function of several independent variables. *Biometrika*, 54(1-2):167–179, 1967.

[41] Frank Rosenblatt. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, CORNELL AERONAUTICAL LAB INC BUFFALO NY, 1961.

[42] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[43] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314, 1989.

[44] Bernhard Schölkopf, Robert C Williamson, Alex J Smola, John Shawe-Taylor, and John C Platt. Support vector method for novelty detection. In *Advances in neural information processing systems*, pages 582–588, 2000.

[45] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 413–422. IEEE, 2008.

[46] Sara Taylor and Natasha Jaques. EDA Explorer. `http://eda-explorer.media.mit.edu/`.

Appendix A

Detailed feature description and ROC curve plots

A.1 The detailed description of all the features

Table A.1: Description of the constructed features

Feature Index	Statistical Features	Data/signal
1	mean	Raw EDA
2	standard deviation	
3	maximum	
4	minimum	
5	mean	First derivative of the raw EDA data
6	standard deviation	
7	maximum	
8	minimum	
9	mean	Second derivative of the raw EDA data
10	standard deviation	
11	maximum	
12	minimum	

Continued on next page

Table A.1 – continued from previous page

Feature Index	Statistical Features	Data/signal
13	mean	1Hz wavelet coefficients of the raw EDA
14	standard deviation	
15	maximum	
16	minimum	
17	mean	2Hz wavelet coefficients of the raw EDA
18	standard deviation	
19	maximum	
20	minimum	
21	mean	4Hz wavelet coefficients of the raw EDA
22	standard deviation	
23	maximum	
24	minimum	
25	mean	3-axis acceleration magnitude
26	standard deviation	
27	maximum	
28	minimum	
29	mean	First derivative of the 3-axis acceleration magnitude
30	standard deviation	
31	maximum	
32	minimum	
33	mean	Second derivative of the 3-axis acceleration magnitude
34	standard deviation	
35	maximum	
36	minimum	
37	mean	x axis acceleration
38	standard deviation	
39	maximum	
40	minimum	

Continued on next page

Table A.1 – continued from previous page

Feature Index	Statistical Features	Data/signal
41	mean	First derivative of the x axis acceleration
42	standard deviation	
43	maximum	
44	minimum	
45	mean	Second derivative of the x axis acceleration
46	standard deviation	
47	maximum	
48	minimum	
49	mean	y axis acceleration
50	standard deviation	
51	maximum	
52	minimum	
53	mean	First derivative of the y axis acceleration
54	standard deviation	
55	maximum	
56	minimum	
57	mean	Second derivative of the y axis acceleration
58	standard deviation	
59	maximum	
60	minimum	
61	mean	z axis acceleration
62	standard deviation	
63	maximum	
64	minimum	
65	mean	First derivative of the z axis acceleration
66	standard deviation	
67	maximum	
68	minimum	

Continued on next page

Table A.1 – continued from previous page

Feature Index	Statistical Features	Data/signal
69	mean	Second derivative of the z axis acceleration
70	standard deviation	
71	maximum	
72	minimum	
73	mean	1Hz wavelet coefficients of the 3-axis acceleration magnitude
74	standard deviation	
75	maximum	
76	minimum	
77	mean	2Hz wavelet coefficients of the 3-axis acceleration magnitude
78	standard deviation	
79	maximum	
80	minimum	
81	mean	4Hz wavelet coefficients of the 3-axis acceleration magnitude
82	standard deviation	
83	maximum	
84	minimum	
85	mean	1Hz wavelet coefficients of the x axis acceleration
86	standard deviation	
87	maximum	
88	minimum	
89	mean	2Hz wavelet coefficients of the x axis acceleration
90	standard deviation	
91	maximum	
92	minimum	
93	mean	4Hz wavelet coefficients of the x axis acceleration
94	standard deviation	
95	maximum	
96	minimum	

Continued on next page

Table A.1 – continued from previous page

Feature Index	Statistical Features	Data/signal
97	mean	1Hz wavelet coefficients of the y axis acceleration
98	standard deviation	
99	maximum	
100	minimum	
101	mean	2Hz wavelet coefficients of the y axis acceleration
102	standard deviation	
103	maximum	
104	minimum	
105	mean	4Hz wavelet coefficients of the y axis acceleration
106	standard deviation	
107	maximum	
108	minimum	
109	mean	1Hz wavelet coefficients of the z axis acceleration
110	standard deviation	
111	maximum	
112	minimum	
113	mean	2Hz wavelet coefficients of the z axis acceleration
114	standard deviation	
115	maximum	
116	minimum	
117	mean	4Hz wavelet coefficients of the z axis acceleration
118	standard deviation	
119	maximum	
120	minimum	

A.2 The ROC curves for the in-sample prediction task

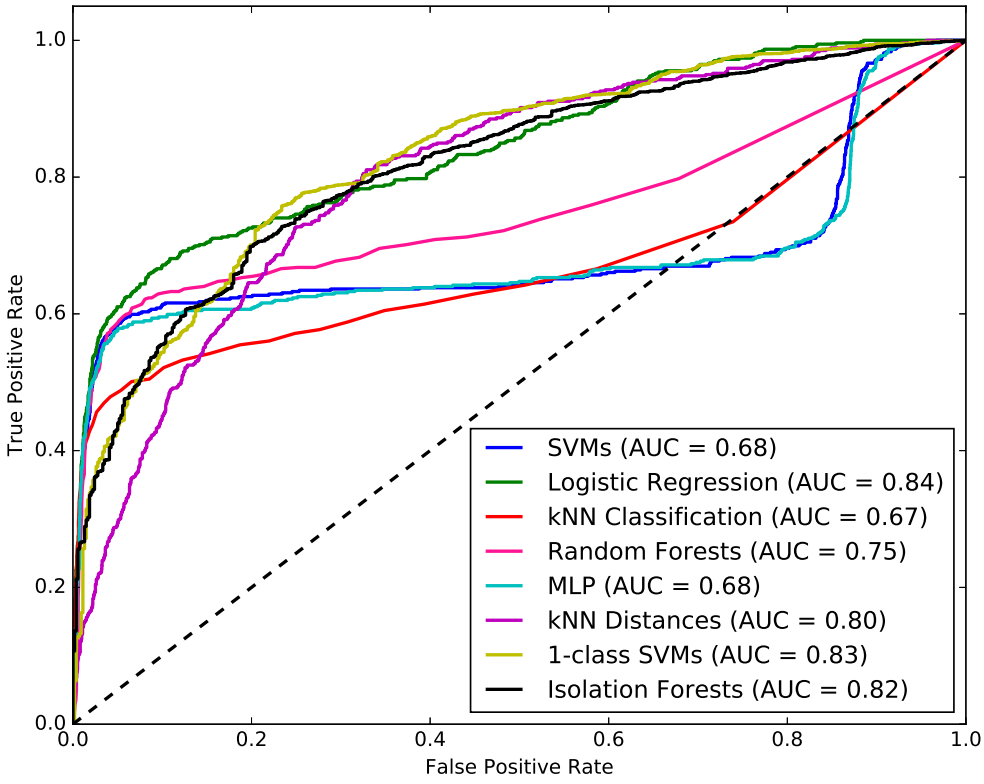

Figure A-1: The ROC curves for Alan walks Wales resting data using all features.

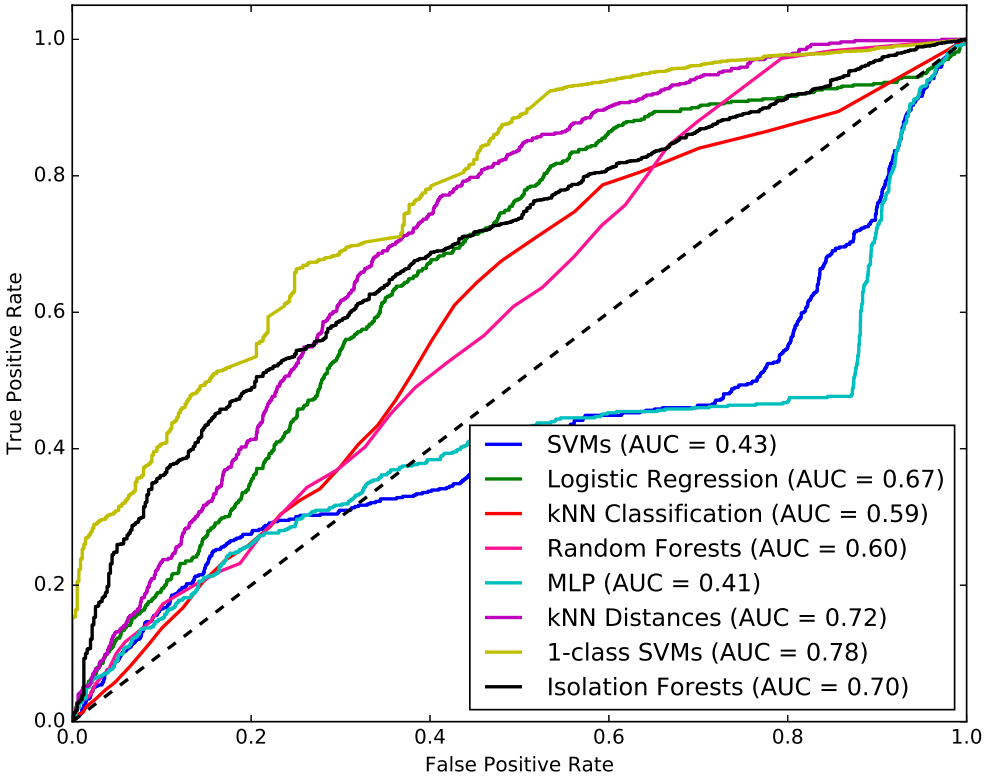

Figure A-2: The ROC curves for Alan walks Wales resting data using only acceleration features.

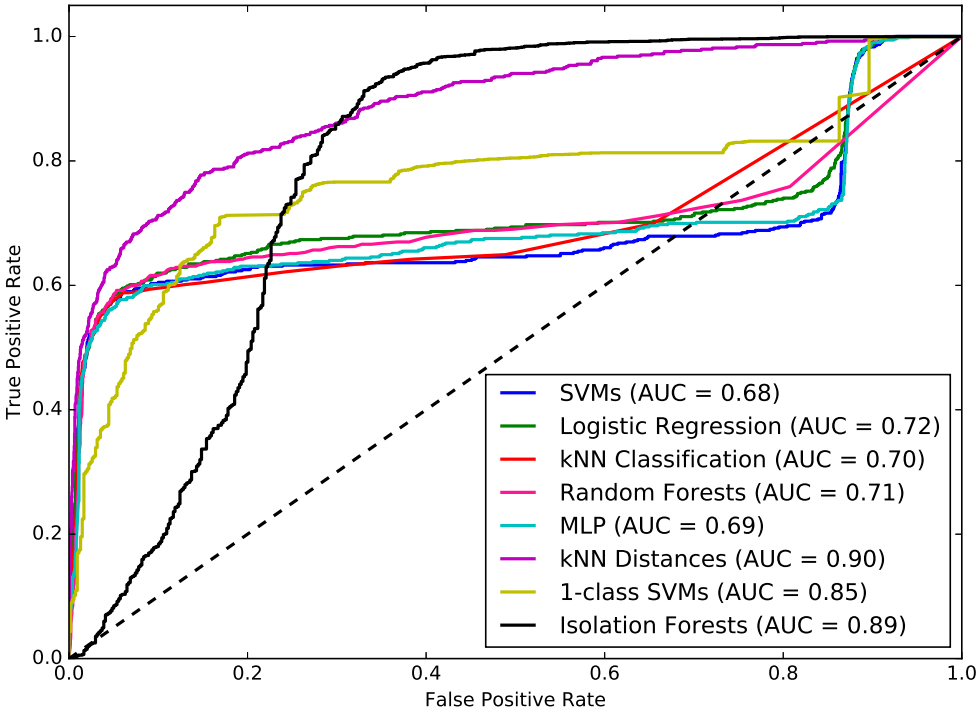

Figure A-3: The ROC curves for Alan walks Wales resting data using only EDA features.

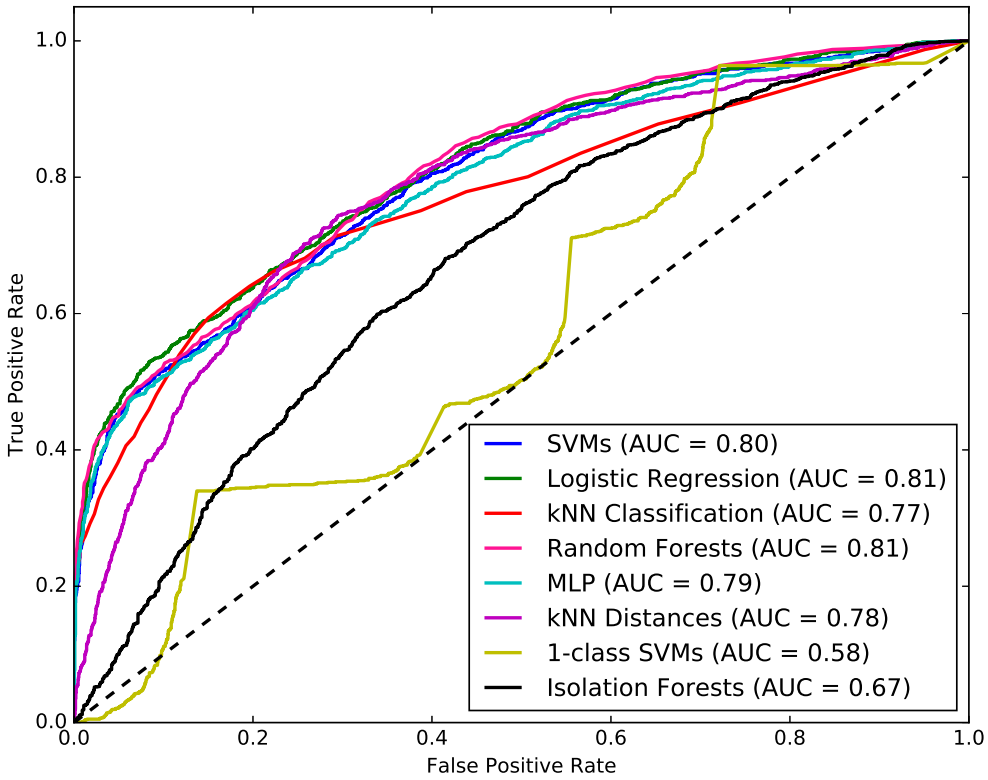

Figure A-4: The ROC curves for Alan walks Wales walking data using all features.

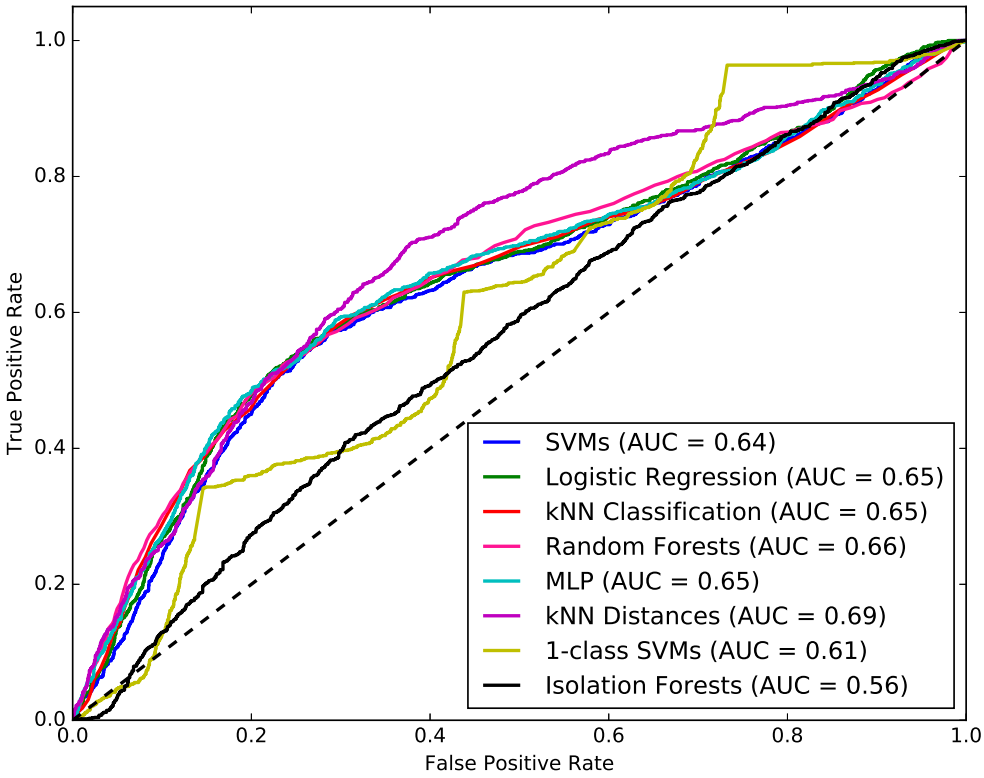

Figure A-5: The ROC curves for Alan walks Wales walking data using only acceleration features.

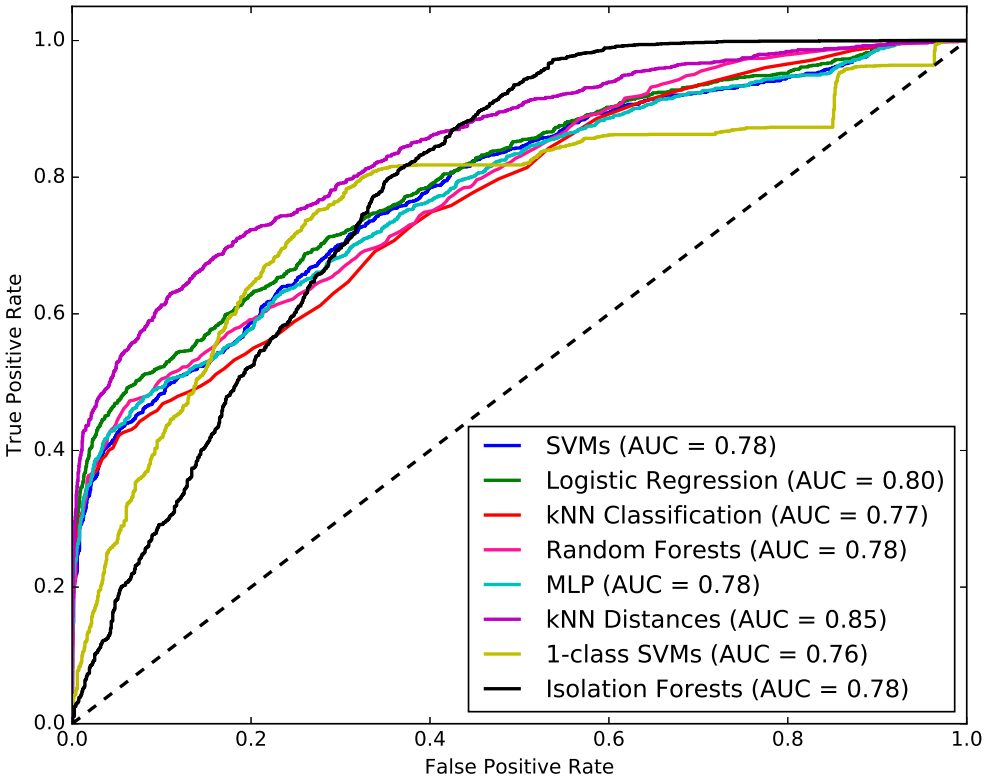

Figure A-6: The ROC curves for Alan walks Wales walking data using only EDA features.

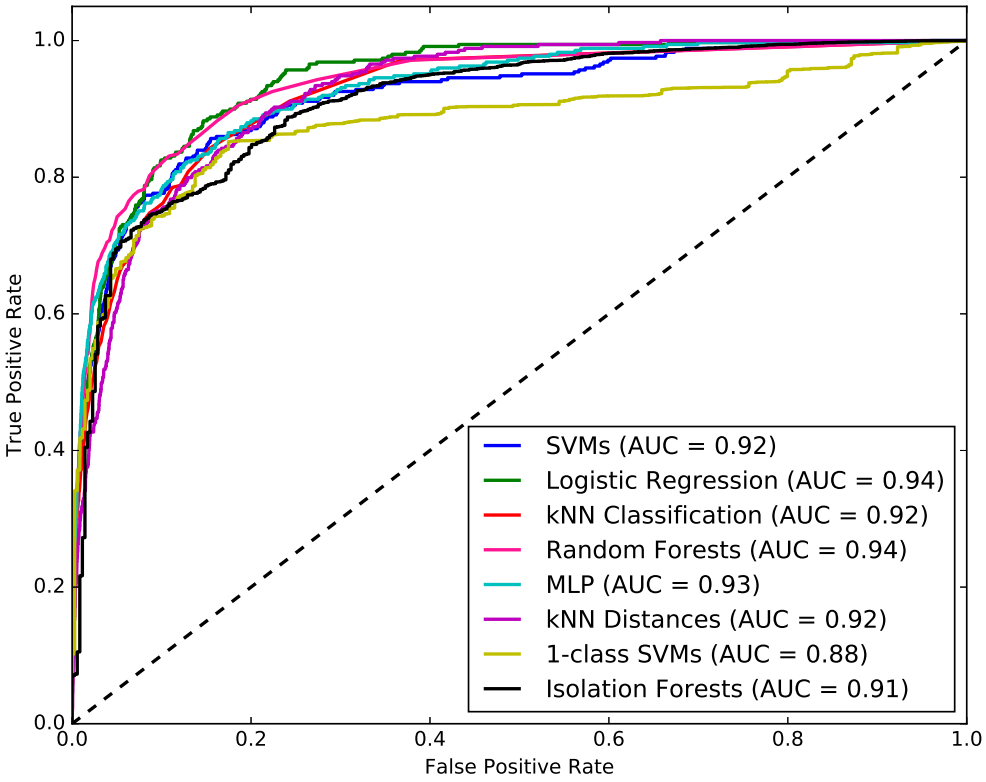

Figure A-7: The ROC curves for UT Dallas data using all features.

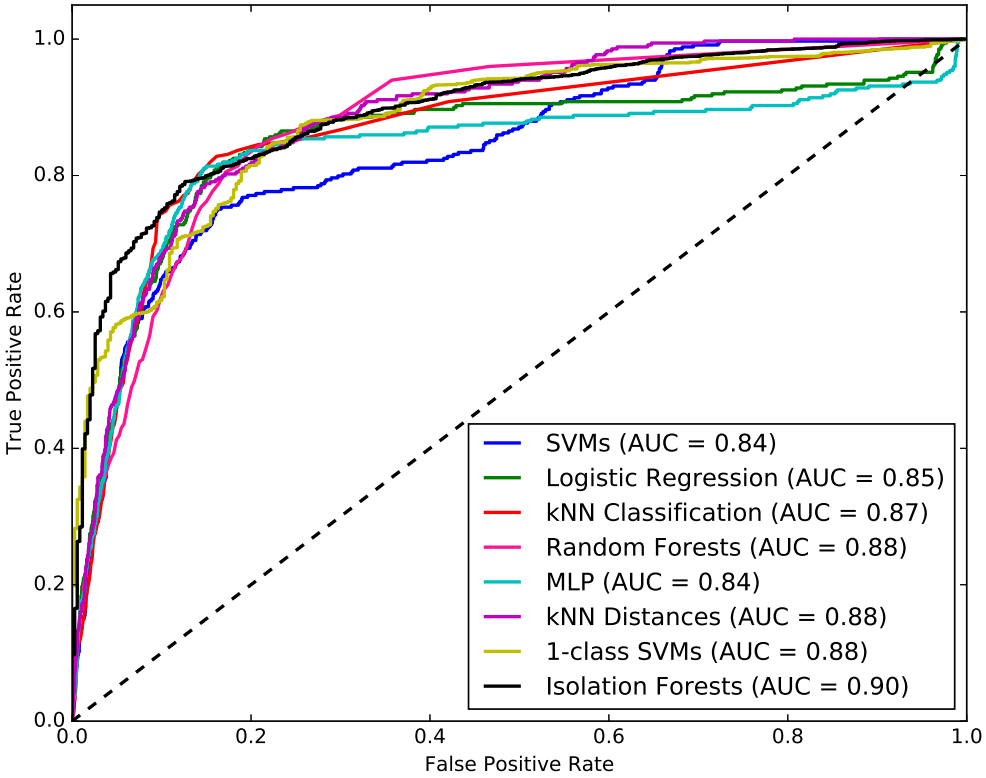

Figure A-8: The ROC curves for UT Dallas data using only acceleration features.

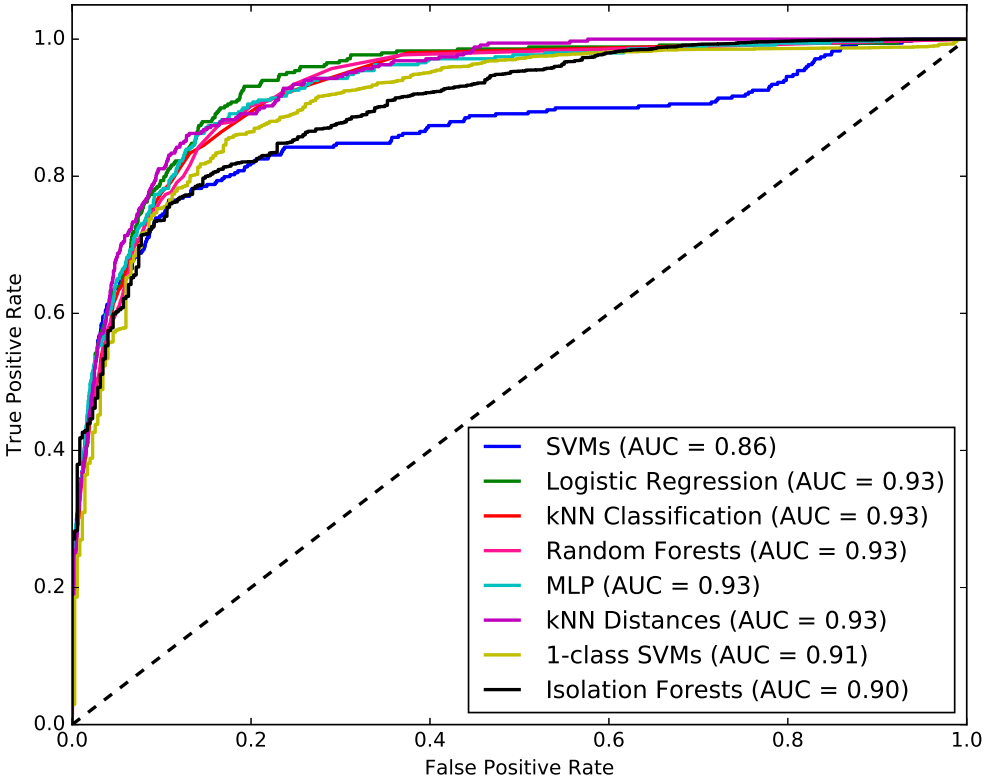

Figure A-9: The ROC curves for UT Dallas data using only EDA features.

A.3 The ROC curves for the out-of-sample prediction task

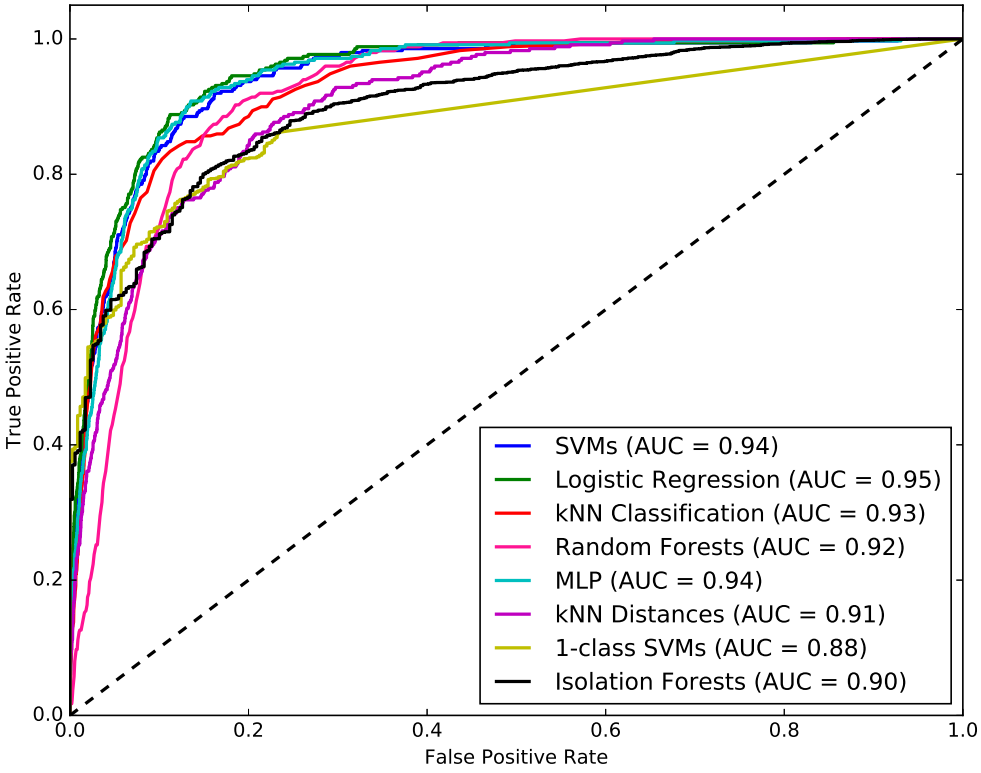

Figure A-10: The ROC curves for training on Alan Walks Wales data and testing on UT Dallas data using all features.

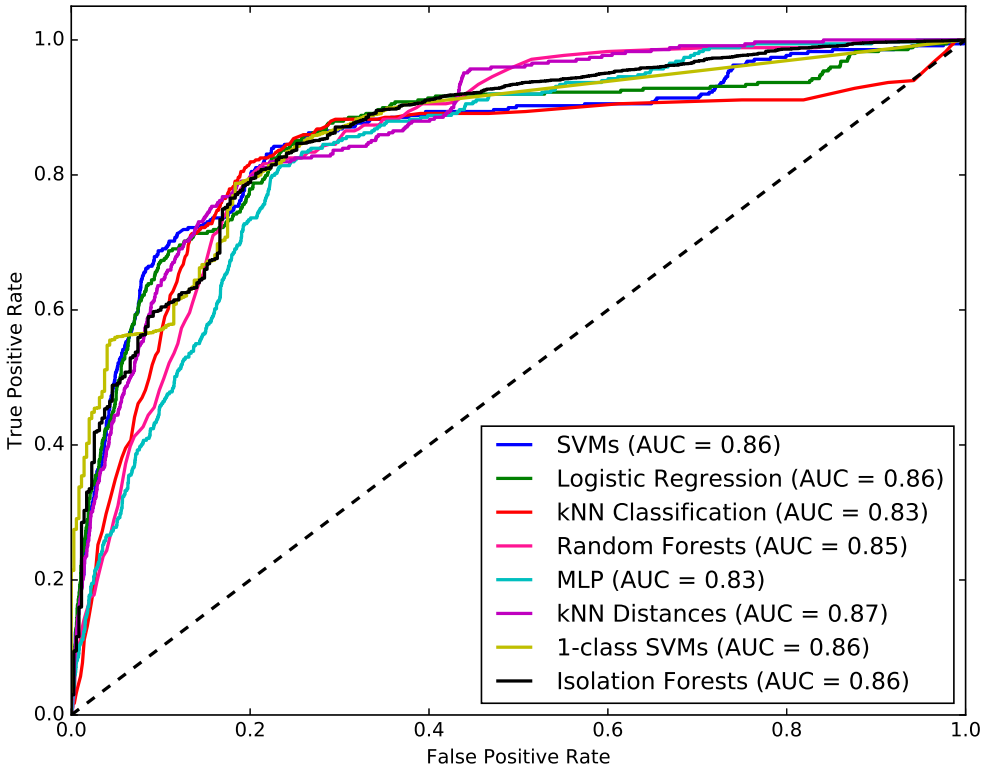

Figure A-11: The ROC curves for training on Alan Walks Wales data and testing on UT Dallas data using only ACC features.

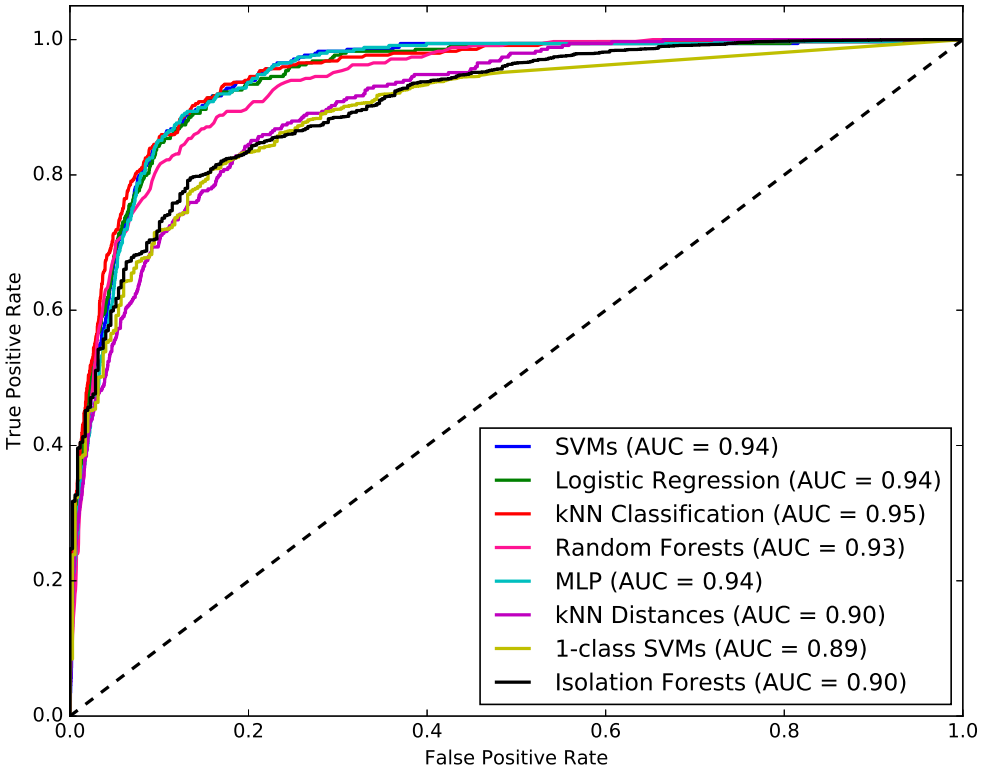

Figure A-12: The ROC curves for training on Alan Walks Wales data and testing on UT Dallas data using only EDA features.

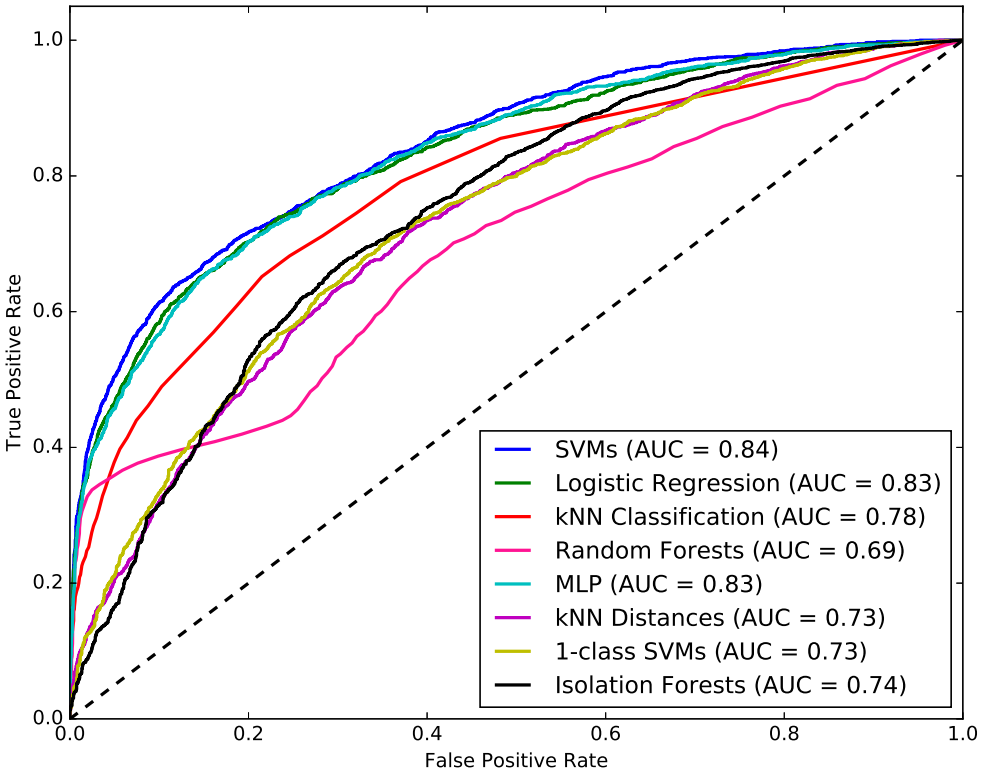

Figure A-13: The ROC curves for training on UT Dallas data and testing on Alan Walks Wales data using all features.

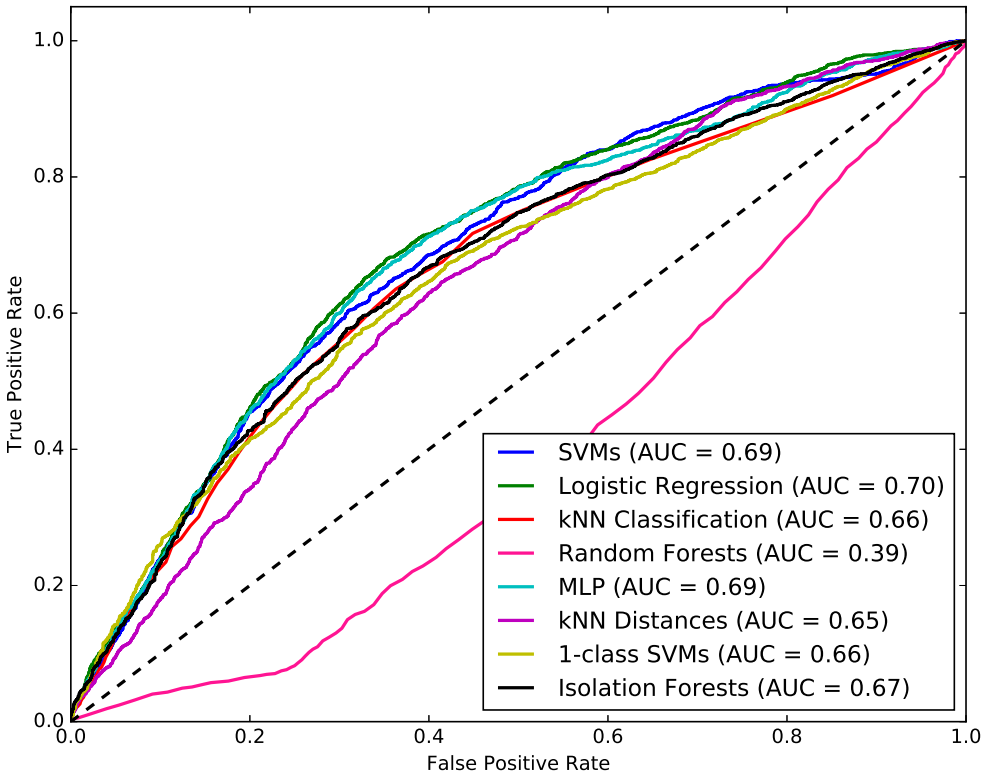

Figure A-14: The ROC curves for training on UT Dallas data and testing on Alan Walks Wales data using only ACC features.

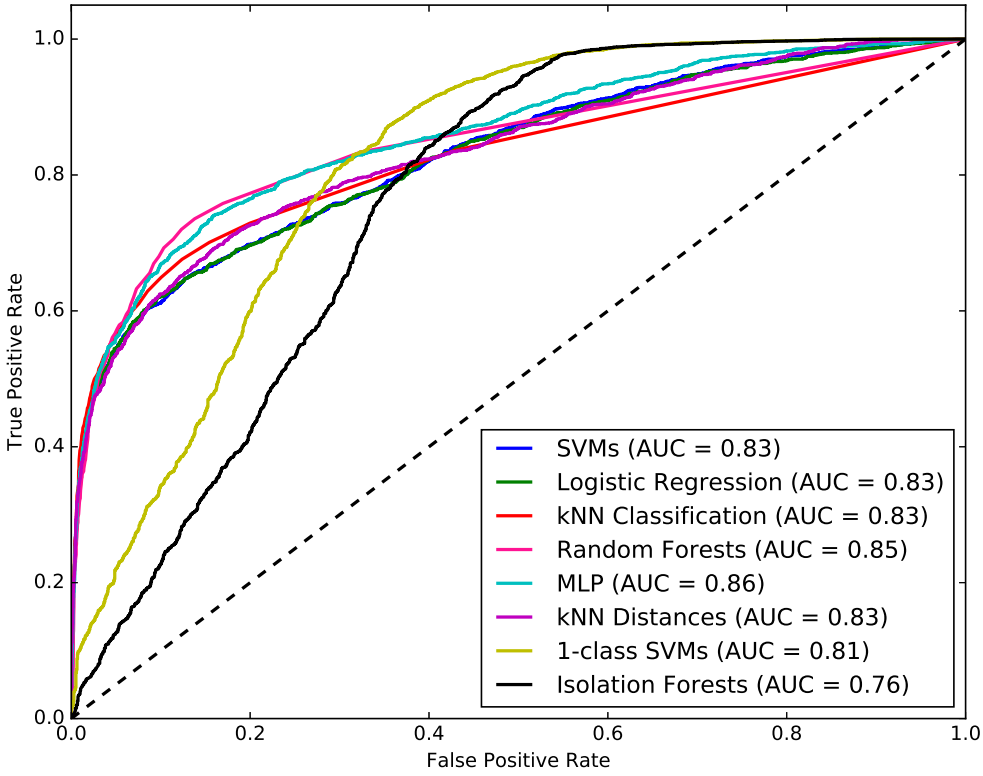

Figure A-15: The ROC curves for training on UT Dallas data and testing on Alan Walks Wales data using only EDA features.

A.4 The AUC tables shown in ISWC paper without feature selection

Table A.2: Alan Walks Wales resting data's in-sample prediction AUC using leave-one-subject-out cross-validation. The top five algorithms are supervised, while the bottom three are unsupervised. Highest value for the data set is shown in bold.

Alan Walks Wales resting data			
Algorithm	**All features**	**ACC only**	**EDA only**
Logistic regression	0.843	0.714	0.775
Multi-layer Perceptron	0.683	0.539	0.696
SVM	0.689	0.582	0.688
kNN classification	0.674	0.582	0.738
Random forest	0.747	0.583	0.712
1-class SVM	0.844	0.763	0.850
kNN distance	0.807	0.723	**0.898**
Isolation forest	0.804	0.711	0.885

Table A.3: Alan Walks Wales walking data's in-sample prediction AUC using leave-one-subject-out cross-validation. The top five algorithms are supervised, while the bottom three are unsupervised. Highest value for the data set is shown in bold.

Alan Walks Wales walking data			
Algorithm	**All features**	**ACC only**	**EDA only**
Logistic regression	0.807	0.649	0.796
Multi-layer Perceptron	0.788	0.663	0.777
SVM	0.798	0.684	0.782
kNN classification	0.740	0.641	0.776
Random forest	0.815	0.671	0.796
1-class SVM	0.768	0.683	0.760
kNN distance	0.774	0.705	**0.847**
Isolation forest	0.693	0.619	0.735

Table A.4: UT Dallas data's in-sample prediction AUC using leave-one-subject-out cross-validation. The top five algorithms are supervised, while the bottom three are unsupervised. Highest value for the data set is shown in bold.

UT Dallas data			
Algorithm	**All features**	**ACC only**	**EDA only**
Logistic regression	**0.941**	0.852	0.935
Multi-layer Perceptron	0.928	0.842	0.928
SVM	0.913	0.852	0.898
kNN classification	0.846	0.832	0.870
Random forest	0.935	0.852	0.937
1-class SVM	0.859	0.862	0.900
kNN distance	0.911	0.875	0.930
Isolation forest	0.909	0.878	0.900

Table A.5: Out-of-sample prediction AUC with EDA only features.

	Train/Test Set	
Algorithm	AWW/UTD	UTD/AWW
Logistic regression	0.943	0.846
Multi-layer Perceptron	0.943	**0.859**
SVM	0.944	0.822
kNN classification	**0.946**	0.827
Random forest	0.940	0.843
1-class SVM	0.891	0.847
kNN distance	0.913	0.854
Isolation forest	0.911	0.774