

UC Irvine

UC Irvine Electronic Theses and Dissertations

Title

pyEDA: An Open-Source and Versatile Feature Extraction Python Toolkit for Electrodermal Activity

Permalink

<https://escholarship.org/uc/item/3xq0084m>

Author

Aqajari, Seyed Amir Hossein

Publication Date

2021

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA,
IRVINE

pyEDA: An Open-Source and Versatile Feature Extraction Python Toolkit for
Electrodermal Activity

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Computer Engineering

by

Seyed Amir Hossein Aqajari

Dissertation Committee:
Professor Amir M. Rahmani, Chair
Professor Nikil Dutt
Professor Fadi Kurdahi

2021

DEDICATION

I'd like to dedicate this work to my parents and my siblings whom I haven't seen in years.

My beloved parents whom have always attentively supported me financially and morally with their fullest and truest attention to accomplish my work with truthful self-confidence.

My brother and my two sisters who have always been unconditionally by my side in my joys and sorrows.

I am thankful for having you all in my life and hope to see you soon!

TABLE OF CONTENTS

	Page
LIST OF FIGURES	iv
LIST OF TABLES	v
ACKNOWLEDGMENTS	vi
ABSTRACT OF THE DISSERTATION	viii
1 Introduction	1
2 Background	5
3 Proposed Pipeline Architecture	8
3.1 Pre-processing	9
3.2 Statistical Feature Extraction	9
3.3 Automatic Feature Extraction	11
4 Experimental Results	14
4.1 Dataset	14
4.2 Machine Learning based Classification Algorithms	15
4.3 Stress Models Accuracy	15
5 Conclusion	18
Bibliography	19

LIST OF FIGURES

	Page
3.1 Proposed Processing Pipeline Architecture for pyEDA	8
3.2 The architecture of the proposed autoencoder.	13
4.1 Validation accuracy on two different set of features using different machine learning algorithms.	16

LIST OF TABLES

	Page
1.1 Comparing the existing toolkits compared with pyEDA for preprocessing and feature extraction of EDA signals.	1
4.1 The summary of our results in compared with the state of the arts	16
4.2 The summary of extracted features from EDA in Schmidt et al. work	17

ACKNOWLEDGMENTS

First and foremost, I would like to thank to my advisor, Professor Amir M. Rahmani, for the continuous support of my research during the past two years. Without your assistance and dedicated involvement in every step throughout the process, this work would have never been accomplished.

Beside my advisor, I would also like to express my sincere gratitude to my committee member Professor Nikil Dutt not only for assigning their valuable time to review my work but also for his insightful feedback that pushed my work to a higher level.

My appreciation also extends to my committee member Professor Fadi Kurdahi, for assigning their priceless time to review my work.

I thank the Elsevier publishing company for allowing me to use the content of our original paper [1] published in the 12th International Conference on Ambient Systems, Networks and Technologies (ANT). Some text of this thesis is a reprint of the material as it appears on ANT'21.

I would also like to thank the Academy of Finland and the US National Science Foundation (NSF) to partially funding this work (through the SLIM project under the grants (316810, 316811) and the UNITE project under the grant SCC CNS-1831918 respectively).

REFEREED CONFERENCE PUBLICATIONS

pyEDA: An Open-Source Python Toolkit for Pre-processing and Feature Extraction of Electrodermal Activity

2021

The 12th International Conference on Ambient Systems, Networks and Technologies (ANT)

ABSTRACT OF THE DISSERTATION

pyEDA: An Open-Source and Versatile Feature Extraction Python Toolkit for
Electrodermal Activity

By

Seyed Amir Hossein Aqajari

Master of Science in Computer Engineering

University of California, Irvine, 2021

Professor Amir M. Rahmani, Chair

Electrodermal Activity (EDA), also known as Galvanic Skin Response (GSR), measures changes in perspiration by detecting the changes in electrical conductivity of skin. The changes in perspiration is one of the examples of physiological response to a stimulus such as stress, emotion, pain, etc. Previous studies have already shown that EDA is one of the leading indicators for a stimulus. However, the EDA signal itself is not trivial to analyze. To detect different stimuli in human subjects, variety of features are extracted from EDA signals such as the number of peaks, max peak amplitude, to name a few, showing the prevalence of this signal in bio-medical as well as ubiquitous and wearable computing research. In this paper, we present an open-source Python toolkit for EDA signal preprocessing and statistical and automatic feature extraction. To the best of our knowledge, this is the first effort for developing a versatile and generic tool to extract any number of automatic features from EDA signals. Our online toolkit is evaluated using different machine learning algorithms applied to Wearable Stress and Affect Detection (WESAD) dataset which is publicly available. Our results show that our proposed pipeline outperforms the state of the art accuracy using either statistical or automatic extracted features on a same dataset. Based on our results, in all of our four machine learning algorithms, we achieve a higher validation accuracy using automatic features compared with statistical features.

Chapter 1

Introduction

Physiological signals can be used to extract valuable information measuring the functional state of various physiological systems [2]. Many studies have validated the effect of different stimuli on the functioning of the physiological systems in the human body such as emotion, stress, and sleep [3, 4, 5, 6, 7, 8].

Table 1.1: Comparing the existing toolkits compared with pyEDA for preprocessing and feature extraction of EDA signals.

Existing Works	Statistical Features	Automatic Features	Generic	Python
pyEDA	✓	✓	✓	✓
TEAP	✓	×	×	×
PhysioLab	✓	×	×	×
ANSLAB	✓	×	×	×
NeuroKit	✓	×	×	✓
Pysiology	✓	×	×	✓

The Electrodermal Activity (EDA), also known as galvanic skin response (GSR), is one of these physiological signals widely used in biomedical and digital health research to detect certain stimuli in human subjects. EDA measures the changes in electrical conductivity of skin. These changes reflect the intensity of the individual's emotional state – or emotional arousal [9].

Numerous prediction models are constructed to predict a stimulus based on raw EDA signals [10, 11]. The EDA signals collected from commercially available devices are usually raw and have motion artifacts that are common in natural, uncontrolled settings that involve body gestures and movements. Therefore, the raw EDA itself cannot be used in these prediction models directly. First, several signal processing steps are needed to remove noise and extract the clean signal. Next, a variety of features are extracted from the clean EDA signal. These features can then be fed to machine learning algorithms to build prediction models detecting different types of stimuli.

Traditional methods extract statistical features such as the number of peaks, max peak amplitude, average, standard deviation, etc., from the EDA signal for prediction models. However, with rapid development of AI and machine learning algorithms, various automatic methods are implemented to extract automatic features from the signal using neural networks [12, 13]. In many situations, these methods outperform the traditional methods' performance in prediction accuracy. Needless to say, an open-source tool providing automatic and statistical set of features extracted from EDA signal can significantly facilitate the research studies in EDA signal processing. Its worth mentioning that these neural network techniques are not limited only to automatic feature extraction from the signals. Zargari et al. used combination of convolutional neural network and recurrent neural network to accurately track in-mouth nutrient sensors position [14]. Mehrabadi et al. used convolutional neural network to detect existence of COVID-19 in patients with ARDS [15]. Recently in [16], the author employed a deep neural network to predict daily cases of COVID-19 leveraged by Google trends. Ashrafiamiri et al. used deep neural networks to secure autonomous driving [17]. In [18], the author employed deep neural networks to implement a defense adversarial network. Recently, generative adversarial networks which consist of two different neural networks competing with each other are also employed for the purpose of signal translation or noise removal [19, 20].

The existing open-source tools for EDA signal processing only focus on the statistical features and do not take into account the embedding features extracted using automatic methods (Table 1.1). Soleymani *et al.* [21] develop a toolbox to extract features from different signals including EDA. The author build classifiers solely using statistical features, including: amplitude and number of peaks, mean, and standard deviation of the signal. Furthermore, there are toolboxes that provide integrated software. PhysioLab [22] and ANSLAB [23] are open-source tools for EDA analysis, which are implemented in Matlab. These tools aggregate the information extracted from different signals including EDA. However, the feature extraction module is limited to non-automated statistical features. Besides, researchers have implemented toolboxes for EDA analysis in Python [24, 25]. These toolboxes also have the same limitation since they also only consider statistical features including: number of peaks, amplitude, rise time and decay time. It should be noted that all of these existing toolboxes tailor the feature extraction phase to their target application (e.g., emotion recognition, pain assessment, etc.) making them application-dependent to a certain degree.

In this work, we present pyEDA [26], an open-source tool in Python to pre-process and filter EDA signals and extract statistical and any arbitrary number of automatic features from them. To the best of our knowledge, this is the first work presenting a versatile user-friendly open-source Python tool which can be used to extract any number of automatic features of EDA signals without the need to have a background in artificial neural networks and auto-encoders. Depending on the application, different statistical features might be conceded for acceptable prediction accuracy to a stimulus [27]. In other words, certain features to a stimulus may not be captured using statistical methods. For these reasons, we present a tool to automatically extract any number of automatic features highly correlated to any type of stimuli. Providing such a versatile and generic tool to extract automatic features of the EDA signal is valuable for the health science and technology community. We also demonstrate the efficacy of our tool by using the Wearable Stress and Affect Detection (WESAD) dataset [28] and a set of machine learning algorithms to evaluate both statistical and automatic features

extracted using pyEDA. Specifically, this work makes the following key contributions:

- Develop an open-source toolkit to extract any number of features of EDA signals.
- Build a scalable architecture to work with EDA signals with any arbitrary length.
- Enable the toolkit to perform generic automatic feature extraction.
- Evaluate the extracted features using a publicly available dataset on stress assessment and beating the state of the art accuracy.

The rest of this work is organized as follows. Section 2 briefly outlines the EDA background. Our proposed processing pipeline architecture is presented in Section 3. In Section 4 we evaluate our result using the WESAD data set. Finally, Section 5 concludes the work.

Chapter 2

Background

The autonomic nervous system automatically regulates involuntary physiologic processes including heart rate, blood pressure, respiration, digestion, and sexual arousal without a person's conscious effort [29]. Disorders of the autonomic nervous system can affect any body part or process. Autonomic disorders may be reversible or progressive [29]. Standard functional assessment of autonomic nervous system (ANS) activity on cardiovascular control relies on spectral analysis of heart rate variability (HRV) series. However, obtaining a reliable measure of sympathetic activity from HRV spectra has some difficulties which limits the exploitation of sympathovagal metrics. On the other hand, electrodermal activity (EDA) measurements have been demonstrated to provide a reliable quantifier of sympathetic dynamics [30].

To better understand how EDA is captured, it is helpful to study the physiological characteristics of the skin described in [9]. Sweat glands are small tubular structures of the skin producing sweat. Our body has approximately three million sweat glands having different density across the body. Sweat glands can be found in large numbers on the soles of the feet, the palms and fingers, and on the forehead and cheeks. They produce moisture through

pores towards the surface of the skin, whenever they are triggered. When the balance of positive and negative ions in this secreted fluid changes, the electrical current flows more readily. This results in decreased skin resistance, or in other words, increased skin conductance. Electrodermal Activity (EDA) is a term used for this change. EDA is also known as Galvanic Skin Response (GSR), Skin Conductance (SC), Electrodermal Response (EDR), and Psychogalvanic Reflex (PGR).

Although one of the main purposes of sweating is thermoregulation, sweating is also triggered whenever a person is exposed to a stimulus such as emotionally loaded images. This type of sweating is called emotional sweating. Sweat secretion, which reflects the changes in arousal, is driven unconsciously by the automatic nervous system (ANS) in order to meet behavioral requests. A number of commercially available devices (e.g., RespiBAN professional, Empatica E4, Fitbit Sense, and Shimmer3 GSR+) can be used to collect EDA signals.

According to [9], EDA signals consist of two main components: Skin Conductance Level (SCL) and Skin Conductance Response (SCR). The SCL changes slightly on a time scale of tens of seconds to minutes. Depending on hydration, skin dryness or automatic regulation of an individual respondent, the rising and declining SCL is continuously changing. SCL, which is also called the tonic level of EDA signal, can differ significantly across different individuals. Due to this, the actual tonic level on its own is not completely informative. SCR, which is also known as the phasic component of EDA, rides on top of the tonic changes and shows much faster alterations. Variations in the phasic component of a EDA signal are visible as EDA bursts or EDA peaks. The phasic component is sensitive to specific emotionally arousing stimulus events (event-related SCRs, ER-SCRs). These bursts can occur between 1-5 seconds after the onset of emotional stimuli. Quite the opposite, non-specific skin conductance responses (NS-SCRs) are not a consequence of any eliciting stimulus. These responses happen at a rate of 1-3 per minute spontaneously. There is a need for a flexible and usable processing toolchain that can be used to efficiently analyze these EDA signals. In the following, we

describe our pyEDA processing pipeline architecture that is designed to meet these needs.

Chapter 3

Proposed Pipeline Architecture

Figure 3.1 shows the proposed processing pipeline architecture for pyEDA to analyze the EDA data. There are two different stages in this pipeline: The pre-processing stage and the feature extraction stage. As shown in this figure, the pre-processing stage consists of two different modules. In the pre-processing stage, the signals are cleaned and prepared for the feature extraction stage. Then, the feature extraction stage uses two different procedures to extract the features from the pre-processed data. We use traditional manual statistical feature extraction as well as automatic feature extraction methods in our proposed architecture. In the following, we explain each stage of the pipeline in detail.

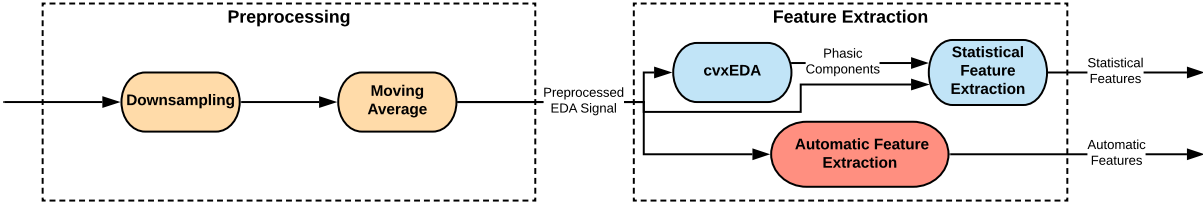


Figure 3.1: Proposed Processing Pipeline Architecture for pyEDA

3.1 Pre-processing

In this stage, we use down-sampling and moving averaging to pre-process the data. At the end of this stage, a pre-processed EDA signal is ready and accessible for further analysis and feature extraction.

The EDA data is usually sampled at much higher frequency than needed. Therefore, down-sampling is done to reduce memory footprint and processing time of the data with a negligible risk of losing important information in the signal. In the pre-processing stage, the raw EDA data is down-sampled to the lower sampling rate. Based on the studies conducted in [9], the EDA data can safely be down-sampled to 20 Hz or even less if the data originally was collected at 128 Hz.

A raw EDA signal varies before or after a peak. This is due to individual differences in the tonic component of EDA or due to the noise caused by movements or respiration artifacts [9]. After down-sampling the data, a moving average across a 1-second window is first used to smooth the data and reduce artifacts such as body gestures and movements, which are common in everyday settings.

3.2 Statistical Feature Extraction

The number of peaks, the mean of EDA, and the max peak amplitude are three statistical features extracted in our pipeline. Calculating the mean of EDA is straightforward. To calculate the other two features, we need to extract the EDA peaks that are induced by eliciting stimulus. A number of signal processing steps are required to derive EDA peaks that are a consequence of eliciting stimulus [9]:

1. The phasic component is extracted from the pre-processed EDA signal.

2. A low-pass Butterworth filter is applied on the phasic data to remove line noise. A cutoff frequency of 5 Hz divided by sampling rate is typically used.
3. Onsets and offsets are identified from the phasic data.
4. The maximum amplitude value within each onset-offset is considered as a peak if and only if its difference with the amplitude value at onset is higher than the threshold, which typically is $0.005 \mu S$.

One of the major tasks in analyzing the EDA signal is to correctly extract the phasic component of the signal from the original signal. In this paper, we use the cvxEDA algorithm to decompose the original signal into a sparse phasic component and a smooth tonic component. The cvxEDA algorithm is a novel algorithm presented in [31] for the EDA analysis based on maximum a posteriori probability, convex optimization, and sparsity. This algorithm has a desirable capability of properly describing the activity of the autonomic nervous system in response to affective stimulation. This model describes the recorded EDA as the sum of three terms: the phasic component, the tonic component, and an additive white Gaussian noise term incorporating model prediction errors, measurement errors, and artifacts. We use this algorithm to extract the phasic component of the pre-processed EDA signal for further analysis.

To calculate the number of peaks corresponding to events related to SCRs, we need to find the maximum amplitude value within each onset-offset window in the original EDA signal [9]. Onsets are all the points in which the phasic component of the signal crosses above the onset threshold, which is typically $0.01 \mu S$. To find the corresponding offset of the computed onset, we check for the point in which the phasic component of the EDA signal crosses below the offset threshold, which is typically $0 \mu S$. For each window, the time difference between its onset and offset needs to be above the duration threshold of 1s. Any peaks before this duration threshold are considered as nonspecific skin conductance responses and does not

need to be extracted.

The clean EDA data, the peaks, and the tonic and phasic components of signal can be used to easily extract extra statistical features if needed.

3.3 Automatic Feature Extraction

Feature extraction becomes increasingly important when the data is high dimensional. There have been several studies which attempt to create classification models based on statistical features extracted from physiological signals such as EDA. However, there is no study suggesting which specific set of statistical features is informative for any type of EDA analytics regardless of the final application (e.g., assessing stress, emotion, pain, risk of seizure, etc.). Therefore, using automatic feature extraction and selection to find the most important set of features for any type of application can be significantly important.

An autoencoder is a type of unsupervised artificial neural network which is used to learn efficient data coding [32, 33]. The aim of an autoencoder is to efficiently learn how to compress and encode the data to reduce the dimensional representation. This lower dimensional representation can be regarded as an abstract set of features of the original high dimensional data. An autoencoder consists of two different parts: the encoder and the decoder. These two parts can be defined as two different functions as follows:

$$\phi : \mathcal{X} \rightarrow \mathcal{Y} \tag{3.1}$$

$$\psi : \mathcal{Y} \rightarrow \mathcal{X}' \tag{3.2}$$

Given one hidden layer for encoder and decoder parts in the simplest case: Function (3.1) can be defined as $\mathbf{y} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$. σ is an element-wised activation function such as sigmoid.

\mathbf{W} is a weight matrix and \mathbf{b} is a bias vector of the encoder part. \mathbf{x} is $\in \mathcal{X}$ and \mathbf{y} is $\in \mathcal{Y}$. Function (3.2) can be defined as $\mathbf{x}' = \sigma(\mathbf{W}'\mathbf{y} + \mathbf{b}')$. σ is an element-wised activation function such as sigmoid. \mathbf{W}' is a weight matrix and \mathbf{b}' is a bias vector of the decoder part. \mathbf{x}' is $\in \mathcal{X}'$ and \mathbf{y} is $\in \mathcal{Y}$.

Autoencoders are trained to minimize the following reconstruction loss:

$$loss(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|^2 \tag{3.3}$$

Weights and biases are initialized randomly, and then iteratively updated based on the reconstruction loss computed in (3.3) during training through Backpropagation. The image \mathbf{y} represents a latent code or latent representation of the input \mathbf{x} . It can be directly used as the set of features of input \mathbf{x} for classification.

Figure 3.2 shows the architecture of the autoencoder implemented in our pipeline for automatic feature extraction. First, a linear layer (L1) is used to downsample the input EDA signal with Input_Shape length to a length that is the closest power of 2 (CP2). This was done to make the model scalable to an arbitrary input size. The encoder half of the network consists of three 1-D convolutional layers (C1, C2, and C3) and a linear layer (L2) which flatten and downsamples the input vector to a lower-dimensional latent vector. The number of dimensions of this latent vector (Feature_Size) corresponds to the number of automatic features extracted and was set prior to training the network. The decoder half of the network consists of three 1-D de-convolutional layers (DeC1, DeC2, and DeC3) to reconstruct the input signal from the latent vector. A final linear layer (L3) is then used to flatten and reconstruct the signal to its original dimension. Both encoder and decoder networks have ReLU (Rectified Linear Unit) activation between layers.

It is worth mentioning that the number of extracted features are fed to our tool as an input

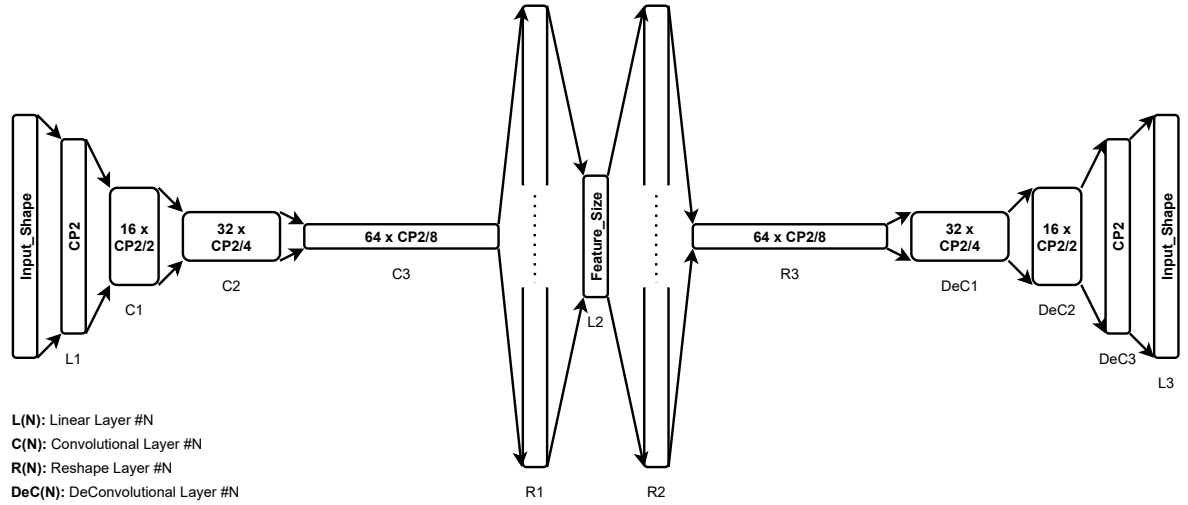


Figure 3.2: The architecture of the proposed autoencoder.

parameter before training the model (in this paper the number of features is 64). We use PyTorch library to implement our autoencoder architecture. PyTorch is an open source machine learning library based on the Torch library, used for applications such as computer vision and natural language processing [34].

Chapter 4

Experimental Results

4.1 Dataset

Wearable Stress and Affect Detection (WESAD) is a publicly available multimodal dataset for stress and affect detection [28]. In this data set, physiological and motion data are recorded from Empatica E4 and Raspbian professional devices from 15 subjects during a lab study.

The goal of this dataset is to elicit three affective states (neutral, stress, amusement) in the participants. There are two different versions of the study protocol in this dataset. These protocols consist of six different tasks labeled as Baseline, Amusement, Medi I, Stress, Rest, and Medi II. They distinguish two different classification tasks based on these protocols. First, they define a three-class problem: baseline vs. stress vs. amusement. Second, they define a binary classification: baseline vs. stress. In this paper, we focus on creating a binary classification to detect stress. We consider EDA data in the Baseline section labeled as "not-stressed" (0), and EDA data in the Stress section labeled as "stressed" (1) to create our model. Our model is created based on the EDA data collected from Empatica E4 wristband

[35].

4.2 Machine Learning based Classification Algorithms

To demonstrate the efficacy of the pyEDA toolkit, we use it to build machine learning based stress and affect models using the WESAD dataset and evaluate the performance of our extracted features. We use four different machine learning algorithms: (1) K-nearest-neighbor (kNN) with k between 1 to 10, (2) Naïve Bayes Gaussian classifier, (3) Random Forest with depth between 1 to 10, and (4) support vector machine (SVM). The kNN method uses k number of nearest data-points and predicts the result based on a majority vote [36]. The Naïve Bayes Gaussian classifier predicts the result based on the probabilities of each feature's Gaussian distribution [37]. The SVM tries to find the best hyper-plane to divide the data points into different classes [38]. The Random Forest classifier fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting [39]. We use the scikit-learn software for classification and prediction. The scikit-learn is an open-source machine learning library for the Python programming language [40].

4.3 Stress Models Accuracy

We build our stress model based on four different machine learning algorithms (kNN, Naive Bayes, Random Forest, and SVM). We train two different models based on the input features for each machine learning algorithm and report their accuracy: (1) The first model trains the data using only statistical features. (2) The second model trains the data using only automatic features. We consider 30 percent of the data (5 subjects) as test and the rest as training. Figure 4.1 shows the accuracy of the four different classifiers used based on two

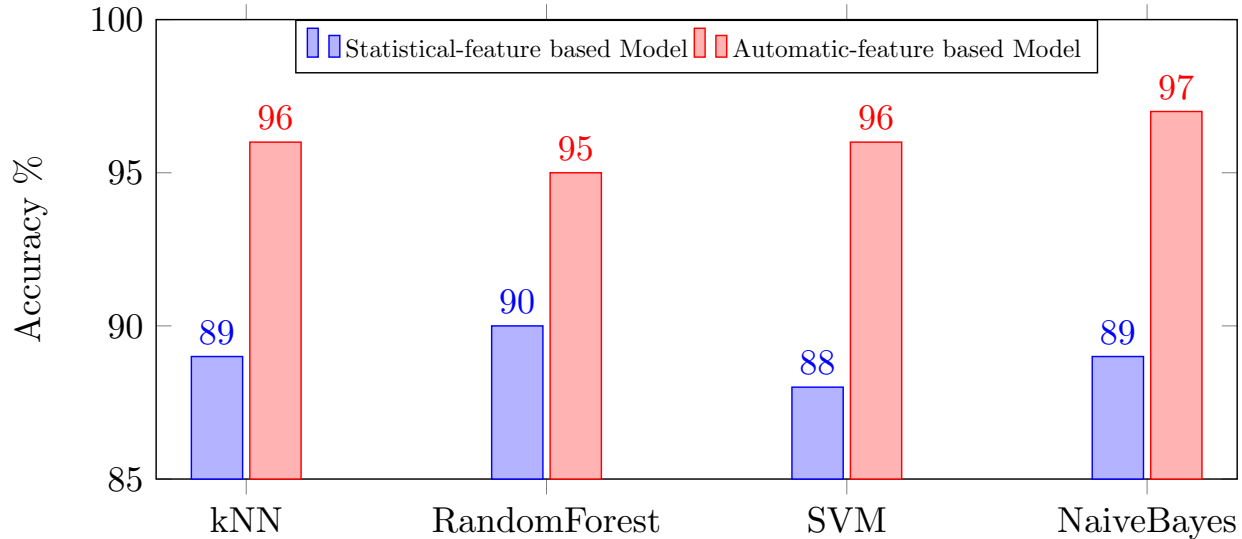


Figure 4.1: Validation accuracy on two different set of features using different machine learning algorithms.

different set of features. The best accuracy for the Statistical-feature based model belongs to the random forest with depth equals to 1, which is 90%. The best accuracy for the automatic-feature based model belongs to the NaiveBayes classifier which is equal to 97%. In [28], they achieve the accuracy of 79.71% using AdaBoost classifier for EDA signals collected from Empatica E4 wristband. Table 4.1 shows the summary of our results in compared with the state of the art. The results show that our proposed pipeline outperforms their method in extracting related features for creating stress models (for both automatic and statistical features).

Table 4.1: The summary of our results in compared with the state of the arts

Existing works	ML algorithm	Features	Accuracy
Schmidt et al. [28]	AdaBoost	statistical features	79.71%
pyEDA	RandomForest	statistical features	90%
pyEDA	NaiveBayes	automatic features	97%

Table 4.2 shows the statistical features extracted from EDA signal in Schmidt et al. work.

According to this table, they are extracting around 14 different statistical features. However, our model still beats their accuracy using only three statistical features. This indicates that

Table 4.2: The summary of extracted features from EDA in Schmidt et al. work

Statistical Extracted Features
Mean, STD, Min and Max value, Slope and Dynamic range, Mean and STD of SCR/SCL, correlation between SCL and time, sum of SCR startle magnitudes and response durations, and Area under the identified SCRs

increasing the number of statistical features extracted from the signal without any proper feature selection algorithm might mislead the model and reduces the accuracy of the model.

Our results show that, in all four machine learning algorithms, we achieve a higher accuracy using automatically extracted features compared with statistical features. To detect other types of stimulus than stress, one might need to add some extra statistical features with proper feature selection methods to achieve an acceptable accuracy. However, automatic feature extraction module in our tool can still be used to extract the most important features regardless of the stimulus. This is the main advantage of our open-source tool in comparison with other tools for EDA feature extraction.

Chapter 5

Conclusion

In this work, we presented a user-friendly open-source toolkit in Python, to extract statistical and automatic features from EDA data. To the best of our knowledge, this is the first work presenting an versatile open-soruce tool (pyEDA) to extract automatic features for EDA data. The pyEDA employs autoencoders to automatically extract any arbitrary number of features representing the lower dimensional representation of the input data. These features can directly be used in machine learning algorithms for classification and stimulus detection. There is no previous study claiming what statistical features of EDA data are the best features to detect different types of stimuli. Different types of stimuli can have different effects on EDA signal, therefore the type of statistical features to extract from the signal can be different in each scenario. As a result, presenting a toolkit which is able to extract both statistical and automatic features of EDA data can facilitate and accelerate research in EDA signal analysis.

Bibliography

- [1] Seyed Amir Hossein Aqajari, Emad Kasaeyan Naeini, Milad Asgari Mehrabadi, Sina Labbaf, Nikil Dutt, and Amir M Rahmani. pyeda: An open-source python toolkit for pre-processing and feature extraction of electrodermal activity. *Procedia Computer Science*, 184:99–106, 2021.
- [2] BIOMEDIKAL.IN, [Online]. <http://biomedikal.in/2011/05/important-physiological-signals-in-the-body/>. Accessed: Oct-2020.
- [3] Lan Li and Ji-hua Chen. Emotion recognition using physiological signals. In *International Conference on Artificial Reality and Telexistence*, pages 437–446. Springer, 2006.
- [4] Lin Shu, Jinyan Xie, Mingyue Yang, Ziyi Li, Zhenqi Li, Dan Liao, Xiangmin Xu, and Xinyi Yang. A review of emotion recognition using physiological signals. *Sensors*, 18(7):2074, 2018.
- [5] Hee Jeong Han, Sina Labbaf, Jessica L Borelli, Nikil Dutt, and Amir M Rahmani. Objective stress monitoring based on wearable sensors in everyday settings. *Journal of Medical Engineering & Technology*, 44(4):177–189, 2020.
- [6] Ali Rostami, Vaibhav Pandey, Nitish Nag, Vesper Wang, and Ramesh Jain. Personal food model. In *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, page 4416–4424, New York, NY, USA, 2020. Association for Computing Machinery.
- [7] Milad Asgari Mehrabadi, Iman Azimi, Fatemeh Sarhaddi, Anna Axelin, Hannakaisa Niela-Vilén, Saana Myllyntausta, Sari Stenholm, Nikil Dutt, Pasi Liljeberg, and Amir M Rahmani. Sleep tracking of a commercially available smart ring and smartwatch against medical-grade actigraphy in everyday settings: Instrument validation study. *JMIR mHealth and uHealth*, 2020.
- [8] Hannakaisa Niela-Vilén, Jennifer Auxier, Eeva Ekholm, Fatemeh Sarhaddi, Milad Asgari Mehrabadi, Aysan Mahmoudzadeh, Iman Azimi, Pasi Liljeberg, Amir M Rahmani, and Anna Axelin. Pregnant women’s daily patterns of well-being before and during the covid-19 pandemic in finland: Longitudinal monitoring through smartwatch technology. *PloS one*, 16(2):e0246494, 2021.
- [9] Galvanic Skin Response. The complete pocket guide. *Imotions–Biometric Research, Simplified*, 2017.

- [10] Pamela Zontone, Antonio Affanni, Riccardo Bernardini, Alessandro Piras, and Roberto Rinaldo. Stress detection through electrodermal activity (eda) and electrocardiogram (ecg) analysis in car drivers. In *2019 27th European Signal Processing Conference (EU-SIPCO)*, pages 1–5. IEEE, 2019.
- [11] Busra T Susam, Murat Akcakaya, Hooman Nezamfar, Damaris Diaz, Xiaojing Xu, Virginia R de Sa, Kenneth D Craig, Jeannie S Huang, and Matthew S Goodwin. Automated pain assessment using electrodermal activity data and machine learning. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 372–375. IEEE, 2018.
- [12] Dian Yu and Shouqian Sun. A systematic exploration of deep neural networks for eda-based emotion recognition. *Information*, 11(4):212, 2020.
- [13] Beanbonyka Rim, Nak-Jun Sung, Sedong Min, and Min Hong. Deep learning in physiological signal data: A survey. *Sensors*, 20(4):969, 2020.
- [14] Amir Hosein Afandizadeh Zargari, Manik Dautta, Marzieh AshrafiAmiri, Minjun Seo, Peter Tseng, and Fadi Kurdahi. Newertrack: MI-based accurate tracking of in-mouth nutrient sensors position using spectrum-wide information. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(11):3833–3841, 2020.
- [15] Milad Asgari Mehrabadi, Seyed Amir Hossein Aqajari, Iman Azimi, Charles A Downs, Nikil Dutt, and Amir M Rahmani. Detection of covid-19 using heart rate and blood pressure: Lessons learned from patients with ards. *arXiv preprint arXiv:2011.10470*, 2020.
- [16] Milad Asgari Mehrabadi, Nikil Dutt, and Amir M Rahmani. The causality inference of public interest in restaurants and bars on daily covid-19 cases in the united states: Google trends analysis. *JMIR public health and surveillance*, 7(4):e22880, 2021.
- [17] Marzieh AshrafiAmiri, Sai Manoj Pudukotai Dinakarrao, Amir Hosein Afandizadeh Zargari, Minjun Seo, Fadi Kurdahi, and Houman Homayoun. R2ad: Randomization and reconstructor-based adversarial defense for deep neural networks. In *2020 ACM/IEEE 2nd Workshop on Machine Learning for CAD (MLCAD)*, pages 21–26. IEEE, 2020.
- [18] Amir Hosein Afandizadeh Zargari, Marzieh AshrafiAmiri, Minjun Seo, Sai Manoj Pudukotai Dinakarrao, Mohammed E Fouda, and Fadi Kurdahi. Captive: Constrained adversarial perturbations to thwart ic reverse engineering. *arXiv preprint arXiv:2110.11459*, 2021.
- [19] Seyed Amir Hossein Aqajari, Rui Cao, Amir Hosein Afandizadeh Zargari, and Amir M Rahmani. An end-to-end and accurate ppg-based respiratory rate estimation approach using cycle generative adversarial networks. *arXiv preprint arXiv:2105.00594*, 2021.
- [20] Amir Hosein Afandizadeh Zargari, Seyed Amir Hossein Aqajari, Hadi Khodabandeh, Amir M Rahmani, and Fadi Kurdahi. An accurate non-accelerometer-based ppg motion artifact removal technique using cyclegan. *arXiv preprint arXiv:2106.11512*, 2021.

- [21] Mohammad Soleymani, Frank Villaro-Dixon, Thierry Pun, and Guillaume Chanel. Toolbox for emotional feature extraction from physiological signals (teap). *Frontiers in ICT*, 4:1, 2017.
- [22] John Edison Muñoz, Elvio Rubio Gouveia, Mónica S Cameirão, and Sergi Bermúdez i Badia. Physiolab-a multivariate physiological computing toolbox for ecg, emg and eda signals: a case of study of cardiorespiratory fitness assessment in the elderly population. *Multimedia Tools and Applications*, 77(9):11521–11546, 2018.
- [23] Jens Blechert, Peter Peyk, Michael Liedlgruber, and Frank H Wilhelm. Anslab: Integrated multichannel peripheral biosignal processing in psychophysiological science. *Behavior Research Methods*, 48(4):1528–1545, 2016.
- [24] Dominique Makowski, Tam Pham, Zen, Jan C. Brammer, Duy Le, Hung Pham (Pham Tien Hùng), François Lesspinasse, Chuan-Peng Hu, and Christopher Schölzel. neuropsychology/neurokit: 0.0.6, January 2020.
- [25] Giulio Gabrieli, Atiqah Azhari, and Gianluca Esposito. Pysiology: A python package for physiological feature extraction. In *Neural Approaches to Dynamics of Signal Exchanges*, pages 395–402. Springer, 2020.
- [26] pyEDA, [Online]. <https://github.com/HealthSciTech/pyEDA>. Accessed: Oct-2020.
- [27] Heera Lee and Andrea Kleinsmith. Public speaking anxiety in a real classroom: Towards developing a reflection system. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2019.
- [28] Philip Schmidt, Attila Reiss, Robert Duerichen, Claus Marberger, and Kristof Van Laerhoven. Introducing wesad, a multimodal dataset for wearable stress and affect detection. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, pages 400–408, 2018.
- [29] MerckManuals.com, [Online]. <https://www.merckmanuals.com/home/brain,-spinal-cord,-and-nerve-disorders/autonomic-nervous-system-disorders/overview-of-the-autonomic-nervous-system>. Accessed: Nov-2021.
- [30] Shadi Ghiasi, Alberto Greco, Riccardo Barbieri, Enzo Pasquale Scilingo, and Gaetano Valenza. Assessing autonomic function from electrodermal activity and heart rate variability during cold-pressor test and emotional challenge. *Scientific reports*, 10(1):1–13, 2020.
- [31] Alberto Greco, Gaetano Valenza, Antonio Lanata, Enzo Pasquale Scilingo, and Luca Citi. cvxeda: A convex optimization approach to electrodermal activity processing. *IEEE Transactions on Biomedical Engineering*, 63(4):797–804, 2015.
- [32] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

- [33] Quoc V Le et al. A tutorial on deep learning part 2: Autoencoders, convolutional neural networks and recurrent neural networks. *Google Brain*, pages 1–20, 2015.
- [34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.
- [35] empatica-e4, [Online]. <https://www.empatica.com/research/e4/>. Accessed: Oct-2020.
- [36] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.
- [37] David J Hand and Keming Yu. Idiot’s bayes—not so stupid after all? *International statistical review*, 69(3):385–398, 2001.
- [38] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [39] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.