Accurate, Energy-efficient, and Secure Machine Learning Models: Applications to Smart Healthcare

Ayten Ozge Akmandor

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Electrical Engineering
Adviser: Professor Niraj K. Jha

June 2020
Abstract

Machine learning (ML) algorithms automate the data-label mapping and decision-making processes. The output of the ML algorithms determines the next step in real-world applications. If the ML output is diagnosis of a disease, e.g., in smart healthcare applications, then the next step is the treatment options. On the other hand, if the ML output is the traffic condition, e.g., in smart city applications, then the next step is traffic regulation. It is desirable to have ML-based systems that are simultaneously accurate, secure, energy-efficient, low-cost, silently operable, maintainable, customizable, low-delay, and scalable. However, there are tradeoffs among these design goals that impact the complexity of the overall system. This thesis focuses on the accuracy, security, and energy-efficiency objectives.

We first target the accuracy objective with a new dual-space (feature and semantic space) classification approach: SECRET. While traditional supervised learning approaches operate in the feature space only, SECRET utilizes both feature and semantic spaces in the classification process. It incorporates class affinity and dissimilarity information into the decision process using the semantic space. This property enables SECRET to make informed decisions on class labels, thus enhancing its overall classification performance. By demonstrating the significant benefits of the semantic space, SECRET opens up a new research direction for the machine learning community.

Next, we introduce an automatic stress detection and alleviation system: SoDA. SoDA takes advantage of emerging wearable medical sensors to continuously monitor human stress levels and mitigate stress as it arises. It performs stress detection and alleviation in a user-transparent manner. When it detects stress, SoDA employs a stress alleviation technique in an adaptive manner based on the stress response of the user.

Furthermore, we present machine learning models that are built with smartphone and wearable medical sensor data and enable the smartphone to understand us. We
call this system your smartphone understands you (YSUY). By understanding our physical, mental, and emotional states, YSUY promotes quality of life of the users by assessing their states and shining light on fundamental human-centric needs.

Finally, we introduce a simultaneously smart, secure, and energy-efficient Internet-of-Things (IoT) sensor architecture: SSE. In SSE, we use inference in the compressively-sensed domain and transmit data to the base station when an event of interest occurs. Since on-sensor compression and inference drastically reduce the amount of data that need to be transmitted, we are able to reduce the IoT sensor energy by two-to-three orders of magnitude. A small part of this energy bonus is used to carry out encryption and hashing to ensure data confidentiality and integrity. Overall, we achieve smartness through decision-making inferences, security through encryption and hashing, and energy efficiency through both compression and decision-making inferences. By performing data compression and machine learning inference on the IoT sensor node, the SSE approach not only enables the IoT system to push signal processing and decision-making to the extreme of the edge-side (i.e., the sensor node), but also solves the data security and energy efficiency problems simultaneously.
Acknowledgments

First of all, I would like to express my deepest gratitude to my parents, Serap and Sinan Akmandor. I would not have been able to finish my Ph.D. study without their unconditional and continuous support, encouragement, and love. My parents’ creativity, positive attitude, strong problem solving skills, ambition, hard work, and strong moral and ethical values have always been my loadstar throughout my life. I am also greatly thankful to my brother, Unver Akmandor. His strong, generous, caring and hardworking attitude and inspirational speeches have had an immensely positive effect on my life. I cannot imagine having better grandparents than Sabiha and Cemal Parlak and Ayten and Neset Akmandor. They have always been the greatest source of joy and unforgettable moments. Ph.D. was a long journey. Although I lost Cemal Parlak on 12/25/2018 and Sabiha Parlak on 06/26/2019, I believe people live as long as we can keep them in our memories. You have been and will be a great inspiration to me. I will always remember you and cherish our memories throughout my life.

I would like to specially thank my Ph.D. adviser Prof. Niraj K. Jha. In 2015, the first time we met at the picnic of Electrical Engineering Department in the beginning of the school year, before I introduced myself, Prof. Jha impressed me by telling me what my areas of interest were based on my Ph.D. application. That was just the start of my Ph.D. and a very small portion of Prof. Jha’s wonderful character. Throughout my Ph.D., Prof. Jha taught me critical thinking, problem-defining, problem-solving, paper writing, and making an effective presentation. Each of our weekly meetings was a rewarding experience for me as Prof. Jha showed me different angles of the problems I was working on while providing full support and encouragement. I am totally indebted to Prof. Jha for his guidance, patience, support, and encouragement throughout my Ph.D.

I would like to also thank to Prof. Naveen Verma, Prof. Prateek Mittal, Prof. James Stur, Prof. Ruby Lee, Prof. Sun-Yuan Kung, Prof. Yuxin Chen, Prof.
Andrej Kosmrlj, Prof. Claire Gmachl, Prof. Sharad Malik, Prof. Kaushik Sengupta, and Prof. Hakan Tureci for all their support and valuable feedback. I would like to also extend my thanks to Colleen Conrad, Roelie Abdi, Lidia Stokman, Katie Watson, and Lori Bailey for their kind assistance and administrative support.

I would like to thank to all the groupmates Debajit Bhattacharya, Arsalan Mosenia, Jie (Lucy) Lu, Abdullah Guler, Ye Yu, Xiaoliang Dai, Hongxu Yin, Shayan Hassantabar, Tanujay Saha, Prerit Terway, Kenza Hamidouche, Wenhan Xia, Sayeri Lala, and Kai-Chieh Hsu for all the fun memories.

I am also very grateful to my friends Burcin Cakir, Levent Aygun, Abdullah Guler, Tugce Tunalilar, Yasin Kaya, Murat Ozatay, Mert Al, Semih Yagli, Xin Sun, Junhwan Alexander Bae, Arjun Nitin Bhagoji, Ting-Jung Chang, Elif Cuce, Akshay Krishna, Meng Ma, Yeohye Im, Kevin Villegas Rosales, Sophie Chopin, Sinem Uysal, Cemil Dibek, Sumegha Garg, Pranav Mundada, and Jyotesh Singh for filling this Ph.D. journey with lots of unforgettable and happy moments.

I would like to express my deepest gratitude to my first grade teacher, Nihal Sav for identifying my huge interest in science and technology at a very early stage and feeding my curiosity by showing me the news articles on the latest advancements followed by enlightening discussions.

I would like to extend my sincere thanks to my high school teacher Ayse Tekin for her vision, rewarding experience, and guidance. With her support throughout my high school years, I got into my first choice university.

Not only the humans, but also several animals had a very special impact on my life. First of all, Siyah, your making sure of me getting into school bus everyday was truly adorable and amazing. I learnt from you the importance of taking responsibility and following through. Taygir, you were very gentlemanly, caring, and joyful. I learnt the importance of having a good attitude from you. Rex, you were a very smart and loving dog. I will never forget you. Paris, your strong and dominant character, Zeytin, your
love for food, Tilkican, your energic behavior, and Dost, your adventurous character are totally amazing. Finally, I am very grateful to all of the Princeton squirrels and deer. Everytime I see each of you, my heart fills with joy and happiness. Thanks a lot.

The work presented in this thesis was supported by NSF under Grant Number CNS-1617628, CNS-1617640, and an IBM summer internship.
### Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-D</td>
<td>Two-dimensional</td>
</tr>
<tr>
<td>AAMI</td>
<td>The Association for the Advancement of Medical Instrumentation</td>
</tr>
<tr>
<td>ACC</td>
<td>Classification accuracy</td>
</tr>
<tr>
<td>ACC-s</td>
<td>Accelerometer within a smartphone</td>
</tr>
<tr>
<td>ACC-w</td>
<td>Accelerometer within a wearable medical sensor</td>
</tr>
<tr>
<td>AES</td>
<td>Advanced Encryption Standard</td>
</tr>
<tr>
<td>AI</td>
<td>Artificial intelligence</td>
</tr>
<tr>
<td>BLE</td>
<td>Bluetooth Low Energy</td>
</tr>
<tr>
<td>BO</td>
<td>Blood oximeter</td>
</tr>
<tr>
<td>BP</td>
<td>Blood pressure</td>
</tr>
<tr>
<td>BPF</td>
<td>Bandpass filter</td>
</tr>
<tr>
<td>BVP</td>
<td>Blood volume pulse</td>
</tr>
<tr>
<td>CBOW</td>
<td>Continuous Bag-of-Words</td>
</tr>
<tr>
<td>CSP</td>
<td>Compressed signal processing</td>
</tr>
<tr>
<td>DDoS</td>
<td>Distributed denial-of-service</td>
</tr>
<tr>
<td>DWT</td>
<td>Discrete Wavelet Transform</td>
</tr>
<tr>
<td>ECG</td>
<td>Electrocardiogram</td>
</tr>
<tr>
<td>EEG</td>
<td>Electroencephalogram</td>
</tr>
<tr>
<td>EMG</td>
<td>Electromyogram</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>FN</td>
<td>False negative</td>
</tr>
<tr>
<td>FNR</td>
<td>False negative rate</td>
</tr>
<tr>
<td>FP</td>
<td>False positive</td>
</tr>
<tr>
<td>FPR</td>
<td>False positive rate</td>
</tr>
<tr>
<td>FoG</td>
<td>Freezing of gait</td>
</tr>
<tr>
<td>GP</td>
<td>Gaussian Process</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Full Form</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>GPS</td>
<td>Global positioning system</td>
</tr>
<tr>
<td>GSR</td>
<td>Galvanic skin response</td>
</tr>
<tr>
<td>GYR</td>
<td>Gyroscope</td>
</tr>
<tr>
<td>HF</td>
<td>High frequency</td>
</tr>
<tr>
<td>HR</td>
<td>Heart rate</td>
</tr>
<tr>
<td>IAPS</td>
<td>International Affective Picture System</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet-of-Things</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>ISM</td>
<td>Industrial Scientific Medical</td>
</tr>
<tr>
<td>kNN</td>
<td>k-nearest neighbor</td>
</tr>
<tr>
<td>LBT</td>
<td>Logic-based therapy</td>
</tr>
<tr>
<td>LED</td>
<td>Light-emitting diode</td>
</tr>
<tr>
<td>LF</td>
<td>Low frequency</td>
</tr>
<tr>
<td>MAC</td>
<td>Multiply-accumulate</td>
</tr>
<tr>
<td>MAP</td>
<td>Mean arterial pressure</td>
</tr>
<tr>
<td>MICS</td>
<td>Medical Implant Communication Service</td>
</tr>
<tr>
<td>ML</td>
<td>Machine learning</td>
</tr>
<tr>
<td>MLP</td>
<td>Multi-layer perceptron</td>
</tr>
<tr>
<td>N/P</td>
<td>Not provided</td>
</tr>
<tr>
<td>NLP</td>
<td>Natural language processing</td>
</tr>
<tr>
<td>PCA</td>
<td>Principal component analysis</td>
</tr>
<tr>
<td>PIR</td>
<td>Passive infrared</td>
</tr>
<tr>
<td>PPG</td>
<td>Photoplethysmogram</td>
</tr>
<tr>
<td>RBF</td>
<td>Radial basis function</td>
</tr>
<tr>
<td>RF</td>
<td>Random forest</td>
</tr>
<tr>
<td>RMS</td>
<td>Root mean square</td>
</tr>
<tr>
<td>SC</td>
<td>Skin conductance</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>SDK</td>
<td>Software development kit</td>
</tr>
<tr>
<td>SECRET</td>
<td>Semantically Enhanced Classification of REal-world Tasks</td>
</tr>
<tr>
<td>SHA</td>
<td>Secure Hash Algorithm</td>
</tr>
<tr>
<td>SoDA</td>
<td>Stress detection and alleviation system</td>
</tr>
<tr>
<td>SRAM</td>
<td>Static random access memory</td>
</tr>
<tr>
<td>SVM</td>
<td>Support vector machine</td>
</tr>
<tr>
<td>TN</td>
<td>True negative</td>
</tr>
<tr>
<td>TNR</td>
<td>True negative rate</td>
</tr>
<tr>
<td>TP</td>
<td>True positive</td>
</tr>
<tr>
<td>TPR</td>
<td>True positive rate</td>
</tr>
<tr>
<td>TPR</td>
<td>True positive rate</td>
</tr>
<tr>
<td>VLF</td>
<td>Very low frequency</td>
</tr>
<tr>
<td>WMS</td>
<td>Wearable medical sensor</td>
</tr>
<tr>
<td>YSUUY</td>
<td>Your smartphone understands you</td>
</tr>
<tr>
<td>ZB</td>
<td>Zettabytes</td>
</tr>
</tbody>
</table>
To my beloved parents, Serap and Sinan Akmandor.
Contents

Abstract ................................................................. iii
Acknowledgments ......................................................... v
Acronyms ................................................................. viii
List of Tables ............................................................. xvii
List of Figures ............................................................ xix

1 Introduction .............................................................. 1
  1.1 IoT Sensors and Their Applications ............................ 1
  1.2 Data Processing and Machine Learning Stages ............... 2
    1.2.1 Data Compression ....................................... 2
    1.2.2 Encryption and Hashing ................................ 4
    1.2.3 Data Transmission ...................................... 5
    1.2.4 Bayesian Optimization for Hyperparameter Tuning .... 5
    1.2.5 Feature Extraction and Inference ....................... 6
  1.3 Semantic Vector Models of Words .............................. 6
  1.4 Edge-side Reference Model .................................... 7
    1.4.1 Design Challenges ...................................... 10
  1.5 Smart Healthcare ............................................... 17
    1.5.1 Health Monitoring ..................................... 19
    1.5.2 Medical Automation .................................... 23
    1.5.3 Stress and Health ...................................... 24
1.5.4 WMSs and Physiological Parameters ........................................ 25
1.6 Thesis contributions ............................................................... 28
1.7 Thesis outline .......................................................................... 30

2 Related Work ................................................................................. 32
2.1 Supervised Classification from Feature and Semantic Space Perspectives 32
2.2 Stress Detection and Alleviation .................................................. 35
2.3 Physical-, Mental-, and Emotional-State Detection ......................... 40
2.4 IoT Systems from Smartness, Security and Energy-Efficiency Perspectives .................................................................................. 44

3 SECRET: Semantically Enhanced Classification of Real-world Tasks ........ 49
3.1 Introduction .................................................................................. 49
3.2 Methodology ............................................................................... 51
   3.2.1 The SECRET Architecture .................................................... 51
   3.2.2 Data Processing .................................................................. 52
   3.2.3 Hyperparameter Tuning ....................................................... 54
   3.2.4 Training of ML Models and Inference ................................... 56
   3.2.5 Confidence Score Computation and Decision ......................... 56
3.3 Experimental Results and Discussion .......................................... 60
   3.3.1 Datasets ............................................................................. 60
   3.3.2 Supervised Classifier vs. SECRET ......................................... 62
   3.3.3 Ensemble Method vs. SECRET ............................................. 66
   3.3.4 RF Decision Node Depth ..................................................... 72
3.4 Discussion ..................................................................................... 76
   3.4.1 What to do for classes with the same label, but different meanings? ...................................................................................... 76
3.4.2 How to decide on the semantic vector of a class whose label has synonyms?

3.4.3 Do the improvements come from the selected word embedding or SECRET? How does SECRET perform when a different word embedding is introduced?

3.4.4 Does the semantic space regressor by itself perform better than SECRET?

3.4.5 Why did SECRET observe high performance improvement on the chess dataset?

3.5 Chapter Summary

4 Keep the Stress Away with SoDA: Stress Detection and Alleviation

4.1 Introduction

4.2 Methodology

4.2.1 Data Collection

4.2.2 Experimental Procedure

4.2.3 Stress Mitigation Techniques

4.2.4 Preprocessing and Feature Extraction

4.2.5 Feature Selection, Thresholding, PCA, and Classification

4.2.6 Stress Alleviation

4.3 Experimental Results and Discussion

4.3.1 Feature Selection, Thresholding, and PCA

4.3.2 Stress Alleviation Protocol and Order of Stress Reduction Techniques

4.4 Chapter Summary
5 YSU: Your Smartphone Understands You – Using Machine Learning to Address Fundamental Human Needs

5.1 Introduction

5.2 Methodology

5.2.1 Sensors and Data Collection

5.2.2 YSU Smartphone Application

5.2.3 Experimental Procedure

5.2.4 Data Processing and Feature Extraction

5.3 Experimental Results

5.3.1 Data Collection

5.3.2 Data Processing and Feature Extraction

5.3.3 ML Hyperparameter Tuning and Decision Making

5.4 Application of YSU to Human Needs

5.4.1 Basic Human Needs

5.4.2 Being

5.4.3 Having

5.4.4 Doing

5.4.5 Interacting

5.5 Chapter Summary

6 Smart, Secure, yet Energy-efficient, Internet-of-Things Sensors

6.1 Introduction

6.2 Design and Analysis Methodology

6.2.1 System Design

6.2.2 Energy Modeling

6.3 Experimental Results

6.3.1 Datasets

6.3.2 Alert Notification
# List of Tables

1.1 IoT Sensors and Corresponding Application Areas ........................................... 3

2.1 Stress-related Studies and Corresponding Information on Setting of the Experiment, and Accuracy .......................................................... 36

2.2 Physical/Mental/Emotional State-related Studies and Corresponding Information on Setting of the Experiment, and Accuracy ...................... 41

3.1 Datasets and their Characteristics .................................................................... 61

3.2 Average RF Decision Node Depth, Overall RF Node Depth Variance and Classification Performance on the cmc Dataset ............................. 73

4.1 WMSs, their Abbreviations, Units, and Total Number of Features Extracted ................................................................. 94

4.2 Selected Feature Set for Input to the PCA Stage ................................................. 101

4.3 PCA-reduced Dimensions for the Generalized Model and Statistics of Reduced Dimensions for the 32 Individualized Models ...................... 102

4.4 CPU Time for Feature Extraction ................................................................. 105

4.5 Statistics of Physiological Signals in the Generalized Model for 0-50 sec. ...... 106

4.6 Statistics of Physiological Signals in the Generalized Model for 60-120 sec. .... 106

4.7 Statistics of Physiological Signals in the Individualized Model for 0-50 sec. ... 108
4.8 Statistics of Physiological Signals in the Individualized Model for 60-120 sec .................................................. 109

4.9 Order of Stress Reduction Techniques ............................................................. 109

5.1 Sensors, Their Abbreviations, Types, and Sampling Rates ........................................ 113

5.2 IAPS Database Picture Numbers for Female Participants ........................................... 118

5.3 IAPS Database Picture Numbers for Male Participants ............................................. 119

5.4 Declared Physical and Mental States for Each Participant ......................................... 127

5.5 Total Time Span and Number of Samples for Each Participant in the Physical State Experiments ................................................................. 131

5.6 Total Time Span and Number of Samples for Each Participant in the Mental State Experiments ................................................................. 132

5.7 ML Algorithms and Hyperparameters ................................................................. 133

5.8 Classification Performance of the Physical State Classifiers ..................................... 133

5.9 Classification Performance of the Mental State Classifiers ..................................... 134

5.10 Classification Performance of the Emotional State Classifiers (Four Classes: High Arousal-High Valence, Low Arousal-High Valence, Low Arousal-Low Valence, and High Arousal-Low Valence) ................................................................. 136

5.11 Classification Performance of the Emotional State Classifiers (Two Classes: High Valence, Low Valence) ................................................................. 137

6.1 Supraventricular (S) Ectopic Beat Detection Performance .................................. 159

6.2 Ventricular (V) Ectopic Beat Detection Performance ............................................ 159

6.3 Parkinson’s Disease Freezing of Gait Detection Performance ................................ 163

6.4 Energy Breakdown for the 19-class Human Activity Classification without Compression ................................................................. 169

6.5 Energy Breakdown for the Six-class Chemical Gas Classification in the No-compression Case ................................................................. 174
List of Figures

1.1 Edge-side reference model ........................................ 9
1.2 Expected trends of the systems and their design challenges .......... 12
1.3 Sequence of operations in smart healthcare systems ............... 18
1.4 Positioning of wearable, implanted, and embedded devices .......... 19
1.5 Application classification of smart healthcare systems ............. 20
1.6 Normalized (a) ECG, (b) GSR, and (c) respiration signals .......... 26
2.1 IoT sensor data: (a) alert and (b) continuous notification scenarios . 47
3.1 Architectures: (a) traditional supervised learning and (b) SECRET . 53
3.2 Confidence score computation in the semantic space. Confidence score is calculated for each $k$, where $k \in [1, C]$ and $C$ represents the total number of classes . 58
3.3 Overall confidence score computation and decision-making stages of SECRET .................................................. 59
3.4 Legend for experiments that compare traditional approaches with SECRET .................................................. 63
3.5 SECRET’s test set (a) accuracy and (b) F1 score improvements over the traditional MLP (feature-space) classifier. SECRET uses MLP as the feature space classifier and RF/MLP as the semantic space regressor. Black arrows indicate when the RF regressor is used, whereas the dark blue and dashed arrows correspond to the MLP regressor. . . . . . 64

3.6 SECRET’s test set (a) accuracy and (b) F1 score improvements over the traditional RF (feature-space) classifier. SECRET uses RF as the feature space classifier and MLP/RF as the semantic space regressor. Black arrows indicate when the MLP regressor is used, whereas the dark blue and dashed arrows correspond to the RF regressor. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66

3.7 Architecture of the traditional ensemble method. . . . . . . . . . . . . 67

3.8 SECRET’s test set (a) accuracy and (b) F1 score improvements over the traditional MLP-MLP ensemble in the feature space. SECRET uses MLP as the feature space classifier and RF/MLP as the semantic space regressor. Black arrows indicate when the RF regressor is used, whereas the dark blue and dashed arrows correspond to the MLP regressor. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 68

3.9 SECRET’s test set (a) accuracy and (b) F1 score improvements over the traditional MLP-RF ensemble in the feature space. SECRET uses MLP as the feature space classifier and RF/MLP as the semantic space regressor. Black arrows indicate when the RF regressor is used, whereas the dark blue and dashed arrows correspond to the MLP regressor. . 69
3.10 SECRET’s test set (a) accuracy and (b) F1 score improvements over the traditional RF-MLP ensemble in the feature space. SECRET uses MLP as the feature space classifier and MLP/RF as the semantic space regressor. Black arrows indicate when the MLP regressor is used, whereas the dark blue and dashed arrows correspond to the RF regressor.

3.11 SECRET’s test set (a) accuracy and (b) F1 score improvements over the traditional RF-RF ensemble in the feature space. SECRET uses MLP as the feature space classifier and MLP/RF as the semantic space regressor. Black arrows indicate when the MLP regressor is used, whereas the dark blue and dashed arrows correspond to the RF regressor.

3.12 Pairwise squared Euclidean distances in between the vector representations of labels in the semantic space. A smaller distance indicates a larger class affinity.

3.13 Variance of RF decision node depth for traditional RF classifier and traditional ensemble and SECRET built on top of (a) MLP and (b) RF.

3.14 Test set classification performance of SECRET on the cardio dataset when built with (a) MLP and (b) RF as the feature space classifier and RF as the semantic space regressor. Case 1 represents the performance of the cardio dataset when the labels are used as they are. Case 2 and Case 3 represent the performance when the ‘REM sleep’ label is replaced with ‘paradoxical sleep’ and ‘dreaming sleep,’ respectively.

3.15 Comparison of SECRET’s test set (a) accuracy and (b) F1 score values when built with pretrained Wikipedia 2014 + Gigaword (Wiki) and PubMed word vectors. SECRET uses MLP as the feature space classifier and RF as the semantic space regressor.
3.16 Comparison of SECRET’s test set (a) accuracy and (b) F1 score values when built with pretrained Wikipedia 2014 + Gigaword (Wiki) and PubMed word vectors. SECRET uses RF as the feature space classifier and RF as the semantic space regressor.

3.17 SECRET’s test set (a) accuracy and (b) F1 score improvements over the MLP regressor in the semantic space. SECRET uses MLP as the semantic space regressor and RF/MLP as the feature space classifier. Black arrows indicate when the RF classifier is used, whereas the dark blue and dashed arrows correspond to the MLP classifier.

3.18 SECRET’s test set (a) accuracy and (b) F1 score improvements over the RF regressor in the semantic space. SECRET uses RF as the semantic space regressor and MLP/RF as the feature space classifier. Black arrows indicate when the MLP classifier is used, whereas the dark blue and dashed arrows correspond to the RF classifier.

3.19 Heatmap based on squared Euclidean distances between label vectors of the chess dataset.

3.20 Heatmap based on squared Euclidean distances between feature vectors for each of the classes of the chess dataset.

4.1 SoDA: Stress detection and alleviation system.

4.2 On-body positions of WMSs.

4.3 Experimental procedure.

4.4 Processing and classification of physiological signals.

4.5 Block diagram of SoDA.

4.6 Recurrence count of features extracted from five physiological signals. Analyses are based on the validation set.

4.7 Validation set accuracy with respect to (a) ECG and (b) GSR+RESP+BO+BP recurrence count limits (thresholds) for different classifiers.
6.1 IoT sensor architectures: (a) traditional sense-and-transmit and (b) smart, secure, and energy-efficient. .................................................. 147
6.2 IoT sensor architecture for the traditional sense-and-transmit approach. 148
6.3 IoT sensor architecture for the sense-compress-transmit approach. . . 148
6.4 IoT sensor architecture that employs direct computations on compressively-sensed data: (a) without classification (b) with classification for alert notification, and (c) with classification for continuous notification. .......................................................... 149
6.5 IoT sensor architecture based on compressed signal processing with (a) direct transmission, (b) machine learning inference for alert notification, (c) machine learning inference for continuous notification, (d) compression, (e) compression and machine learning inference for alert notification, and (f) compression and machine learning inference for continuous notification. .......................................................... 150
6.6 IoT sensor applications for alert and continuous notification. . . . . . 155
6.7 Total energy consumption of S-beat and V-beat detecting architectural paths (i.e., those shown in Fig. 6.2, 6.3, 6.4b, and 6.5e) with 1×, 5×, 10×, 15×, 20× compression factors. .................................................. 160
6.8 Total energy consumption for Parkinson’s disease freezing of gait detecting architectural paths (i.e., Fig. 6.2, 6.3, 6.4b, and 6.5e) with 1×, 5×, and 10× compression factors. .................................................. 164
6.9 Total energy consumption for EEG seizure detection for various architectural paths (i.e., those shown in Fig. 6.2, 6.3, 6.4b, and 6.5e) with 1×, 12×, and 24× compression factors. .................................................. 166
6.10 Compressed-domain classification accuracy for the 19-class daily activity classification task using random forest. .................................................. 169

xxiv
6.11 Total energy consumption for neural prosthesis for various architectural paths (i.e., those shown in Fig. 6.2, 6.5d, 6.4c, and 6.5f) with 1×, 4×, and 8× compression factors. The path shown in Fig. 6.5d transmits compressed neural spikes after encryption.

6.12 A new architectural path that contains the nonlinear transformation block.
Chapter 1

Introduction

In this chapter, we first introduce IoT sensors and their applications, various data processing and machine learning stages, semantic vector models of words, then continue with the edge-side reference model, its design objectives, and finally smart healthcare as a real-world application. We end this chapter with our thesis contributions.

1.1 IoT Sensors and Their Applications

IoT sensors measure a physical quantity and communicate with other sensors, actuators, and applications by utilizing the Internet. Owing to their Internet connection, possibly through gateway devices, IoT sensors no longer cater to a single functionality, but are integrated into systems with artificial intelligence (AI) capabilities. With this technological transformation, IoT systems become capable of processing data and making a decision, thus imparting smartness to the system. These capabilities have given rise to myriad applications. For example, healthcare applications include monitoring the health indicators of the user to avoid accidents, detect diseases at an early stage, enhance patient care, infuse precise amounts of medication into the body, and support patient treatment [1]. Agricultural applications include the monitoring of animals, assessing their breeding, and analyzing agricultural production [2].
mental monitoring applications track the chemical properties of air, measure humidity/temperature/water levels, and anticipate/analyze natural/human-made hazards [3]. City/district applications facilitate utilization of parking lots, regulate traffic, assess weather conditions, trace environmental pollution, and ensure safety of the city [4]. Vehicle/transportation applications automate payment for parking, toll, etc., anticipate/report traffic accidents, provide information on road topology, and enable the driver to navigate to a specific location [5]. Power grid applications monitor the use of electricity, automate energy processes, analyze reliability, and enhance security/privacy of the overall system [6]. Home/residence applications track physiological signals from the human body and obtain data from embedded sensors in the environment to guide the user towards a healthier, safer, and more comfortable lifestyle [7].

Some of the IoT sensors and their corresponding application areas are listed in Table 1.1. It gives an inkling of the wide applicability of these sensors.

1.2 Data Processing and Machine Learning Stages

In this section, we discuss data compression, encryption and hashing, data transmission, machine learning hyperparameter tuning, feature extraction, and inference.

1.2.1 Data Compression

Compression decreases data size while aiming to preserve the information embedded in the data. It reduces system resources devoted to processing, inference, storage, and transmission [51]. This leads to energy and storage benefits. This is especially beneficial to systems with severe resource constraints. Compressive sensing is one such method. In compressive sensing, the data are randomly projected to the compressed domain and, when needed, the compressed data are reconstructed by exploiting spar-
<table>
<thead>
<tr>
<th>IoT Sensor</th>
<th>Application Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerometer</td>
<td>Healthcare [8], agriculture [10], environmental monitoring [11], vehicle/transportation [12], home/residence [14], [15], [16], [17].</td>
</tr>
<tr>
<td>Humidity/moisture</td>
<td>Agriculture [18], environmental monitoring [19], [20], city/district [20], [21], vehicle/transportation [22], home/residence [23].</td>
</tr>
<tr>
<td>Acoustic/sound</td>
<td>Healthcare [21], city/district [21], [25], home/residence [16], vehicle/transportation [12].</td>
</tr>
<tr>
<td>Image</td>
<td>Healthcare [26], city/district [27], vehicle/transportation [12], [28], home/residence [29].</td>
</tr>
<tr>
<td>Air-water quality/dust</td>
<td>City/district [21], environmental monitoring [19].</td>
</tr>
<tr>
<td>Barometer</td>
<td>Vehicle/transportation [30].</td>
</tr>
<tr>
<td>Liquid level</td>
<td>Environmental monitoring [31], city/district [32].</td>
</tr>
<tr>
<td>BO</td>
<td>Healthcare [33].</td>
</tr>
<tr>
<td>BP</td>
<td>Healthcare [33], [34].</td>
</tr>
<tr>
<td>Luminosity/light</td>
<td>Agriculture [35], city/district [21].</td>
</tr>
<tr>
<td>Chemical/gas</td>
<td>Healthcare [9], agriculture [36], environmental monitoring [37], city/district [20], vehicle/transportation [22], home/residence [17].</td>
</tr>
<tr>
<td>Magnetometer</td>
<td>Healthcare [9], city/district [38], power grid [39], home/residence [15].</td>
</tr>
<tr>
<td>Passive infrared (PIR)</td>
<td>Healthcare, [9], city/district [25], home/residence [23].</td>
</tr>
<tr>
<td>ECG</td>
<td>Healthcare [33], [40], [34].</td>
</tr>
<tr>
<td>pH (potential of Hydrogen)</td>
<td>Environmental monitoring [19].</td>
</tr>
<tr>
<td>EEG</td>
<td>Healthcare [41].</td>
</tr>
<tr>
<td>Pressure</td>
<td>Healthcare [42], vehicle/transportation [5].</td>
</tr>
<tr>
<td>EMG</td>
<td>Healthcare [40].</td>
</tr>
<tr>
<td>Flow meter</td>
<td>Home/residence [43].</td>
</tr>
<tr>
<td>Respiration rate</td>
<td>Healthcare [33].</td>
</tr>
<tr>
<td>Force/torque</td>
<td>Healthcare [44], vehicle/transportation [13].</td>
</tr>
<tr>
<td>Skin temperature</td>
<td>Healthcare [45].</td>
</tr>
<tr>
<td>Temperature</td>
<td>Agriculture [18], environmental monitoring [19], [46], city/district [20], vehicle/transportation [22], power grid [39], home/residence [17].</td>
</tr>
<tr>
<td>GSR</td>
<td>Healthcare [33].</td>
</tr>
<tr>
<td>GYR</td>
<td>Healthcare [9], [44], vehicle/transportation [12], [13], home/residence [15].</td>
</tr>
<tr>
<td>Hall effect</td>
<td>Home/residence [47].</td>
</tr>
<tr>
<td>Ultrasound</td>
<td>Agriculture [48], vehicle/transportation [49], home/residence [50].</td>
</tr>
</tbody>
</table>
sity in a secondary basis (i.e., the basis in which the data are sparse). The original data are retrieved if the random projection matrix and dictionary of the sparse basis are incoherent with each other \[51\], \[52\]. In this technique, although the compression stage has low computational cost (just one matrix multiplication), data reconstruction is very energy-intensive (three to four orders of magnitude more than data compression) and thus not feasible on the sensor. To find a way around this problem, Shoaib et al. \[51\], \[52\] directly perform signal processing and inference on compressively-sensed data, thus eliminate the need for data reconstruction. Since machine learning algorithms often rely on distance metrics for classification, they focus on minimizing the inner product error in the case of the uncompressed and compressed feature vectors.

CSP \[53\] is another data compression approach. It processes the data in the Nyquist domain, but also relies on random projections to reduce system resources. This reduces the inner product error even more, thus improving classification accuracy of inference.

### 1.2.2 Encryption and Hashing

IoT sensors collect sensitive information, thus requiring meticulous conservation of the security principles: confidentiality, integrity, and availability. These principles can be secured through encryption and hashing. Advanced Encryption Standard (AES) \[54\] and Secure Hash Algorithm (SHA) \[55\] are widely deployed encryption and hashing methods, respectively. AES uses a symmetric key for encryption and decryption. It encrypts 128-bit blocks with 128-, 192-, or 256-bit keys. It includes four main operations: SubBytes, ShiftRows, MixColumns, and AddRoundKey \[54\]. These operations are repeated in multiple rounds based on the number of bits in the key. The last round does not involve the MixColumns step and outputs the ciphertext (i.e., encrypted plaintext). The size of the ciphertext is equal to the plaintext, which
is a multiple of 128 bits \[54\]. SHA-3 is the latest hashing technique used for integrity checking \[55\]. It uses the KECCAK algorithm and produces fixed-length outputs (160, 224, 256, 384, or 512 bits). SHA-3 prevents malicious manipulation of data. If the data are tampered with, the hash algorithm gives a different output and reveals the manipulation \[55\].

1.2.3 Data Transmission

Communication and data transmission between IoT sensors and the base station are crucial to the operation of the IoT system. Many data transmission protocols are available. The choice depends on energy/storage resources, latency tolerance, and data size/frequency. Bluetooth Low Energy (BLE) is one of the most widely used data transmission protocols. It provides short-range communication in the 2.4 GHz Industrial Scientific Medical (ISM) band \[56\]. It uses master and slave devices. Each master has multiple slaves. The master is responsible for determining the listening schedule for and providing connection/frequency information to the slave. Except for waking up at specific time intervals to listen to the packet, slaves stay in the sleep mode. This saves system energy \[56\]. The Medical Implant Communication Service (MICS) band is a widely used communication band that supports communication between low-power implanted medical devices and external monitoring or control equipment \[57\]. The 402 to 405 MHz MICS band offers reasonable propagation characteristics for signals within and around human bodies. The introduction of MICS has led to the advent of new medical applications, where various wireless nodes in, on, or around a human body can collaborate to monitor vital signs.

1.2.4 Bayesian Optimization for Hyperparameter Tuning

The selected set of hyperparameter values has a direct impact on classification/regression performance. Hand-tuning, random search, grid search, and Bayesian
optimization are commonly used methods for finding the best set of hyperparameter values. We adopt Bayesian optimization as it is known, in general, to provide an unbiased analysis and higher classification/regression performance, while requiring a small number of iterations due to the utilization of results from past iterations [58].

Bayesian optimization integrates exploration and exploitation. It starts with a prior belief over the unknown objective function. It then evaluates the optimization goal function with available data (target hyperparameter values chosen for the iteration). Based on input data and the corresponding optimization goal outputs, it updates the beliefs and selects the next set of hyperparameter values to be evaluated. The process is repeated until a maximum number of iterations is reached [59].

1.2.5 Feature Extraction and Inference

The feature extraction stage extracts informative values (i.e., features) from the data collected through IoT sensors. The feature extraction process may lead to linear or nonlinear features. The inference stage takes these extracted features as input and makes a decision by utilizing previously-trained machine learning models. These models can be derived using any machine learning algorithm. The choice of algorithm depends on the resultant model complexity, available energy/storage resources, and data characteristics.

1.3 Semantic Vector Models of Words

Semantic vector models assign a compact real-valued vector to each word in a dictionary. The vector captures the word’s semantic relationships with the remaining words in the dictionary. Words with close meanings are represented by closely-spaced vectors in the semantic space. Some of the algorithms that derive semantic vector word representations are Skip-gram and Continuous Bag-of-Words (CBOW) archi-
tectures of word2vec \[60\], GloVe \[61\], vLBL \[62\], ivLBL \[62\], Hellinger PCA \[63\], and recurrent neural networks \[64\].

GloVe is an unsupervised method. It uses the co-occurrence ratio of words within a pre-specified window length to obtain the word vectors. Use of this ratio enhances the distinction between two relevant words or a relevant word and an irrelevant one. The GloVe algorithm is based on weighted least squares regression. As shown in Eq. (1.1), it aims to minimize the difference between the scalar product of the two word vectors and the logarithm of their co-occurrence value. Weights are used to avoid dominance (overweighting) by both very frequent and rare co-occurrences. The corresponding weighting function is shown in Eq. (1.2). In \[61\], \(\alpha = \frac{3}{4}\) has been found to yield good results.

\[
J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij})(w_i^T w_j + b_i + b_j - \log(X_{ij}))^2,
\]

where \(V\): vocabulary size,

\(f(x)\): weighting function,

\(X\): co-occurrence matrix,

\(w\): word vector,

\(b\): bias.

\[
f(x) = \begin{cases} 
(x/x_{max})^\alpha, & \text{if } x < x_{max} \\
1, & \text{otherwise}
\end{cases}
\] (1.2)

### 1.4 Edge-side Reference Model

Technological advancements have led to a rapid increase in computing power of electronic devices. This improvement has enhanced the functionality of the systems and shifted various operations (e.g., data acquisition, communication, processing, and
analysis) towards the edge-side. Carrying out these operations on the edge-side leads to benefits in terms of energy consumption, latency, security, cost, etc. Figure 1.1 shows the reference edge-side model with three main levels: sensor node, communication, and base station [65]. An application might employ all three levels, whereas another application might only use the first level, thus carrying out the analyses only on the sensor node.

We explain the function of each level next.

**Sensor node (Level 1):** The data are acquired through sensors (wearable, implanted, and embedded), the first level of the reference model shown in Fig. 1.1. Following data acquisition, depending on the healthcare application and available resources, either data processing and inference are performed (Level 1) or the data are sent to the base station (Level 3) through a communication network (Level 2). Latency and energy consumption are minimized if data are processed and inference is performed at the sensor node. This may be a significant consideration in emergency healthcare applications that require real-time alerts or responses. Processing of the data includes re-formatting, compression/expansion, de-noising, and feature extraction. The inference stage includes previously trained machine learning models. The type and ordering of processing stages and inference depend on the application. If the data are sent to the base station due to limited resources of the sensor node, latency and energy consumption increase. However, the sensor node complexity decreases. In this case, there is no need for the processing and inference stages at the sensor node; the system directly transmits data to the communication stage (Level 2).

**Communication (Level 2):** In case of limited energy and storage at the sensor node, the data are transferred to the second level. This level of the reference model includes a data communication network [65]. It enables communication among sensor nodes in the first level, between different communication networks in the second level,
Figure 1.1: Edge-side reference model.
and base station in the third level [65]. ZigBee, Bluetooth, IEEE 802.11n/WiFi, and IEEE 802.11g/WiFi are examples of possible communication networks [66], [67].

**Base station (Level 3):** Base stations comprise the third level of the reference model shown in Fig. [1.1]. If the data are not processed in the first level due to resource constraints, they are processed at the base station. This level provides significant energy and storage resources that compensate for the limited capacity of the sensor nodes and carry out the processing (i.e., re-formatting, feature extraction, etc.) and inference operations. Completing processing and analyses of the data at this level has significant benefits. Since the need for further transmission to and processing of data in the cloud is eliminated, energy, time, and cost are saved. Moreover, possible security attacks that target user’s sensitive data are minimized. Therefore, use of this level improves system energy consumption, latency, security, and cost. However, there are many other design challenges that affect the operation and applicability of the corresponding system. Many research studies have been carried out at the edge-side to improve performance and overcome these design challenges, as described next.

### 1.4.1 Design Challenges

The systems execute the tasks assigned to them. However, performing the task is only one aspect. There are other factors that affect design quality, such as accuracy of the algorithms employed in processing and inference stages, security of the overall system, cost of implementation, etc. We call these factors design challenges. Some of the design challenges are as follows:

- Accuracy
- Security
- Silent operability
- Latency
- Maintainability
- Scalability
- Energy consumption
- Cost
- Customizability
Along with the system components, design challenges and their expected trends are shown in Figure 1.2. Due to advancements in the computational power of electronic devices, system components shift towards the edge-side, i.e., no longer need to depend on cloud-side services. If processing and analyses are completed in the edge-side levels, time, energy, and cost of data transmission to the cloud servers is saved. However, while improving these metrics, systems of the application-of-interest also try to address the remaining design challenges: accuracy, security, maintainability, silent operability, scalability, and customizability. The system that includes only edge-side components has fewer levels than the system that uses cloud services. Therefore, edge-side systems are less error-prone. Moreover, due to fewer levels, edge-side computing improves resiliency of the system to security attacks, enables customization of the system to user’s needs, and improves maintainability. Also, since the edge-side depends on local computing, it also enables silent operability without involving the user too much and scalability to more sensors or disease diagnoses, etc.

Each of the design challenges and methods to address them are explained in detail next.

**Accuracy**

Smart systems make decisions, raise alarms, and guide the users according to the machine learning based inferences carried out on the processed data. Therefore, accuracy of the processing algorithms and inferences is important. For smart healthcare as the application, if the health condition of the user is determined with high accuracy, the system becomes more reliable and useful for both health monitoring and medical automation purposes. Use of more data points and features, de-noising, normalization, feature selection, dimensionality reduction, reducing imbalance between the classes, kernel methods, and ensemble methods have been shown to improve the accuracy of the decision-making systems.
Figure 1.2: Expected trends of the systems and their design challenges.
Latency

A timely alert plays an important role in many systems. For example, in smart healthcare, a prompt notification of a heart attack or seizure might save the life of the user. This means low system latency is important.

Quality of the network connection and processing of the data have an obvious impact on the latency. If the data are transmitted to cloud servers through weak network connection with unforeseen failures, latency is affected negatively. Therefore, carrying out the operations at the edge-side of the system has significant advantages [68]. Moreover, in order to minimize the latency, data need to be processed and re-formatted at the edge-side. Compression, data abstraction, and inference are some of the operations that can reduce overall system latency.

Energy Consumption

Data collection, analysis, and transmission are all energy-consuming operations. When the energy consumption is high, the systems require frequent battery change or recharging. This has negative effects on both the practicality of the system and health condition of the user. For example, if the battery energy of a pacemaker runs out, it requires surgery for battery replacement, with all its attendant risks of infections and post-operative complications. Therefore, the systems should ideally have low energy consumption.

Data abstraction is one way to reduce energy consumption. Ganz et al. [69] have described the common processes involved in data abstraction that convert raw data into a semantic representation. The common processes are: preprocessing, dimensionality reduction, feature extraction, and inference. How each process is employed depends on the healthcare application. Another method for reducing energy consumption is data compression. However, if data signals need to be reconstructed in order to perform signal processing on them, the reconstruction process often incurs
significant energy costs (several orders of magnitude higher than the compression process itself) \[51, 52\]. In order to address this problem and enable edge-side computing, Shoaib et al. \[52\] have introduced methods for signal processing on compressed data. By using compressed representations of the data throughout the system, they achieve comparable accuracy to the case when signal processing is done on uncompressed data, however, with \(54\times\) fewer samples for the neural prosthesis application and \(21\times\) fewer samples for the epileptic seizure detection application. In another study, Lu et al. \[53\] have introduced the concept of compressed signal processing. This is different from the work of Shoaib et al. It takes advantage of random projections of the uncompressed data collected in the Nyquist domain. This dramatically reduces error in the inner product computation, which forms the backbone of many inference methods, yet providing comparable accuracy with \(32\times\) fewer samples for both the neural prosthesis and epileptic seizure detection applications.

Security

Most of the systems collect data from the user, carry out analysis or inference on the collected data, and provide feedback, accordingly. From data collection to feedback, each level of the smart healthcare system carries sensitive information belonging to the user. Thus, unauthorized access, data manipulation, and data availability reduction might have serious negative impact on the user. Some of these threats arise from side-channel analysis, denial-of-service (DoS) attacks, node replication, eavesdropping, malicious injection, and machine learning attacks \[70\]. In order to circumvent the possible threats, the systems need to ensure that the data acquisition, processing, and transmission levels are secure.

Encryption and hashing are two of the important steps one can take in enhancing data security. However, introducing encryption and hashing increases the overall energy consumption significantly, which is not practical in most cases. Zhang et al. \[71\]
address these concerns through a concept called encompression that combines encryption/hashing with data compression. Their experiments with application of encompression to data acquisition and transmission show that the energy consumption can be reduced by 78% relative to a sensor that incorporates encryption/hashing. In fact, it reduces energy by 14% when employing $10\times$ compression even when compared to a traditional baseline sensor that does not employ compression or encryption/hashing. This represents an energy bonus.

**Maintainability**

Over time, the systems need software and hardware maintenance. This involves operations such as data login, network setting changes, sensor substitutions, and machine learning inference updates. The ability to enable these operations is called maintainability. This is an important property for many systems including smart healthcare so that they can provide long-term utility and user-centered treatment.

Maintainable systems provide flexibility by enabling changes to the treatment processes, updates to system settings, inclusion/exclusion/substitution of devices, and fixes for system errors. System maintainability needs to be taken into account and analyzed carefully during the design stage itself. Although this may incur extra design cost, maintainable systems prove their value in the long run. In the case of system failure or a change, the system does not necessarily have to be discarded, but just updated.

**Cost**

Affordability has a direct impact on how quickly a system gets adopted in the marketplace. Affordable costs encourage more people to start using these systems, for example for smart healthcare either to monitor their health or to automate their treatment and drug delivery operations. Moreover, as costs decrease, hospitals and
insurance companies become more interested in supporting the new technology. This improves service quality. Thus, cost also needs to be taken into account and optimized in the design phase to enable incorporation of more high-quality features, but at low cost. Employing edge-side computing is one way to reduce cost. Since data analyses are carried out at the edge-side, the cost of data transmission to and analysis in the cloud is saved.

**Silent Operability**

If the smart healthcare system requires frequent user interaction, it becomes less appealing, and may lead to either incorrect decisions or discontinued operation. To be successful in its objective of providing long-term support, the systems should operate in a silent manner, i.e., without seeking frequent inputs from the user.

Use of edge-side inference makes it possible to make important decisions without requiring too many user interactions. This involves collecting data from sensors, processing the data to derive useful features from them, and feeding the feature values to the inference stage. This automation has the potential to significantly expand the scope of the application because of its ability to continuously monitor the user in the background.

**Scalability**

The systems will succeed when they become ubiquitous. This would depend on how adaptable they are to the number of sensors as well as on their fault tolerance, availability, and security. Thus, robustness of such systems to inclusion/exclusion of sensors and amount of data need to be ascertained in the design stage [72].

Edge Mesh is one of the proposed scalable edge-side computing models [68]. Based on available resources and task objectives, Edge Mesh divides the computational task into smaller pieces and assigns them to various edge-side devices. It promotes scalabil-
ity by providing a hierarchical computational environment and enabling cooperation among multiple devices [68].

**Customizability**

Most of the applications are user-dependent. For example, in smart healthcare, the health indications, tolerance to treatments, and physiological responses to rehabilitations are user-dependent. Thus, to provide accurate user-specific health monitoring, the healthcare system needs to be cognizant of these variations. Customizable smart healthcare systems should make it easy to personalize their parameters to a particular user and health condition. This will lead to more accurate responses.

A personalized system can be made possible through the development of user-specific machine learning models that can be incorporated into the inference stage by utilizing the corresponding user’s data. However, customizability is not limited to inference. For example, adjusting the types and parameters of wearable, implanted, and embedded sensors to a specific condition is also an avenue to customization. Customizability of the system also needs to be taken into account during the design stage.

**1.5 Smart Healthcare**

Smart healthcare improves quality of healthcare by closely monitoring the health indicators of the user. Figure 1.3 shows the main operations involved in smart healthcare and their sequence. First, the system acquires data from the sensors. Then, it processes the data to remove unwanted components (i.e., de-noise the data), normalizes the data to within a specific range, and selects informative features. Following the processing stage, it analyzes the features to assess whether a health alert should be
triggered. In the case of an alert, the system either directly performs the corresponding task or else informs and guides the user, accordingly.

Depending on the application, smart healthcare systems employ different sets of devices. Many of these devices are wearable in nature. Wearable devices are either embedded into clothing or worn on the body. They are capable of measuring physiological signals through, for example, electrocardiogram (ECG), electroencephalogram (EEG), skin conductance, respiration rate monitor, blood oximeter, activity tracker, etc. However, smart healthcare is not limited to wearables. It also includes devices that are implanted in the body or embedded in the furniture or appliances. Some examples of implanted devices are heart monitor, pacemaker, infusion pump, and hearing aid. Similarly, pressure-sensitive grab bars [73], electronic noses [73], ECG measuring toilet seats [74], pressure-sensitive mats [73], and nutritional information tracking fridges [75] are examples of embedded devices. Figure 1.4 shows how smart healthcare devices may be positioned. However, these positions are not fixed. For instance, the blood oximeter may also take oxygen saturation (SpO2) measurements.

![Figure 1.3: Sequence of operations in smart healthcare systems.](image-url)
from the ear lobe, as opposed to the fingertip. Healthcare applications, physicians, and user preferences determine how smart healthcare devices are positioned.

Based on their purpose and function, smart healthcare systems are divided into two main categories: health monitoring and medical automation [76]. Health monitoring systems track the health condition of the user to provide a ‘no emergency’ confirmation, proactive prevention strategies against a medical condition, and early diagnosis of a disease. Medical automation systems provide treatment and rehabilitative care [76]. Figure 1.5 shows the various categories of smart healthcare systems. These are described next.

### 1.5.1 Health Monitoring

Smart health monitoring systems use wearable, ingestible, and embedded sensors to track the health condition of the user. These sensors enhance doctor-patient communication, speed up disease diagnosis, and decrease healthcare costs [33]. Such systems can be divided into three main groups: protective, preventive, and responsive. Protec-
Figure 1.5: Application classification of smart healthcare systems.

tive systems check whether there is an emergency situation that requires immediate action. Preventive systems implement proactive strategies to prevent possible diseases from arising in the future. Responsive systems detect medical conditions at an early stage and inform the user or the physician [76].

Protective systems

Protective health monitoring systems track the health indicators of the user and check for the occurrence of an emergency health situation. In case of an emergency, an alert is produced and the emergency contact informed. Such systems generally focus on children, elderly people, chronically ill people, and people with disabilities [66]. Based on their function, protective systems can be divided into two categories: pre-incident and post-operative.

Pre-incident: Pre-incident health monitoring systems use health indicators to assess the critical circumstances that might lead to accidents. These systems promote wellness by forestalling possible accidents. Fall detection and location tracking are the most common examples of pre-incident health monitoring.

Fall detection systems generally use accelerometers, gyroscopes, or cameras to distinguish the fall event from other physical activities. Since camera use leads to privacy concerns, Li et al. [77] propose a novel fall detection system based on accelerometers
and gyroscopes. By analyzing the collected signals, their system determines whether the transitions from one body posture to another are intentional or not. If the transition is unintentional, the corresponding case is classified as a fall. Their system achieves 91% sensitivity and 92% specificity.

Location tracking systems monitor users and detect accidents with the help of wearable and embedded devices. Ultra Batch is a continuous location monitoring system that uses ultrasonic sensors [78]. It tracks the location of wheelchairs and activities of elderly people. It detects the position of the user’s head within a 5 cm error range. Upon detecting a critical event, it sends an alarm to a caregiver.

**Post-operative:** Post-operative health monitoring systems track the health indicators of the user to detect infections or complications after surgery. Thus, they enable early intervention. Aziz et al. [79] propose a wearable system that uses a pulse oximeter and accelerometers to assess the recovery of the patient after abdominal surgery. It calculates the recovery rate by analyzing respiratory function and mobility. Various experiments verify its efficacy in post-operative monitoring.

**Preventive systems**

Preventive health monitoring systems detect unhealthy user behavior and offer proactive strategies to deal with the behavior. Such systems promote wellness by taking action before a medical condition emerges [76]. Posture correctors, fitness trackers, and stress management systems are the most common examples of preventive health monitoring systems.

Posture correctors help the user maintain proper posture. When an unhealthy posture is detected, they alert the user and provide feedback for straightening the back. Harms et al. [80] propose a wearable posture corrector that uses acceleration sensors to detect bending of the back. They experiment with six bending positions.
of the back that are separately evaluated by therapists. The measurements obtained by their system are shown to have a significant correlation with expert ratings.

Fitness trackers are among the most widely used health monitoring systems. They measure physiological signals, such as the heart rate, and assess the number of steps taken, calories burnt, fitness progress, etc. They report the measurements to the user and guide them towards a healthier lifestyle [81].

Stress management systems utilize wearable and embedded sensors to keep the stress level of the user within normal levels. SoDA is a recently proposed continuous and automatic stress management system that uses wearable sensors to detect and mitigate stress [33]. When stress is detected, SoDA offers a sequence of stress reduction techniques and analyzes the physiological signals when the user adopts one of these techniques. If the physiological signals show a tendency towards the relaxed state, the technique is pursued until a ‘no stress’ state is achieved. Otherwise, SoDA offers the next technique and repeats the procedure. It achieves 95.8% classification accuracy for the stress detection stage and an efficient response for the stress alleviation stage.

Responsive systems

Responsive health monitoring systems provide early disease diagnosis by tracking the health indicators of the user continuously [76]. This enables the system to identify the disease even from minor symptoms. Based on their objective, such systems can be divided into two categories: emergency care and regular care.

Emergency care: Emergency care responsive systems use health indicators of the user to detect an emergency medical condition and activate emergency care. CodeBlue is an example of such a system [82]. It monitors the health indicators, assesses the medical condition of the patient, and organizes the triage and resources of the hospital accordingly.
**Regular care:** Regular care responsive systems also track the health status of the user continuously. When a health condition is detected, the system provides a report to the user or informs the corresponding physician. Nia et al. propose three schemes for continuous and energy-efficient personal health monitoring: sample aggregation, anomaly-driven transmission, and compressive sensing [83]. They analytically model the energy and storage requirements of the system under these schemes. This system leads to two to three order magnitude reduction in energy and storage requirements. In another study, Yin et al. propose a hierarchical health decision support system that uses wearable sensors to diagnose various diseases with machine learning ensembles [34]. Their system has a multi-tier structure, with the first tier being the wearable sensor tier. The efficacy of the approach is established through diagnosis of five diseases: arrhythmia, type-2 diabetes, urinary bladder disorder, renal pelvis nephritis, and hypothyroid. They are diagnosed with classification accuracies between 78% and 99% [34]. MobiHealth is another approach for regular care [84]. It provides an end-to-end mobile platform over UMTS and GPRS networks. It collects data from medical sensors and transmits them to healthcare providers. Different health conditions have been evaluated using MobiHealth with user groups in various European countries. It was shown to successfully provide remote health monitoring, without negatively impacting the mobility of the user.

### 1.5.2 Medical Automation

Medical automation systems use wearable, ingestible, and embedded sensors to support medical treatment of the user. They promote wellness by continuously tracking the health indicators of the user and providing on-time medication or rehabilitation accompanied with the correct dose. Based on their objective, such systems can be divided into two categories: drug infusion and rehabilitation.
Drug infusion systems

Drug infusion systems inject precise amounts of medication into the body through electronically controlled and implanted infusion pumps [85]. Such systems are used for insulin delivery to diabetes patients, morphine transmission for chronic pain management, and antineoplastic therapy transmission for liver cancer patients [85].

Rehabilitation systems

Rehabilitation systems are used to improve the physical abilities of the user after a serious incident, such as a stroke, traumatic injury, arthritic condition, etc. These systems assist the user, provide objective feedback, and reduce hospital dependency [86]. A wearable rehabilitation robotic hand is an example of a rehabilitation system. Xing et al. show how a robotic hand can assist with repetitive hand motions, provide feedback, and encourage user’s efforts [86]. Their design is implemented with a PID controller and pneumatic artificial muscles. The effectiveness of this rehabilitation system has been validated through detailed experiments.

1.5.3 Stress and Health

Stress is a wide-ranging and complex topic that does not have a specific definition. According to Lazarus and Folkman, stress is the relationship between the person and a situation, which adversely impacts the happiness and health of the sufferer [87]. Another definition of stress is that it is a physiological reaction that aims to protect the individual from possible threats emanating from the environment [88]. These definitions indicate that stress arises from a threatening situation. Our body activates its defense mechanism to adapt to or overcome the stressful circumstance. When the stressor disappears, our body returns to normal operation. However, this recovery takes some time since stress results in chemical changes in our body. Thus, continuous
exposure to stress prevents the body from returning to normal, and thus has long-term health consequences, ranging from cardiovascular to psychological problems.

### 1.5.4 WMSs and Physiological Parameters

WMSs are noninvasive and autonomous devices that are used to monitor human health. They are called wearable since they are placed on the human body or clothing. They come in various forms: patches, bandages, glasses, rings, bracelets, etc. [89]. Currently available WMSs can monitor posture, fetal health, heart disease, obesity, diabetes, epilepsy, sleep quality, cigarette smoking, etc. WMSs are a rapidly growing market and expected to become ubiquitous [90].

The relationship between stress and physiological parameters has been studied and indicative ones identified. These parameters are heart rate [91], blood pressure [91], skin conductivity [92], respiration rate [93], blood oxygen level [94], electromyograph (EMG) of trapezius muscles [95], pupil diameter [96], and cortisol level [97]. Among these, EMG requires multiple electrodes in the form of patches or needles, pupil diameter measurement requires standing still, and cortisol level requires either blood or saliva sample. Since these three measurements are not easy to obtain in our daily lives, we do not use them in our experiments. Obtaining the remaining parameters are minimally obtrusive: heart rate, blood pressure, skin conductivity, respiration rate, and blood oxygen level. We briefly describe these WMSs next.

**Electrocardiogram (ECG):** The ECG sensor measures the electrical activity of the heart during a cardiac cycle [98]. It is noninvasively obtained by relying on body fluids as conductors and comparing the potential difference between the electrodes [99]. A typical ECG signal is shown in Fig. 1.6a. A cardiac cycle is composed of P-Q-R-S-T waves. The most detectable part of the ECG signal is the Q-R-S complex. In this complex, the first negative deflection is called the Q-wave, which is followed by a large positive deflection called the R-wave, and the next negative deflection called
the S-wave. Generally, the Q-R-S complex lasts for 60-100 milliseconds in adults. From the ECG signal, values for various parameters, such as the heart rate (HR),
heart rate variability (HRV), R-R interval, etc., can be derived. Deviations in the values of these parameters may indicate stress.

**Galvanic skin response (GSR):** GSR indicates the change in electrical characteristics of the skin due to perspiration from the body [100]. It measures skin conductance (SC) noninvasively by applying a low constant voltage through the electrodes. An example normalized GSR signal is shown in Fig. 1.6b. The measured SC is composed of two electrodermal activities: phasic and tonic [101]. Phasic activity is the high frequency component and thus varies quickly, whereas tonic activity is the low frequency component and thus changes more slowly [102]. In a stressful situation, our body produces sweat, which changes skin conductance.

**Respiration monitor:** Respiration is composed of inhalation and exhalation. An example waveform of the respiration signal is shown in Fig. 1.6c. During respiration, oxygen is transmitted to the cells and the accumulated carbon dioxide is removed [102]. The normal respiration rate for adults is 12-16 breaths per minute [102]. Although respiration rate can be obtained through different methods, the respiration monitor we use measures thoracic expansion to obtain respiratory information. Stressors influence the duration and amplitude of inhalation and exhalation.

**Blood oximeter:** A blood oximeter noninvasively measures the oxygen saturation, SpO2, with the help of light-emitting diodes (LEDs). Blood consists of hemoglobin molecules. When these molecules have different oxygen levels, they lead to different levels of absorption of the light emitted through the LEDs [103]. Light intensity after absorption indicates the fraction of blood hemoglobin in the oxygenated state (SpO2). SpO2 can be measured in reflectance or transmittance mode. In the reflectance mode, after emission of the light by LEDs, the blood oximeter analyzes the backscattered light to obtain the SpO2 information. In the transmittance mode, the blood oximeter emits light from one side of a fingertip or earlobe and analyzes
the received signal emanating from the other side to assess SpO2. Stress also has an impact on SpO2.

**Blood pressure monitor:** Blood pressure is the force per unit area exerted on blood vessels of the circulatory system. It has two components: systolic and diastolic. Systolic blood pressure indicates the pressure when the heart pumps blood into the arteries, whereas diastolic blood pressure indicates the pressure when the arteries withstand the blood flow. Both systolic and diastolic blood pressures can be obtained through the blood pressure monitor. The normal range for the systolic blood pressure is 90-120 mmHg and for the diastolic blood pressure 60-80 mmHg. In the presence of a stressor, systolic and diastolic blood pressures deviate from their baseline levels.

### 1.6 Thesis contributions

This section presents the thesis contributions.

In Chapter 3 we propose a dual-space machine learning decision process: SECRET. SECRET performs classifications by fusing the semantic information of the labels with the available data: it combines the feature space of the supervised algorithms with the semantic space of the NLP algorithms and predicts labels based on this joint space. Experimental results on ten different datasets indicate that, compared to traditional supervised learning, SECRET achieves up to 14.0% (avg. 2.0%) accuracy and 13.1% (avg. 3.3%) F1 score improvements. Moreover, compared to ensemble methods, SECRET achieves up to 12.7% (avg. 1.8%) accuracy and 13.3% (avg. 2.8%) F1 score improvements.

In Chapter 4 we present an automatic stress detection and alleviation system, called SoDA. SoDA takes advantage of emerging wearable medical sensors (WMSs), specifically, electrocardiogram (ECG), galvanic skin response (GSR), respiration rate,
blood pressure, and blood oximeter, to continuously monitor human stress levels and mitigate stress as it arises. It performs stress detection and alleviation in a user-transparent manner, i.e., without the need for user intervention. When SoDA detects stress, it employs a stress alleviation technique in an adaptive manner based on the stress response of the user. We establish the effectiveness of the proposed system through a detailed analysis of data collected from 32 participants. A total of four stressors and three stress reduction techniques are employed. In the stress detection stage, SoDA achieves 95.8% accuracy with a distinct combination of supervised feature selection and unsupervised dimensionality reduction. In the stress alleviation stage, we compare SoDA with the no-alleviation baseline and validate its efficacy in responding to and alleviating stress.

In Chapter 5 we present ML models that can be used to understand us. We call this system your smartphone understands you (YSUY). YSUY uses wearable medical sensors to understand our physical, mental, and four-class (two-class) emotional states with 90.0%, 90.3%, and 98.4% (99.5%) accuracy, respectively. After verifying YSUY’s ability to understand the human condition from various perspectives, we evaluate the relationship between the different states and discuss how YSUY can be taken one step further to start playing a helpful role in addressing human needs of the types mentioned above from four different perspectives: ‘being,’ ‘doing,’ ‘having,’ and ‘interacting.’ We show that YSUY is a promising candidate for adapting ML models to human-centric needs.

In Chapter 6 we propose simultaneously smart, secure, and energy-efficient IoT sensor architecture, SSE, by employing signal compression and machine learning inference on the IoT sensor node. An important sensor operation scenario is for the sensor to transmit data to the base station immediately when an event of interest occurs, e.g., arrhythmia is detected by a smart ECG sensor or seizure is detected by a smart EEG sensor, and transmit data on a less urgent basis otherwise. Since
on-sensor compression and inference drastically reduce the amount of data that need to be transmitted, we actually end up with a dramatic energy bonus relative to the traditional sense-and-transmit IoT sensor. We use a part of this energy bonus to carry out encryption and hashing to ensure data confidentiality and integrity. We analyze the effectiveness of this approach on six different IoT applications with two data transmission scenarios: alert notification and continuous notification. The experimental results indicate that relative to the traditional sense-and-transmit sensor, IoT sensor energy is reduced by $57.1 \times$ for ECG sensor based arrhythmia detection, $379.8 \times$ for freezing of gait detection in the context of Parkinson’s disease, $139.7 \times$ for EEG sensor based seizure detection, $216.6 \times$ for human activity classification, $162.8 \times$ for neural prosthesis spike sorting, and $912.6 \times$ for chemical gas classification. Our approach not only enables the IoT system to push signal processing and decision-making to the extreme of the edge-side (i.e., the sensor node), but also solves data security and energy efficiency problems simultaneously.

1.7 Thesis outline

The remainder of the thesis is organized as follows. Chapter 2 presents related work from the literature on supervised classification from feature and semantic space perspectives, stress detection and alleviation, physical, mental, and emotional state detection, and IoT systems from smartness, security, and energy-efficiency aspects. Chapter 3 describes dual-space (feature and semantic) machine learning decision process, SECRET, with the methodologies underpinning the proposed architecture, data processing, hyperparameter tuning, machine learning algorithm training in the feature and semantic spaces, confidence score calculation, and decision process. Chapter 4 presents stress detection and alleviation system, SoDA with methodologies for data collection, experimental procedure, processing of physiological signals, classification,
and stress alleviation. Chapter 5 describes the YSU Y system that is focused on understanding the physical, mental, and emotional states of the users. The chapter introduces the methodologies for data collection, experimental procedure, data processing, feature extraction, machine learning hyperparameter tuning, and decision making with detailed experimental results. Chapter 6 presents simultaneously smart, secure, and energy-efficient IoT sensor architecture with methodologies for system design and energy modeling. Finally, Chapter 7 concludes the thesis with a brief summary and future research directions.
Chapter 2

Related Work

This chapter presents prior work on supervised classification from feature and semantic space perspectives, stress detection and alleviation, physical-, mental-, and emotional-state detection, and IoT systems from smartness, security and energy-efficiency perspectives.

2.1 Supervised Classification from Feature and Semantic Space Perspectives

Enhancing classification performance of the ML algorithms has been a well-targeted area of research for decades. Various approaches have been proposed. These include data augmentation [106], data generation [107], boosting [108], ensemble learning [109], and dimensionality reduction [110]. In addition to these promising techniques, various ML algorithms (information-based, similarity-based, probability-based, and error-based [111]) and architectures have been designed. Specifically, for big data, neural network models [112] have revolutionized the classification task due to their ability to model complex data-label relationships. Although these algorithms and techniques have made significant contributions to enhancing classification perfor-
mance, they all operate in the feature space. Additionally, various feature space approaches utilizing the word embeddings have been proposed. Lilleberg et al. [113] used word embeddings as additional features to term frequency-inverse document frequency (TFIDF) for the text classification task. Although this approach showed the effectiveness of semantic word embeddings in improving classification accuracy, its scope is limited to the feature space since word embeddings were used as features and semantic space mapping was not performed. Another feature space approach by Vo and Zhang [114] used an extensive set of features, including word embeddings, to accurately perform Twitter sentiment classification. Wang et al. [115] proposed a CNN-RNN text-image embedding framework, taking image data and radiological reports as input and performing disease diagnosis. The proposed approach obtained 6% AUC improvement, on an average, relative to the state of the art. However, similar to the studies in [113] and [114], it adds semantic information in the form of additional features and the semantic relationship between the labels are not taken into account while building the classifier. This limited the approach to the feature space. Kusner et al. [116] introduced a novel distance metric (Word Mover’s Distance) to effectively model the text documents with a set of word vectors. Vector representations act as features in a traditional classification task and are mapped to a pre-defined set of labels with the k-nearest neighbor algorithm. This is a feature space approach since word vectors are used as features and mapped to a specific set of labels, without considering the meaning-based relationships among labels.

If we change our perspective and look at the related work in the semantic space, we observe that word representations have been widely used in NLP applications. Liu et al. [117] proposed a novel task-oriented word embedding method to assess the salient word for the text classification task. All analyses are carried out in the semantic space. Bordes et al. [118] targeted question answering by representing the question in a vector form in the semantic space and mapping it to the answer again in the seman-
tic space. Bengio and Heigold [119] go far away from the semantic space by training vector representations of words without considering their meaning relationships, but targeting how similar the words sound. Vectors of sound-alike (not semantically similar) words have a smaller Euclidean distance between them. Wang et al. [120] focused on the design of a CNN-RNN framework that maps image data to label embedding to perform multi-label classification while taking label co-occurrence and semantic redundancy into account. The proposed approach targeted only the image datasets with a fixed ML design. Karpathy and Fei-Fei [121] obtained figure captions using image datasets and word embeddings. Palatucci et al. [122] and Socher et al. [123] carried out zero-shot learning by mapping real-world data to semantic vector representations of words. SoundSemantics [124] extended the zero-shot learning approach proposed by Palatucci et al. [122] to sound datasets. The proposed approach initially classified input sound as heard or unheard. In case of unheard, zero-shot learning was performed using word embeddings. Norouzi et al. [125] proposed a zero-shot learning method that did not require retraining of the machine learning model, but used predictive probabilities for labels to obtain a vector representation of the novel class label. All the zero-shot approaches proposed in [122], [123], [124], and [125] are limited to semantic space information as they focused on label word embeddings or mapping of data to label word embeddings. Murthy et al. [126] proposed a regression model based on CNN (CNN-R) for image annotation. CNN-R maps data to vector representation of the labels. They pointed to the need for regularization for performance improvement. However, the study was limited to the semantic space as feature space information was not utilized. Deng et al. [127] built Hierarchy and Exclusion (HEX) graphs to represent label relationships (mutual exclusion, overlap, and subsumption) and output label probabilities. This approach replaced the decision-making stage of traditional classifiers (softmax or independent logistic regressions) with HEX graphs. It did not utilize feature space decision-making process, but op-
erated in the semantic space by mapping data to a semantically-built graph. Dogan et al. [128] proposed label-similarity curriculum learning (LCL) to gradually update the label representations from word embeddings to one-hot encodings. This approach operates in between the feature and semantic spaces. While LCL showed classification accuracy benefits of using the semantic space information, it allowed one space to dominate the other by gradually shifting label representations from the semantic space to the feature space. Wang et al. [129] performed zero-shot recognition with word embeddings and knowledge graph (KG) that includes links between different classes. Use of the additional information resource, i.e., KG, leads to a significant improvement over the state of the art.

Overall, the above-mentioned approaches have had a significant influence on the development of NLP applications; however, they exploit either the feature space or the semantic space when performing classification. In Chapter 3, we propose a dual-space decision process: SECRET. SECRET integrates the feature and semantic spaces. Thus, SECRET can be differentiated from previous work and looks at real-world classification tasks in a new way.

2.2 Stress Detection and Alleviation

Table 2.1 shows previously proposed stress-related studies. It summarizes the number of participants (#P), WMSs used, stressors applied, stress alleviation techniques practiced (if any), whether there is user interaction or not (e.g., through questionnaires, surveys, participant notes, etc.), the accuracy of stress detection, and whether the effectiveness of the stress alleviation stage is compared with the ‘no therapy’ baseline or not (in case stress alleviation is employed). Only studies in [132], [134], and [138] offer a solution for stress alleviation. However, they do not evaluate its effectiveness. In [132], stress level is defined quantitatively after combining time- and frequency-
<table>
<thead>
<tr>
<th>Paper #P</th>
<th>WMSs</th>
<th>Stressor(s)</th>
<th>Stress Alleviation Technique(s)</th>
<th>User Interaction</th>
<th>Stress Detection Accuracy (%)</th>
<th>Stress Alleviation Effectiveness</th>
</tr>
</thead>
<tbody>
<tr>
<td>130</td>
<td>30</td>
<td>ECG, RESP, EMG, GSR</td>
<td>Calculation Task, Logical Puzzle, Memory Task</td>
<td>Yes</td>
<td>74.5</td>
<td>-</td>
</tr>
<tr>
<td>131</td>
<td>4</td>
<td>EMG, ECG, RESP, GSR</td>
<td>Driving Tasks</td>
<td>Yes</td>
<td>97.4</td>
<td>-</td>
</tr>
<tr>
<td>132</td>
<td>30</td>
<td>PPG</td>
<td>Daily Stress</td>
<td>Controlled Respiration</td>
<td>Yes</td>
<td>N/P</td>
</tr>
<tr>
<td>133</td>
<td>21</td>
<td>Skin TEMP, Ambient TEMP, ACC, ECG, GSR, RESP</td>
<td>Public Speaking, Mental Arithmetic, Cold Pressor</td>
<td>Yes</td>
<td>90.2</td>
<td>-</td>
</tr>
<tr>
<td>134</td>
<td>24</td>
<td>Finger TEMP</td>
<td>Verbal, Math</td>
<td>Biofeedback</td>
<td>No</td>
<td>80.0</td>
</tr>
<tr>
<td>135</td>
<td>3</td>
<td>HR Monitor, RESP, GSR</td>
<td>Stroop Color Test, Mental Arithmetic</td>
<td>No</td>
<td>83.0</td>
<td>-</td>
</tr>
<tr>
<td>136</td>
<td>25</td>
<td>BP, GSR</td>
<td>Physical Activity</td>
<td>-</td>
<td>No</td>
<td>N/P</td>
</tr>
<tr>
<td>137</td>
<td>5</td>
<td>GSR, Skin TEMP, PPG, ACC, GYR</td>
<td>Regular Desk Jobs (Daily Stress)</td>
<td>Yes</td>
<td>91.0</td>
<td>-</td>
</tr>
<tr>
<td>Paper #</td>
<td>WMSs</td>
<td>Stressor(s)</td>
<td>Stress Alleviation Technique(s)</td>
<td>User Interaction</td>
<td>Stress Detection Accuracy (%)</td>
<td>Stress Alleviation Effectiveness</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
<td>-------------</td>
<td>---------------------------------</td>
<td>------------------</td>
<td>------------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>[138]</td>
<td>6 HR, Humidity, TEMP, Microphone, Movement</td>
<td>Physical Activity</td>
<td>Biofeedback</td>
<td>No</td>
<td>85.7</td>
<td>N/P</td>
</tr>
<tr>
<td>[139]</td>
<td>35 HR Variability, Microphone, ACC, GPS, Phone calls, Address book, Calendar, Battery</td>
<td>Regular Desk Jobs (Daily Stress)</td>
<td>-</td>
<td>Yes</td>
<td>61.0</td>
<td>-</td>
</tr>
<tr>
<td>[140]</td>
<td>10 HR Monitor, RESP, GSR, EMG</td>
<td>Dual Tracking, Memory Search, Mirror Tracing, Stroop Color Test, Public Speech</td>
<td>-</td>
<td>Yes</td>
<td>81.0</td>
<td>-</td>
</tr>
<tr>
<td>[141]</td>
<td>42 ECG</td>
<td>Verbal University Examination</td>
<td>-</td>
<td>No</td>
<td>90.0</td>
<td>-</td>
</tr>
<tr>
<td>[142]</td>
<td>8 ECG</td>
<td>Stroop Color Test, Video</td>
<td>-</td>
<td>No</td>
<td>80.0</td>
<td>-</td>
</tr>
<tr>
<td>[143]</td>
<td>32 GSR, BVP, Pupil Diameter, Skin TEMP</td>
<td>Stroop Color Test</td>
<td>-</td>
<td>No</td>
<td>90.1</td>
<td>-</td>
</tr>
<tr>
<td>Paper #</td>
<td>WMSs</td>
<td>Stressor(s)</td>
<td>Stress Alleviation Technique(s)</td>
<td>User Interaction</td>
<td>Stress Detection Accuracy (%)</td>
<td>Stress Alleviation Effectiveness</td>
</tr>
<tr>
<td>---------</td>
<td>---------------------------</td>
<td>-----------------------------</td>
<td>---------------------------------</td>
<td>------------------</td>
<td>-------------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>144</td>
<td>EMG, ECG, EEG, GSR, BO</td>
<td>Physical Activity</td>
<td>-</td>
<td>Yes</td>
<td>85.2</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cognitive Task</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>145</td>
<td>Accelerometer, GSR, Mobile Phone</td>
<td>Daily Stress</td>
<td>-</td>
<td>Yes</td>
<td>100</td>
<td>-</td>
</tr>
<tr>
<td>146</td>
<td>HR Monitor, EMG, GSR</td>
<td>Dual Tracking</td>
<td>-</td>
<td>No</td>
<td>65.5</td>
<td>-</td>
</tr>
<tr>
<td>SoDA</td>
<td>ECG, BP, GSR, RESP, BO</td>
<td>Memory Game, Fly Sound, IAPS, Ice Test</td>
<td>Micro-meditation, Warm Stone, Good News</td>
<td>No</td>
<td>95.8</td>
<td>Faster relief than ‘no therapy’ baseline</td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
domain analyses. In order to show the effect of controlled respiration (therapy), the
stressor and therapy are applied during different tasks, and the proposed stress allevi-
ation technique is concluded to be working by comparing corresponding stress levels.
However, there is also a possibility that the participant may have relaxed even without
stress alleviation. Since controlled breathing is not the only parameter varying in the
comparison, the results do not establish the effectiveness of the therapy rigorously.
Moreover, no stress detection accuracy is reported. In [134], finger temperature data
are used for stress detection and reported to have a classification accuracy of 80.0%.
When stress is detected, biofeedback is proposed for therapy. However, biofeedback is
not compared against the 'no therapy' baseline and no quantitative analysis is carried
out to show its effectiveness. In [138], a wearable biofeedback system is proposed that
triggers an alarm when stress is detected. However, again, no quantitative evaluation
of the effectiveness of the stress alleviation method is presented.

The other studies either do not employ multiple stressors [131, 138, 141, 143,
146], have low stress detection accuracy [130, 134, 135, 138, 139, 140, 142,
144, 146], distract the participant with a questionnaire, survey, etc. [130, 131,
132, 133, 137, 139, 140, 144, 145] or include very few participants [131,
135, 137, 138, 140, 142, 145, 146]. However, multiple stressors are required
to model various real-life situations and arrive at more realistic conclusions. High
detection accuracy (> 90%) is needed to respond to the user’s needs effectively and
in real-time. User interaction needs to be minimized as much as possible to ensure a
user-friendly and minimally distracting system. Finally, the tests need to be carried
out on a sufficient number of participants to assess user needs and system performance
more effectively. The studies in [131] and [145] either include only one stressor or few
participants, however, report high stress detection accuracy. Since the study in [131]
experiments with only four participants and one stressor, it is difficult to generalize
its conclusions. Moreover, it uses questionnaires, which may distract the participants
and are impractical in real-life situations. In [145], the classification accuracy is computed through 10-fold cross-validation. However, since the data collected by WMSs constitute a time series and since cross-validation ignores time dependencies, cross-validation should not be used in this case [147]. Instead of cross-validation, the data can be divided into training and testing parts for model generation and evaluation. In Chapter [4] we introduce SoDA, which overcomes the shortcomings of the above-mentioned methods. It enables both automatic stress detection and alleviation in a user-transparent manner, and provides quantitative evaluations using multiple WMSs, stressors, and therapies. It also offers high classification accuracy.

2.3 Physical-, Mental-, and Emotional-State Detection

Table 2.2 presents previous physical, mental, and emotional state-related studies. It summarizes the states analyzed, links among them, whether the experiments are performed in a real-life situation (i.e., in an uncontrolled or random environment), whether their effect on satisfiability of fundamental human needs is analyzed, total number of participants, and the overall physical/mental/emotional state detection accuracy. The number of participants (#P) in the studies ranges between 3 and 213,112. Nine out of 16 studies involve 10 or fewer participants. All the previous studies focus on a single state (e.g., physical, mental, or emotional). They do not consider the other states, therefore ignore the links between them. The study in [152] does not analyze mental or emotional states, but assesses the location information in addition to the physical state of the users. Thus, it does highlight the need for multi-aspect analyses. The study in [159] also focuses on a single state (emotional); however, it establishes communication with the subject (provides/receives feedback) during the experiments. Moreover, as none of these studies analyze more than one
Table 2.2: Physical/Mental/Emotional State-related Studies and Corresponding Information on Setting of the Experiment, and Accuracy

<table>
<thead>
<tr>
<th>Paper</th>
<th>Physical states analyzed</th>
<th>Mental states analyzed</th>
<th>Emotional states analyzed</th>
<th>Links btw. states based on real-life aspect</th>
<th>Human needs</th>
<th>#P</th>
<th>ACC (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[148]</td>
<td>Sitting down, Standing up, Reaching, Walking, Turning</td>
<td>-</td>
<td>-</td>
<td>No</td>
<td>No</td>
<td>9</td>
<td>90.0</td>
</tr>
<tr>
<td>[149]</td>
<td>Sawing, Filing/Drilling, Sanding, etc.</td>
<td>-</td>
<td>-</td>
<td>No</td>
<td>No</td>
<td>5</td>
<td>98.3</td>
</tr>
<tr>
<td>[150]</td>
<td>Lying, Walking (Slow/Normal/Fast), Fall (Active/Inactive/Chair), Sit-to-stand, etc.</td>
<td>-</td>
<td>-</td>
<td>No</td>
<td>No</td>
<td>10</td>
<td>90.8</td>
</tr>
<tr>
<td>[151]</td>
<td>Lying, Sitting/Standing, Walking, Running, etc.</td>
<td>-</td>
<td>-</td>
<td>Yes/No</td>
<td>No</td>
<td>12</td>
<td>89.0</td>
</tr>
<tr>
<td>[152]</td>
<td>Walking, Lying, Bicycling, Running, etc.</td>
<td>-</td>
<td>-</td>
<td>Yes</td>
<td>No</td>
<td>60</td>
<td>87.0</td>
</tr>
<tr>
<td>[153]</td>
<td>Sitting, Laying, Standing, Walking, Jogging</td>
<td>-</td>
<td>-</td>
<td>Yes</td>
<td>No</td>
<td>10</td>
<td>99.0</td>
</tr>
<tr>
<td>[154]</td>
<td>Drowsy, Very drowsy, Awake</td>
<td>-</td>
<td>-</td>
<td>No</td>
<td>No</td>
<td>20</td>
<td>80.6</td>
</tr>
<tr>
<td>Paper</td>
<td>Physical states analyzed</td>
<td>Mental states analyzed</td>
<td>Emotional states analyzed</td>
<td>Links btw. states analyzed</td>
<td>Exp. based on real-life aspect</td>
<td>Human needs</td>
<td>#P</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------------</td>
<td>------------------------</td>
<td>---------------------------</td>
<td>----------------------------</td>
<td>-------------------------------</td>
<td>-------------</td>
<td>----</td>
</tr>
<tr>
<td>155</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Yes</td>
<td>No</td>
<td>10</td>
</tr>
<tr>
<td>156</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Yes</td>
<td>No</td>
<td>213,112</td>
</tr>
<tr>
<td>157</td>
<td>-</td>
<td>-</td>
<td>Sadness, Disgust</td>
<td>-</td>
<td>No</td>
<td>No</td>
<td>12</td>
</tr>
<tr>
<td>158</td>
<td>-</td>
<td>-</td>
<td>Anger, Boredom, Disgust, Fear, Happiness, Sadness, Neutral</td>
<td>-</td>
<td>No</td>
<td>No</td>
<td>10</td>
</tr>
<tr>
<td>159</td>
<td>-</td>
<td>-</td>
<td>Positive valence/Low arousal/High Dominance, Negative Valence/High Arousal/Low Dominance</td>
<td>-</td>
<td>No</td>
<td>No</td>
<td>20</td>
</tr>
<tr>
<td>160</td>
<td>-</td>
<td>-</td>
<td>High/Low stress, Disappointment, Euphoria</td>
<td>-</td>
<td>No</td>
<td>No</td>
<td>10</td>
</tr>
<tr>
<td>161</td>
<td>-</td>
<td>-</td>
<td>Anxiety, Boredom, Engagement</td>
<td>-</td>
<td>Yes</td>
<td>No</td>
<td>20</td>
</tr>
<tr>
<td>162</td>
<td>-</td>
<td>-</td>
<td>Positive/High arousal, Negative/High arousal, Negative/Low arousal, Positive/Low arousal</td>
<td>-</td>
<td>Yes</td>
<td>No</td>
<td>3</td>
</tr>
<tr>
<td>Paper</td>
<td>Physical states</td>
<td>Mental states</td>
<td>Emotional states</td>
<td>Links btw. states</td>
<td>Exp. based on needs</td>
<td>Human real-life aspect</td>
<td>#P</td>
</tr>
<tr>
<td>-------</td>
<td>----------------</td>
<td>---------------</td>
<td>------------------</td>
<td>------------------</td>
<td>-------------------</td>
<td>---------------------</td>
<td>----</td>
</tr>
<tr>
<td></td>
<td>analyzed</td>
<td>analyzed</td>
<td>analyzed</td>
<td>analyzed</td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Anger, Anxiety, Boredom, Disgust, etc.</td>
<td>-</td>
<td>Yes</td>
<td>No</td>
<td>85.0 (video)</td>
<td></td>
</tr>
<tr>
<td>SUY</td>
<td>Driving, Sleeping, Stationary, Typing, Walking</td>
<td>Bored, Anxious, Normal, Sleepy, Thirsty, Full, Hungry, Tired, Excited, Preoccupied</td>
<td>High arousal-valence, High valence, Low arousal-valence, High valence, Low arousal-valence, High arousal-valence, Low valence</td>
<td>Physical-Mental</td>
<td>Yes</td>
<td>Yes</td>
<td>90.0 (Physical), 90.3 (Mental), 98.4 (Emotional-4), 99.5 (Emotional-2)</td>
</tr>
</tbody>
</table>

[163]
human state, they cannot analyze the in-between links. Studies in [152], [153], [155], [156], [161], [162], and [163] are based on real-life experiments. The study in [151] is based on experiments partially carried out in a real-life and partially in a controlled environment. The remaining studies limit the participant to a specific experimental protocol/environment, and do not introduce any randomness. Moreover, none of the studies analyze their designs in the context of addressing fundamental human needs. Although these studies report successful physical/mental/emotional state detection performance, they remain task-oriented instead of being user-centric. This is the main reason behind the current gap between AI technologies and humans.

YSUY, presented in Chapter 5, gets around the above-mentioned problems. It analyses user states from three different perspectives: physical, mental, and emotional. It depends on experiments performed in real-life situations (physical and mental state experiments in an uncontrolled environment and emotional state experiments in a randomness-including environment), considers the links between the states, and targets fulfillment of fundamental human needs. It keeps users at the center of its design, aided by high state classification performance.

2.4 IoT Systems from Smartness, Security and Energy-Efficiency Perspectives

IoT sensors are widely used in various applications such as healthcare, agriculture, industry, transportation, independent living, energy management and optimization, public safety, etc. Based on the user’s needs and system goals, IoT applications either utilize a single sensor or an array of sensors included in a sensor node.

IoT sensors collect data from the environment. However, the collected data are typically raw (i.e., require signal processing before any action can be taken) and the sensor node has limited processing and storage resources. In order to carry out the
processing and make a decision, data are transmitted to the cloud servers through base stations. However, data transmission poses serious design obstacles: insufficient energy, limited bandwidth, and security vulnerabilities. Limited available energy necessitates frequent battery replacement or recharging of the sensor node. This negatively impacts the practicality of the deployed IoT system. Limited bandwidth increases decision-making latency owing to the busy Internet Protocol (IP) traffic. The IP traffic increases as the number of IoT devices connected to and utilizing the Internet increases, thus further exacerbating the bandwidth problem. CISCO’s analysis of IP traffic provides support for this trend [164]. This analysis indicates that the rate of IP traffic in 2016 was 1.2 ZB per year. In 2021, it is expected to increase by approximately 3×, reaching 3.3 ZB per year.

In terms of security, various IoT sensors have been shown to be vulnerable to attacks that reveal sensitive information of the user or system. In one case, 73,011 security cameras in a large number of countries were attacked [165]. Since these cameras used publicly known login information (i.e., username and password), user privacy could be compromised without much effort. Moreover, recently a smart TV company was charged with collecting, storing, and marketing users’ sensitive information without obtaining prior consent [166]. In October 2016, a distributed denial-of-service (DDoS) attack was carried out in the United States and Europe by infecting a significant number of IoT devices with the Mirai malware [167]. This is reported to be the most widespread DDoS attack in history. It exposed the security vulnerabilities of IoT devices and the seriousness of the consequences of deploying such devices [167]. As another example, Mosenia et al. analyzed 70 on-market IoT sensors from a security point of view [168]. They demonstrated attacks on eight of these sensors when employed in two different IoT applications: residential and industrial automation/monitoring. In the case of the residential automation/monitoring system, they targeted motion, door, and smoke detector sensors. They could obtain
the pin numbers of these sensors in a few seconds and reverse-engineer the wireless packet format of the sensors. In the industrial system setting, they targeted fluid level sensors. Reverse engineering of the packet allowed them to generate malicious packets that are not recognized as malicious by the base station. This allowed them to generate continuous false alarms in order to wear the user down into deactivating the alarm system, which could be a precursor to a more serious attack on the system. To minimize these security vulnerabilities in IoT sensors, time-stamps, cryptographic techniques, an uncommon username/password combination, and a longer sensor identification number could be utilized [168]. However, incorporation of these techniques adversely impacts the energy consumption of the IoT sensor by either increasing the packet size in data transmission or the amount of computation in processing and decision-making operations. For example, incorporating cryptographic techniques (i.e., encryption and hashing) increases energy consumption by approximately 150% compared to the traditional sense-and-transmit approach [71]. Therefore, simultaneously achieving energy efficiency and security is difficult. Moreover, imparting smartness to IoT sensors requires additional computations for feature extraction and classification, thus increasing overall energy consumption further. On the other hand, we need sensor-side:

- smartness to alleviate system bandwidth concerns,
- security to avoid confidentiality/integrity attacks on sensitive sensor data, and
- energy efficiency in order to enable long-term use without the need for frequent battery change or recharging.

Fig. 2.1 shows various scenarios of how IoT sensor data can be transmitted and/or processed for two different scenarios: alert notification (Fig. 2.1a) and continuous notification (Fig. 2.1b). Alert notification is applicable to scenarios when the base station needs to be notified when a rare event, such as arrhythmia, has been detected.
Figure 2.1: IoT sensor data: (a) alert and (b) continuous notification scenarios.

through on-sensor inference. Continuous notification is applicable to scenarios when the base station needs to be continuously notified of the on-sensor inference outcome, e.g., in the case of human activity detection. In both approaches, sending all the raw data from a traditional sense-and-transmit IoT sensor to the base station or cloud is energy-intensive (even more so when the data are encrypted/hashed before transmission). This is depicted by the red arrows. When no encryption/hashing is employed, the data are vulnerable to eavesdropping and confidentiality/integrity attacks. This is depicted by the blue arrow. The propagation of decisions from the base station or cloud to user-side applications can also be energy-intensive and vulnerable to manipulation. The green arrows depict the paths taken by our approach
that is presented in Chapter 6. Within the sensor node, we compress/process the
data using two different approaches: direct computations on compressively-sensed
data [51], [52] and CSP [53]. In the former, signal processing is directly performed
on a compressed representation of data, whereas in the latter, signal processing and
compression are carried out in the Nyquist domain. Inference is performed in the
compressed domain itself, based on the IoT application of interest. The inference
outcome determines what data are transmitted further and how often. This enables
us to simultaneously achieve

- smartness through machine learning inferences,

- security through encryption and hashing, and

- energy efficiency through both compression and machine learning inferences,
  which enable a drastic reduction in the amount of data transmitted to the base
  station.
Chapter 3

SECRET: Semantically Enhanced Classification of Real-world Tasks

In this chapter, we present our dual-space (feature and semantic) classification approach, SECRET. On ten different datasets, SECRET achieves up to 14.0% (avg. 2.0%) accuracy and 13.1% (avg. 3.3%) F1 score improvements, compared to traditional supervised learning. Also, compared to ensemble methods, SECRET achieves up to 12.7% (avg. 1.8%) accuracy and 13.3% (avg. 2.8%) F1 score improvements. Moreover, we provide further analyses on the effect of integration of semantic space information with the classification task and SECRET’s performance under various experimental conditions [169].

3.1 Introduction

Significant progress has been made in natural language processing (NLP) and supervised machine learning (ML) algorithms over the past two decades. NLP successes include machine translation, speech/emotion/sentiment recognition, machine reading, and social media mining [170]. Hence, NLP is beginning to become widely used in real-world applications that include either speech or text. Supervised ML algo-
Supervised ML algorithms train on feature-label pairs to model the application of interest and predict labels. The label involves semantic information. Palatucci et al. [122] use this information through vector representations of words to find the novel class within the dataset. Karpathy and Fei-Fei [121] generate figure captions based on the collective use of image datasets and word embeddings. Such studies indicate that data features and semantic relationships correlate well. However, current supervised ML algorithms do not utilize such correlations in the decision-making (prediction) process. Their decisions are only based on the feature-label relationship, while neglecting significant information hidden in the labels, i.e., meaning-based (semantic) relationships among labels. Thus, they are not able to exploit synergies between the feature and semantic spaces.

In this chapter, we show the above synergies can be exploited to improve the prediction performance of ML algorithms. Our method, called SECRET, combines vector representations of labels in the semantic space with available data in the feature space within various operations (e.g., ML hyperparameter optimization and confidence score computation) to make the final decisions (assign labels to datapoints). Since SECRET does not target any particular ML algorithm or data structure, it is widely applicable.

The main contributions of this chapter are as follows:

1. We introduce a dual-space ML decision process called SECRET. It combines the new dimension (semantic space) with the traditional (single-space) classifiers that operate in the feature space. Thus, SECRET not only utilizes available data-label pairs, but also takes advantage of meaning-based (semantic) relationships among labels to perform classification for a given real-world task.
2. We demonstrate the general applicability of SECRET to various supervised ML algorithms and a wide range of datasets for various real-world tasks.

3. We demonstrate the advantages of SECRET’s new dimension (semantic space) through detailed comparisons with traditional ML approaches that have the same processing and information (except semantic) resources.

4. We compare the semantic space ML model with traditional approaches. We shed light on how SECRET builds the semantic space component and its impact on overall classification performance.

The remainder of the chapter is organized as follows. Section 3.2 introduces the methodologies underpinning the SECRET architecture, data processing, hyperparameter tuning, ML algorithm training in the feature and semantic spaces, confidence score calculation, and decision process. Section 3.3 presents experimental results and provides comparisons with traditional ML approaches. Section 3.4 provides a detailed discussion on SECRET from different perspectives. Finally, Section 3.5 concludes the chapter.

3.2 Methodology

In this section, we describe SECRET’s data processing and dual-space classification procedure in detail.

3.2.1 The SECRET Architecture

SECRET integrates information from two sources: feature space and semantic space.

- The feature space includes data, extracted features (if available), and the corresponding labels.
• The semantic space includes meaning-based relationships among labels in the form of real-valued word vectors.

As shown in Fig. 3.1a, traditional supervised learning operates in the feature space. It uses the features to model the data-label relationship. On the other hand, SECRET not only uses data available in the feature space, but also integrates meaning-based relationships among labels (semantic space) into the decision process, as shown in Fig. 3.1b. SECRET requires vector representations of the training labels as an additional input, relative to the traditional supervised learning approach. Vector representations are obtained using semantic vector generation algorithms that are trained with a large number of documents. Depending on the available computational resources, SECRET can be implemented with either pre-trained semantic vectors that are available on the web [171], [172], [173], or specially-trained semantic vectors obtained from a given corpus. Neither implementation needs the involvement of an expert, unlike the case of labeling data in supervised learning.

The novelty of SECRET is that it enables an interaction between the two spaces while constructing the classifiers and regressors. In Fig. 3.1b, the interaction is depicted by the arrow in between the two spaces. The hyperparameter values of the semantic (feature) space are not aimed at maximizing the performance of the semantic space regressor (classifier), but that of the overall SECRET architecture. However, the interaction does not only take place during hyperparameter tuning. Unlike the traditional approaches, the classifier and regressor do not make individual decisions. Both provide confidence scores for each label. This information is used by SECRET to predict the label for a new query data instance. We explain each block next.

3.2.2 Data Processing

Data processing is an important part of any ML decision process. Data in the raw form require: (i) denoising [174], [175], [176], [177], (ii) outlier elimination [178], (iii) feature
Figure 3.1: Architectures: (a) traditional supervised learning and (b) SECRET.
3.2.3 Hyperparameter Tuning

The selected set of hyperparameter values has a direct impact on classification/regression performance. In this work, we adopt Bayesian optimization. By integrating exploration and exploitation, Bayesian optimization outputs the set of hyperparameter values that maximizes the optimization goal function. This function indicates the overall performance of the chosen supervised ML algorithm. Therefore, it guides Bayesian optimization to find the appropriate set of hyperparameter values in order to enhance the performance of real-world decision processes. The pseudocode for the hyperparameter tuning stage is shown in Algorithm 1. Following preprocessing of training and validation data, the Gaussian Process (GP) of Bayesian optimization is initialized. Bayesian optimization takes hyperparameters (as variables, not their values), their ranges, and optimization goal function as input. The hyperparameters depend on the chosen ML algorithm. For example, whereas the total number of trees may be a hyperparameter for the random forest (RF) algorithm, the number of layers and neurons in each layer may be hyperparameters for the multi-layer perceptron (MLP) algorithm. The optimization goal function reflects the purpose of the task being performed. Depending on whether SECRET is implemented on top of a traditional supervised (feature space) classifier or built from the ground up, the optimization goal function takes into account either available feature and semantic space hyperparameter values or both semantic and feature space hyperparameter values. The function outputs performance metrics, such as accuracy, F1 score, etc., based on training and validation data. We implement SECRET on top of a traditional supervised classifier in order to compare it with classification algo-
Algorithm 1  SECRET - Hyperparameter Tuning

Input:
DataTr: Training data
DataVal: Validation data
LabelTr: Labels corresponding to training data
LabelVal: Labels corresponding to validation data
FSHyp: Feature space hyperparameter values
V: Vector representations of the labels

Output:
SSHyp: Semantic space hyperparameter values

1: Preprocess DataTr, See Section 3.2.2
2: Preprocess DataVal, See Section 3.2.2
3: ACCmax ← 0
4: SSHyp ← null
5: Initialize GP of Bayesian Optimization
6: for i = 1, ..., BOiter
7: FSModel ← FSClassifier(DataTr,LabelTr,FSHyp)
8: FSConf ← FSModel(DataVal)
9: SSHyp_i ← Exploration+Exploitation
10: SSMODEL_i ← SSRegressor(DataTr,LabelTr,SSHyp_i)
11: SSOOut ← SSMODEL_i(DataVal)
12: for j = 1, ..., #instances
13: for k = 1, ..., C
14: SSConf_jk = \frac{1}{\sum_{l=1}^{D}(V_k(l) - SSOOut_jk(l))^2} \sum_{m=1}^{C} \frac{1}{\sum_{l=1}^{D}(V_m(l) - SSOOut_jk(l))^2}
15: LabelValPred ← argmax_{class}(\text{avg}(FSConf,SSConf))
16: ACC_i ← ACC(LabelVal,LabelValPred)
17: if ACC_i > ACCmax
18: ACCmax = ACC_i
19: SSHyp = SSHyp_i
20: Update GP
21: return SSHyp

Algorithms from the literature. Therefore, in Algorithm 1, the feature and semantic space algorithms are trained with already assigned hyperparameter values and acquisition function outputs (semantic space hyperparameter values), respectively. Then the feature and semantic space confidence scores are calculated and labels are assigned (see Section 3.2.5 for details). The optimization goal function output (classification
performance on validation set) is used to update the beliefs and obtain the next set of semantic space hyperparameter values. The above process is repeated with this new set. When the maximum number of allowed iterations ($BOiter$) is reached, the process is stopped and the set of hyperparameter values ($SSHyp$) that leads to the highest validation set performance is selected for application to the test set.

### 3.2.4 Training of ML Models and Inference

The data-label relationships exist in different forms in the feature and semantic spaces and are captured through ML algorithms. The feature space does not take into account the meaning-based relationships among labels. However, the semantic space takes into account the affinity and dissimilarity information between labels that is captured in a vector form. Hence, whereas the feature space decision process maps data to the label with the help of a classifier, the semantic space decision process relies on a regressor. The choice of the regressor has a direct impact on SECRET’s performance. Thus, for a fixed feature space classifier, we carry out performance analyses with various regressors on the training and validation data and select the one that best maps data to labels. After finding the best set of hyperparameter values in both spaces through joint optimization, we train the ML algorithms. We train the feature space classifier with the selected hyperparameter values, training data, and training labels. Then we train the semantic space regressor with the selected hyperparameter values, training data, and vector representations of the labels. Following the training stage, we perform inference on the test data and obtain the confidence scores. We show the operations corresponding to this stage in lines 3 through 6 in Algorithm 2.

### 3.2.5 Confidence Score Computation and Decision

The inference stage outputs the confidence scores for each data instance for both spaces. The feature space confidence score ($FSConf$) computation depends on the
Algorithm 2  SECRET - ML Training - Inference - Decision

Input:
DataTrVal: Training and validation data
DataTe: Test data
LabelTrVal: Labels corresponding to training and validation data
LabelTe: Labels corresponding to test data
FSHyp: Feature space hyperparameter values
SSHyp: Semantic space hyperparameter values
V: Vector representations of the labels

Output:
ACC: SECRET’s accuracy on the test set
F1: SECRET’s F1 score on the test set

1: Preprocess DataTrVal, See Section 3.2.2
2: Preprocess DataTe, See Section 3.2.2
3: FSModel ← FSClassifier(DataTrVal,LabelTrVal,FSHyp)
4: FSConf ← FSModel(DataTe)
5: SSModel ← SSRegressor(DataTrVal,LabelTrVal,SSHyp)
6: SSOut ← SSModel(DataTe)
7: for $j = 1,...,\#\text{instances}$
8: for $k = 1,...,C$
9: $\text{SSConf}_{jk} = \frac{\sum_{l=1}^{D} (V_{k(l)} - SSOut_{jk(l)})^2 + \epsilon}{\sum_{m=1}^{C} \sum_{l=1}^{D} (V_{m(l)} - SSOut_{jk(l)})^2 + \epsilon}$
10: LabelTePred ← argmax
11: ACC ← ACC(LabelTe,LabelTePred)
12: F1 ← F1(LabelTe,LabelTePred)
13: return ACC, F1

chosen ML algorithm. For example, the confidence score of the RF classifier is computed as the average class probabilities of all trees. The class probability for each tree is computed as the fraction of the samples that belong to the class of interest in a leaf. However, the confidence score of the MLP classifier is the same as the output of the activation function in the outermost (final) layer. On the other hand, as shown in Fig. 3.2, the semantic space confidence score, $SSConf$, is based on Euclidean distance, in line with the main motivation behind semantic vector representations. As shown in Eq. (3.1), $SSConf$ is computed through the normalized inverse ratio of the squared distance between the assigned vector and label vector. In Algorithm 2, the
Figure 3.2: Confidence score computation in the semantic space. Confidence score is calculated for each $k$, where $k \in [1, C]$ and $C$ represents the total number of classes.

corresponding operations are shown in lines 7 through 9. $D$, $V$, and $C$ represent the dimension of the semantic word vector, semantic vector of the class label, and total number of classes, respectively. $\epsilon$ refers to additive shift. In line 9, $\epsilon$ is used to avoid divergence of the algorithm when the assigned vector (regressor output) overlaps with the vector of the class label. The value of $\epsilon$ needs to be smaller than the minimum difference between vectors of the assigned label and the class label. However, in the hyperparameter tuning stage (line 14 in Algorithm 1), $\epsilon$ is assigned 0 and divergence is permitted to avoid overfitting. Since divergence blocks label assignment, an overlap between the assigned and class labels degrades classification performance on the validation set significantly. Therefore, the hyperparameters corresponding to this case are not selected (line 17 through line 19 in Algorithm 1) and overfitting is avoided.

In summary, while assigning 0 to $\epsilon$ in the hyperparameter tuning stage is beneficial for preventing overfitting, a nonzero value in the decision making process is needed to avoid divergence.

$$SSConf_k = \frac{1}{\sum_{i=1}^{C} \frac{1}{d_i + \epsilon}}, k \in [1, C],$$

where $C$: Total number of classes,

$d$: Distance between assigned vector and class label,

$\epsilon$: Additive shift.

The overall confidence score is computed by taking the average of $FSConf$ and $SSConf$ for each class (Fig. 3.3). In the decision stage, each data instance in the
Figure 3.3: Overall confidence score computation and decision-making stages of SEEK-RET.

test set is assigned the label of the class that has the highest overall confidence score (line 10 in Algorithm 2). Following the labeling stage, SEEK-RET’s classification performance is assessed through accuracy ($ACC$) and F1 score metrics computed using Eq. (3.2) and Eq. (3.3), respectively. Accuracy depicts the ratio of the number of correctly classified instances and the total number of instances. However, the F1 score indicates the fraction of correctly classified instances for each class within the dataset.

\[
ACC = \frac{TP + TN}{TP + TN + FP + FN}
\]

where $TP$: True positive, $TN$: True negative, $FP$: False positive, $FN$: False negative.

\[
F1 = \frac{\sum_{i=1}^{C} 2 \times PREC_i \times REC_i / (PREC_i + REC_i)}{C},
\]

where $PREC = \frac{TP}{TP + FP}, REC = \frac{TP}{TP + FN}$

\[C\]: Total number of classes,

$PREC$: Precision, $REC$: Recall.
3.3 Experimental Results and Discussion

In this section, we present the experimental results for SECRET and provide comparisons with traditional supervised classifiers and ensemble methods. Then, we analyze the effect of feature-semantic space variations on SECRET’s classification performance.

3.3.1 Datasets

SECRET’s flexible design ensures applicability to a broad spectrum of real-world classification tasks. We analyze its performance on datasets for ten different applications, ranging from biomedical disease diagnosis to sonar-based object detection. Table 3.1 describes these datasets and their characteristics. The datasets are taken from the UCI Machine Learning Repository [179]. They focus on the classification task.

The UCI Connectionist Bench Dataset is based on sonar signals that are reflected from a rock or metal cylinder. It discriminates between these two obstacles. The UCI Indian Liver Patient Dataset is composed of patient records, such as age, gender, total bilirubin, total protein, albumin, etc. It is aimed at diagnosing liver disease. The UCI Breast Cancer Wisconsin Dataset is built using images of a fine needle aspiration of the breast. It focuses on classifying the cell nucleus as malignant or benign. The Statlog Dataset is composed of physiologic signal and demographic information of patients. It is aimed at predicting the absence or presence of heart disease. The UCI Contraceptive Method Choice Dataset is composed of demographic and socioeconomic information of married women. It is aimed at identifying their contraceptive method choices. The UCI Lymphography Dataset includes features extracted from lymphography images. It is aimed at classifying different types of lymph nodes. The UCI Nursery Dataset includes parental occupation, child’s nursery condition, family structure, and the family’s social, health, and financial status as features. It
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Abbreviation</th>
<th># Instances</th>
<th># Features</th>
<th># Classes</th>
<th>Class Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCI Connectionist Bench (Sonar, Mines vs. Rocks) Dataset</td>
<td>sonar</td>
<td>208</td>
<td>60</td>
<td>2</td>
<td>Rock, Metal cylinder</td>
</tr>
<tr>
<td>UCI ILPD (Indian Liver Patient Dataset)</td>
<td>liver</td>
<td>583</td>
<td>10</td>
<td>2</td>
<td>Liver patient, Not liver patient</td>
</tr>
<tr>
<td>UCI Breast Cancer Wisconsin (Diagnostic) Dataset</td>
<td>wdbc</td>
<td>569</td>
<td>30</td>
<td>2</td>
<td>Benign, Malignant</td>
</tr>
<tr>
<td>UCI Statlog (Heart) Dataset</td>
<td>heart</td>
<td>270</td>
<td>13</td>
<td>2</td>
<td>Absence, Presence</td>
</tr>
<tr>
<td>UCI Contraceptive Method Choice Dataset</td>
<td>cmc</td>
<td>1473</td>
<td>9</td>
<td>3</td>
<td>No use, Short-term methods, Long-term methods</td>
</tr>
<tr>
<td>UCI Lymphography Dataset</td>
<td>lymph</td>
<td>148</td>
<td>18</td>
<td>4</td>
<td>Normal find, Metastases, Malign lymph, Fibrosis</td>
</tr>
<tr>
<td>UCI Nursery Dataset</td>
<td>nursery</td>
<td>12960</td>
<td>8</td>
<td>5</td>
<td>Not recommended, Recommended, Very recommended, Priority, Special Priority</td>
</tr>
<tr>
<td>UCI Cardiotocography Dataset</td>
<td>cardio</td>
<td>2126</td>
<td>21</td>
<td>10</td>
<td>Calm sleep, REM sleep, Calm vigilance, Active vigilance, Shift pattern, Stress situation, Vagal stimulation, Largely vagal stimulation, Pathological state, Suspect pattern</td>
</tr>
<tr>
<td>UCI Chess (King-Rook vs. King) Dataset</td>
<td>chess</td>
<td>28056</td>
<td>6</td>
<td>18</td>
<td>Draw, Zero, One, Two, Three, Four, Five, Six, Seven, Eight, Nine, Ten, Eleven, Twelve, Thirteen, Fourteen, Fifteen, Sixteen</td>
</tr>
</tbody>
</table>
is targeted at ranking of nursery school applications. The UCI Cardiotocography Dataset is formed using cardiotocography features that are based on fetal heart rate and uterine contraction. It focuses on classification of ten different fetal morphologic patterns. The UCI Chess Dataset is built using the king-rook and rook positions on a chessboard. It is targeted at the depth of a win. The UCI Letter Recognition Dataset is formed using black-and-white image pixels of letters of the English alphabet. It is targeted at classifying each of the 26 letters.

### 3.3.2 Supervised Classifier vs. SECRET

We hypothesize that feature space information is not the only source of information that can be used for classification. In order to test this hypothesis, we compare the classification performance of the supervised classifier with that of SECRET. The supervised classifier uses feature space information to model the data-label relationship and predict labels of unlabeled data instances. On the other hand, SECRET fuses the feature and semantic space information to predict labels. By analyzing the two approaches, we aim to identify the impact of semantic space information on classification performance. In order to minimize dependency on an ML algorithm, we use two classifiers (RF and MLP) and their regressor versions. Decision trees in RF are information/impurity-based; however, MLP is error-based. Second, in order to avoid a biased evaluation of classification performance, we compute both accuracy and F1 scores by comparing the predicted labels with actual ones in the test set. Accuracy reflects the percentage of correctly classified samples within the test set. However, the F1 score incorporates precision and recall values, which are computed from the false positive, false negative, and true positive values for each class, and then taking the average. Therefore, accuracy and F1 score assess the classification performance from different perspectives.
Figure 3.4: Legend for experiments that compare traditional approaches with SECRET.

We implement SECRET with a Bayesian optimization framework [180] (used for determining the number of neurons in an MLP with a single hidden layer and the number of trees in RF), scikit-learn [181], and 50-dimensional GloVe vectors [171] (pretrained with Wikipedia 2014 + Gigaword 5). In the case of a label with more than one word, we take the average of the word vectors corresponding to the words within the label to find the overall label vector. We compute the semantic space confidence score using the equations shown in line 14 of Algorithm 1 and line 9 of Algorithm 2. Based on our analyses (see the conditions mentioned in Section 3.2.5) and IEEE-754 Floating Point Standard [182] on representation of the 64-bit floating point numbers, $\epsilon$ is assigned $10^{-200}$ in the experiments. For classification performance analyses, we use stratified 10-fold sampling for each dataset and report the average accuracy and F1 score values.

Fig. 3.4 shows the legend for the plots that depict classification performance of the traditional feature space approach and SECRET. The starting points of the arrows indicate accuracy and F1 scores of the feature-space classifiers. The ending points indicate the impact of including semantic space information on classification. The numbers above/below the arrows show the percentage improvement. The arrow sizes are scaled accordingly. Dataset names are ordered based on the amount of change in classification performance caused by SECRET.
Figure 3.5: SECRET’s test set (a) accuracy and (b) F1 score improvements over the traditional MLP (feature-space) classifier. SECRET uses MLP as the feature space classifier and RF/MLP as the semantic space regressor. Black arrows indicate when the RF regressor is used, whereas the dark blue and dashed arrows correspond to the MLP regressor.

Fig. 3.5 shows the accuracy and F1 scores of the traditional MLP classifier and SECRET with the format shown in Fig. 3.4. In this case, SECRET integrates semantic information into the MLP classifier with the help of either an RF or MLP regressor. It chooses the type of regressor based on validation set performance, builds the overall classifier using both the semantic and feature spaces, and makes the final decision on the test labels. The color of arrows in Fig. 3.5 indicates the chosen regressor type. If both black and dark blue colored arrows are shown for a dataset, then the validation set classification performance is inconclusive (both regressors perform equally well on the validation set with less than or equal to 1.0% difference in accuracy and F1 score values) in determining the better regressor. We present results with both regressors. The chess dataset has 14.0% and 13.1% improvements in accuracy and F1 scores, respectively. While the liver dataset has 1.2% improvement in accuracy, it has the
second highest F1 score improvement of 13.0% among all datasets. This shows the importance of analyzing classification performance from different perspectives. When there is class imbalance, F1-macro score shines light on classification performance of the minority class. As in the case of the liver dataset, biomedical disease diagnosis datasets are likely to be imbalanced. Although the main goal is to detect the disease, in general, the ‘healthy’ class has more datapoints than the disease one. Class imbalance might affect performance. However, it does not only affect the performance of the feature space classifier, but also that of SECRET. We upsampled the minority class with the SMOTE method and repeated the experiments to test this point. While the feature space resulted in 60.3% accuracy and 58.1% F1 score, SECRET led to 69.8% accuracy and 65.0% F1 score. Although upsampling improved classification performance of both approaches, SECRET’s accuracy and F1 score dominated the traditional feature space classifier by 9.5% and 6.9%, respectively. Moreover, except for the lymph and heart datasets, we observe that the arrows point to the right, indicating that SECRET improves accuracy as well as the F1 score. More specifically, use of the RF or MLP regressor in the semantic space leads to classification performance improvement in nine out of ten and eight out of ten datasets, respectively. The amount of improvement depends on dataset characteristics, feature space classifier, and chosen semantic space regressor.

Fig. 3.6 shows the classification performance of the traditional RF classifier and SECRET implemented with an MLP or RF regressor. In this experimental setup, SECRET shows 11.8% accuracy and 11.7% F1 score improvements for the chess dataset. Moreover, SECRET achieves 4.4% accuracy and 4.6% F1 score improvements for the sonar dataset. For the lymph dataset, we observe a 2.7% accuracy and 4.7% F1 score improvement with the MLP regressor. For the liver dataset, as in the case of Fig. 3.5 a 1.9% increase in the F1 score points to the positive impact of the semantic space information on decreasing the false positive and false negative
Figure 3.6: SECRET’s test set (a) accuracy and (b) F1 score improvements over the traditional RF (feature-space) classifier. SECRET uses RF as the feature space classifier and MLP/RF as the semantic space regressor. Black arrows indicate when the MLP regressor is used, whereas the dark blue and dashed arrows correspond to the RF regressor.

rates, thus increasing the precision and recall values. None of the arrows in Fig. 3.6 point to the left, confirming the stable performance enhancement of SECRET that is independent of application type.

Overall, the results shown in Fig. 3.5 and Fig. 3.6 demonstrate classification performance improvement with SECRET.

3.3.3 Ensemble Method vs. SECRET

We saw in the previous section that SECRET outperforms traditional supervised ML classifiers. However, the traditional classifier also can be made more robust by using an ensemble method. In this section, we compare ensemble methods with SECRET to show that the semantic space offers a different type of information source that pays rich dividends. In order to have a fair comparison between traditional ensemble
methods and SECRET, we replace the red ‘Semantic Space’ block in Fig. 3.1b with a ‘Feature Space’ block. The corresponding block diagram for the ensemble method is shown in Fig. 3.7. The ensemble method is composed of only feature space classifiers. In the experiments, we provide the same amount of processing, hyperparameter tuning, and decision-making resources to the two approaches. The only difference is that only the feature space information is used in the ensemble method, whereas both the feature and semantic space information is used in SECRET. We analyze the ensembles (formed with MLP and RF algorithms) and compare them with SECRET next.

Fig. 3.8 shows the accuracy and F1 scores of the traditional ensemble method and SECRET on the ten datasets. The ensemble is built using an MLP classifier whose performance is maximized with the best set of hyperparameter values. Then this classification performance is enhanced by combining the classifier with another MLP with hyperparameter values that maximize the overall performance of the ensemble. SECRET is built in the same way. However, the feature space classifier is replaced with a regressor that models the data and semantic vector relationship. Since the lymph dataset has the smallest size, the use of one regressor or the other leads to a significant classification performance degradation or some improvement. Due to
Figure 3.8: SECRET’s test set (a) accuracy and (b) F1 score improvements over the traditional MLP-MLP ensemble in the feature space. SECRET uses MLP as the feature space classifier and RF/MLP as the semantic space regressor. Black arrows indicate when the RF regressor is used, whereas the dark blue and dashed arrows correspond to the MLP regressor.

this instability, we do not use the lymph dataset to come to a conclusion. For seven datasets (chess, sonar, cardio, wdbc, liver, heart, and cmc), SECRET achieves a 0.1% to 12.7% higher accuracy and a 0.2% to 13.3% higher F1 score relative to the ensemble method. For the nursery dataset, both approaches show comparable classification performance. Also, as in the experiments described in Section 3.3.2, while the liver dataset has a 0.9% increase in accuracy with SECRET, it obtains the highest F1 score improvement of 13.3%. Overall, as indicated by rightward-pointing arrows, SECRET can be seen to outperform the ensemble method.

Fig. 3.9 shows individual and relative classification performance of the MLP-RF ensemble and SECRET. Again, the lymph dataset shows performance instability due to its size. For chess, cmc, liver, sonar, and cardio datasets, SECRET improves classification performance, whereas for the rest, SECRET either obtains the same or
Figure 3.9: SECRET’s test set (a) accuracy and (b) F1 score improvements over the traditional MLP-RF ensemble in the feature space. SECRET uses MLP as the feature space classifier and RF/MLP as the semantic space regressor. Black arrows indicate when the RF regressor is used, whereas the dark blue and dashed arrows correspond to the MLP regressor.

less than 0.8% lower performance relative to the ensemble method. While SECRET improves the accuracy by 0.2% to 7.6% for the five datasets (excluding lymph), the ensemble method only outperforms SECRET by a maximum of 0.8% in the case of two datasets. SECRET and the ensemble method obtain comparable performance for the nursery and heart datasets.

Fig. 3.10 presents accuracy and F1 scores of the RF-MLP ensemble and SECRET. If we had not implemented SECRET on top of a traditional supervised (feature space) classifier, but built it from ground up, the MLP-RF ensemble would yield the same results as RF-MLP. However, since we would like to compare SECRET with the traditional approach, the hyperparameter values are determined by also taking into account the assigned hyperparameter values of the feature space block. Since SECRET determines the semantic space hyperparameters using joint information from
Figure 3.10: SECRET’s test set (a) accuracy and (b) F1 score improvements over the traditional RF-MLP ensemble in the feature space. SECRET uses MLP as the feature space classifier and MLP/RF as the semantic space regressor. Black arrows indicate when the MLP regressor is used, whereas the dark blue and dashed arrows correspond to the RF regressor.

For the two spaces, for a fair comparison, we provide the same opportunity to the ensemble method while determining the hyperparameter values of the second feature space block. Therefore, while RF hyperparameter values take advantage of the knowledge of MLP hyperparameter values in Fig. 3.9, MLP hyperparameter values take advantage of the knowledge of RF hyperparameter values in Fig. 3.10. Due to very similar validation set performance (≤ 1% difference), we were not able to determine the regressor type for the nursery and cmc datasets. Therefore, we present both results for SECRET with RF and MLP regressors. Although the accuracy improvement of SECRET on the cmc dataset is inconclusive, SECRET obtains consistent F1 score improvements with both regressors. The lymph dataset shows very slight decrease (0.2%) in accuracy; however, improvement (2.0%) in F1 score with SECRET. For the letter and nursery datasets, the ensemble method has less than 0.1% to 0.7%
Figure 3.11: SECRET’s test set (a) accuracy and (b) F1 score improvements over the traditional RF-RF ensemble in the feature space. SECRET uses MLP as the feature space classifier and MLP/RF as the semantic space regressor. Black arrows indicate when the MLP regressor is used, whereas the dark blue and dashed arrows correspond to the RF regressor.

higher F1 score, whereas SECRET has a 0.1% to 12.2% F1 score improvement on the remaining eight datasets.

Fig. 3.11 shows the experimental results for the RF-RF ensemble and SECRET. For the letter dataset, SECRET obtains comparable performance with the ensemble method. For the remaining nine datasets, SECRET provides up to 11.9% and 11.8% accuracy and F1 score improvements, respectively.

From the above experiments, we can conclude that SECRET leads to either significantly higher or comparable classification performance with respect to the ensemble method.
3.3.4 RF Decision Node Depth

In this section, we provide insight into how SECRET’s semantic space RF models differ from the traditional feature space ones. Since SECRET uses meaning-based relationships among labels, it is able to divide the classes into ‘easy-to-classify’ and ‘difficult-to-classify’. We expect SECRET to adjust the RF decision node depths according to both the semantic relationships among labels and data characteristics, and traditional approaches to adjust only according to data characteristics. Therefore, we hypothesize that the decision node depth for different classes varies more in SECRET compared to the traditional approaches as labels show heterogeneous distribution in the semantic space and SECRET is able to integrate this information into the classification task (focus on ‘difficult-to-classify’ classes in deeper nodes with the help of its semantic space component).

We carry out the RF decision node depth experiments on six datasets (cmc, chess, lymph, cardio, nursery, and letter) to validate our hypothesis. The remaining four datasets (sonar, liver, wdbc, and heart) have two classes. When one class is assigned, the other class also gets distinguished. Therefore, the variance of the decision node depth for these four datasets tends to zero, which is not informative. For the six datasets that include three or more classes, we take each decision tree in the RF model and assess the decision nodes, their depth, and assigned classes. It is important to note that the RF models for the feature and semantic spaces are obtained with a classifier and a regressor, respectively. To make a fair comparison, we convert the semantic-space RF regressor to a classifier by performing labeling with only the regressor outputs. For both models, within a tree, we compute the average decision node depth for each class. We repeat this process for each tree to assess the overall average decision node depth for the RF model.

As an example, Table 3.2 shows RF decision node depth variance and classification performance on the cmc dataset for both the traditional approaches and SECRET.
<table>
<thead>
<tr>
<th>Approach</th>
<th>Average RF Node Depth</th>
<th>Overall Variance of RF Node Depth</th>
<th>Classification Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No use</td>
<td>Long-term methods</td>
<td>Short-term methods</td>
</tr>
<tr>
<td>Traditional Classifier</td>
<td>11.6</td>
<td>12.4</td>
<td>12.4</td>
</tr>
<tr>
<td>Traditional Ensemble (built on top of MLP)</td>
<td>11.6</td>
<td>12.5</td>
<td>12.5</td>
</tr>
<tr>
<td>Traditional Ensemble (built on top of RF)</td>
<td>11.5</td>
<td>12.4</td>
<td>12.4</td>
</tr>
<tr>
<td>SECRET (built on top of MLP)</td>
<td>11.0</td>
<td>12.3</td>
<td>12.2</td>
</tr>
<tr>
<td>SECRET (built on top of RF)</td>
<td>11.0</td>
<td>12.3</td>
<td>12.2</td>
</tr>
</tbody>
</table>
While the traditional approaches assign ‘no use,’ ‘long-term methods,’ and ‘short-term methods’ at closer node depths by taking data characteristics into account, SECRET uses both the data characteristics and semantic relationships among labels (Fig. 3.12). As the ‘no use’ class is located farther away (in Euclidean distance) from the ‘long-term methods’ and ‘short-term methods’ classes, SECRET assigns ‘no use’ to shallower depths and focuses on details to distinguish ‘long-term methods’ and ‘short-term methods’ at deeper nodes. We summarize this result into one value, which is the overall variance of the RF node depth among all classes and 10 folds. While SECRET takes the heterogeneous distribution of the labels (in the semantic space) into account and shapes the tree depths accordingly, the traditional classifiers or ensembles ignore this point. They tend to assign labels at similar average node depths, resulting in a smaller overall variance compared to SECRET. As a result of SECRET’s directed attention to ‘easy-to-classify’ and ‘difficult-to-classify’ classes, it outperforms the traditional approaches, as shown in the right column of Table 3.2.

For the remaining datasets, we carried out the same analyses. Fig. 3.13 shows the overall variance of RF decision node depth for ‘Traditional classifier,’ ‘Traditional ensemble,’ and ‘SECRET.’ In Fig. 3.13a and Fig. 3.13b ‘Traditional classifier’ represents the variance of RF model’s decision node depth. In Fig. 3.13a ‘Traditional Ensemble’ and ‘SECRET’ represent variance of decision node depth of RF models.
Figure 3.13: Variance of RF decision node depth for traditional RF classifier and traditional ensemble and SECRET built on top of (a) MLP and (b) RF.

that are built on top of the MLP model, as shown in Fig. 3.1b and Fig. 3.7, respectively. In Fig. 3.13b, ‘Traditional Ensemble’ and ‘SECRET’ are built on top of an RF model. In five of the datasets (except lymph), we observe a larger variance in the overall decision node depth of SECRET compared to the traditional approaches. In
line with this observation, SECRET obtains up to 11.2% and 12.1% accuracy and F1 score improvements, respectively, over the traditional classifier and up to 7.6% and 7.4% accuracy and F1 score improvement in accuracy and F1 score over the traditional ensemble method depicted in Fig. 3.7. For the other case shown in Fig. 3.13b, SECRET obtains up to 11.8% and 11.7% accuracy and F1 score improvements, respectively, over the traditional classifier and up to 11.9% and 11.8% improvements in accuracy and F1 score, respectively, over the traditional ensemble method. For the letter dataset, we observe comparable performance (maximum 0.3% decrease in accuracy/F1 score) with the traditional approaches. For the lymph dataset, while RF node depth variance is smaller for SECRET, we observe 0.1% to 3.6% accuracy and 0.0% to 5.8% improvement over the traditional approaches. This is inconclusive. As we also obtain inconclusive results throughout Section 3.3 due to its size, we do not discuss the lymph dataset further.

Overall, a larger variance in RF node depth indicates that SECRET is distinguishing ‘easy-to-classify’ and ‘difficult-to-classify’ cases more clearly than the traditional approaches and focusing on detailed characteristics at deeper nodes to separate the ‘difficult-to-classify’ cases further. As a result, we observe an enhancement in classification performance with SECRET. This is commensurate with our hypothesis.

3.4 Discussion

In this section, we analyze SECRET from different perspectives.

3.4.1 What to do for classes with the same label, but different meanings?

Word embedding algorithms, such as GloVe, capture the word’s semantic relationships with the remaining words in the dictionary. Words with close meanings are
Figure 3.14: Test set classification performance of SECRET on the cardio dataset when built with (a) MLP and (b) RF as the feature space classifier and RF as the semantic space regressor. Case 1 represents the performance of the cardio dataset when the labels are used as they are. Case 2 and Case 3 represent the performance when the ‘REM sleep’ label is replaced with ‘paradoxical sleep’ and ‘dreaming sleep,’ respectively.

represented by closely-spaced vectors in the semantic space. Therefore, vectors do not represent the sounding, but the meaning relationships between words. None of the datasets used in our experiments (Table 3.1) included homographs (words with the same spelling, but different meanings). However, in case of a homograph (such as ‘spring’), the corresponding semantic vectors should be obtained with context-aware word embedding algorithms, as proposed by studies in [183], [184], [185], [186].

3.4.2 How to decide on the semantic vector of a class whose label has synonyms?

We hypothesize that SECRET provides very similar classification outputs when the synonyms of a class label are used interchangeably. We tested this hypothesis on the cardio dataset. As indicated by studies in [187] and [188], the ‘REM sleep’ label means the same as ‘paradoxical sleep’ and ‘dreaming sleep.’ Fig. 3.14 shows the classification accuracy and F1 scores for three cases when SECRET is built on top of the MLP and RF feature space classifiers. In all cases, SECRET obtains nearly the same classification performance, buttressing our hypothesis.
3.4.3 Do the improvements come from the selected word embedding or SECRET? How does SECRET perform when a different word embedding is introduced?

In order to demonstrate the effectiveness of SECRET and its performance stability on various word vectors, we repeated the experiments with a different set of pretrained word vectors [173] that is obtained using PubMed texts and the word2vec algorithm. PubMed stores biomedical literature and the corresponding word vectors are 200-dimensional. Considering the application area and size of the datasets (we need to avoid overfitting), we focus on the cardio, liver, wdbc, and cmc datasets. Fig. 3.15 shows the classification performance of SECRET when built with MLP as the feature space classifier and RF as the semantic space regressor. Similarly, Fig. 3.16 shows the performance with RF as the feature space classifier as well as the semantic space regressor. In both figures, we observe nearly identical classification performance for Wiki and PubMed-based word vectors. These results provide evidence that the classification performance improvements demonstrated in Figs. 3.5, 3.6, 3.8, 3.9, 3.10, and 3.11 do not arise from Wiki-based word vectors, but the integration of semantic information into the classification task.

3.4.4 Does the semantic space regressor by itself perform better than SECRET?

We carried out additional experiments for the semantic space (only) case. We show the results in Fig. 3.17 and Fig. 3.18 for MLP and RF regressors, respectively. As in the case of the feature space (only), SECRET dominates the semantic space (only) too. This result was expected since SECRET integrates the feature and semantic spaces to benefit from their complementary advantages.
3.4.5 Why did SECRET observe high performance improvement on the chess dataset?

The UCI Chess Dataset includes king-rook and rook positions on a chessboard and the corresponding depth of win for White or draw. The depth of win (zero to 16) and draw conditions account for 18 different labels. Draw indicates the condition when
neither White nor Black dominates the game. On the other hand, the remaining labels (zero to 16) indicate the optimal number of moves for White to win. Semantics of these numbers, e.g., whether they are exact or lower-bounded, have been extensively studied in the literature [189]. Although an assessment of semantics of numbers is out of scope of our study, we analyzed the semantic relationship between labels in terms of squared Euclidean distances. Fig. 3.19 shows the heatmap based on pretrained GloVe embeddings [171]. The heatmap clearly shows smaller Euclidean distances between consecutive numbers (both single-digit and double-digit cases). This indicates consecutive numbers have semantic similarities.

Next, we analyzed similarities between data instances in terms of squared Euclidean distances. Fig. 3.20 shows the heatmap corresponding to feature space vectors, where the vectors are obtained through averaging of the feature vectors for each
class. Size of the class vectors is equal to the number of features. On the other hand, size of the semantic space vectors is equal to the size of the word embeddings. Due to the size and range differences, squared Euclidean distance values are not comparable between Fig. 3.19 and Fig. 3.20. However, the distribution of the heatmap color is comparable and very similar in Fig. 3.19 and Fig. 3.20. This shines light on why performance improves with SECRET. When the feature space values are similar between the two classes, they are difficult to distinguish. Use of one-hot encoding of labels (traditional feature space approach) does not exhibit this challenge. On the other hand, semantic similarity (smaller Euclidean distance) between the labels presents this challenge to the classifier and guides it to distinguish between the classes further. This leads to an improvement of overall classification performance.
It is important to note that Fig. 3.19 and Fig. 3.20 are not exactly the same. This indicates the need for a different information resource. To get around this problem, we use traditional feature space classifiers. However, as demonstrated through the experimental findings, neither the feature space (Fig. 3.5 and Fig. 3.6) nor the semantic
space (Fig. 3.17 and Fig. 3.18) outperforms SECRET. Complementary advantages of both spaces need to be combined as done in SECRET.

3.5 Chapter Summary

In this chapter, we introduced a new dimension (semantic space) to the feature space based decision-making employed in ML algorithms. We call this dual-space classification approach, SECRET. Traditional supervised learning operates in the feature space. SECRET, on the other hand, also incorporates class affinity and dissimilarity information into the decision process. This property enables SECRET to make informed decisions on class labels. Therefore, it is able to deliver higher classification performance relative to traditional approaches because of its reliance on a richer semantic+feature space.
Chapter 4

Keep the Stress Away with SoDA:
Stress Detection and Alleviation
System

In this chapter, we present an automatic stress detection and alleviation system, called SoDA. SoDA uses wearable medical sensors to continuously monitor human stress levels and mitigate stress as it arises. SoDA achieves 95.8% stress detection accuracy. Moreover, SoDA responds to and alleviates stress effectively. Overall, SoDA performs stress detection and alleviation in a user-transparent manner [33].

4.1 Introduction

Stress is a serious health problem that afflicts a large fraction of humanity. In the United States, three out of four visits to the doctor are due to stress-related disorders [190]. In Europe, stress is reported to be the second most common health problem [130]. Stress also has a severe adverse impact on the country’s economy. According to The American Institute of Stress, $300 billion is spent each year on the treatment of stress-induced disorders [190].
Stress can be divided into two parts: stressor and reaction. Stressor is the activity or effect that triggers a change in the physiological parameter values of the human body. Reaction is the deviation of these parameter values from their normal levels. When confronted with a stressor, the body raises an alarm that results in the stress response [191]. The stress response of the body depends on the duration for which the stressor is active. With long and frequent stress responses, the person becomes more likely to develop serious health problems. The health problems vary in a very broad range. For example, excessive exposure to stress may result in depression [192], cardiovascular diseases [193], sleep disorders [194], degradation in the immune system [195], cancer [196], etc. In addition to stressor duration, personal traits also play a significant role in stress response. These traits have an impact on physiological signals, and indirectly on the emotional condition [197].

It is desirable to have immediate stress alleviation when a stress response is detected. Stress alleviation should ideally be tailored to the individual to have maximum impact. This is the motivation behind the stress detection and alleviation system (SoDA) proposed in this chapter. As shown in Fig. 4.1, SoDA first detects stress and then employs a stress alleviation technique based on the stress characteristics of the person. Stress characteristics are deduced from the physiological signals obtained through wearable medical sensors (WMSs). We currently employ five WMSs: ECG, GSR, respiration rate, blood pressure, and blood oximeter. However, the method is not limited to these sensors. More sophisticated and other types of WMSs can be easily incorporated into SoDA as and when they become available. Use of WMSs offers several advantages. First, WMSs continuously collect data from the human body, making it possible to detect a stress response very quickly. Second, they also enable real-time stress mitigation. Alternatively, since the WMS data are typically communicated to an on-body device, such as a smartphone, and thereon to a health server that is accessible to a doctor, it has the potential to enhance the ability of
a doctor to intervene much faster than currently possible. Third, if stress-induced disorders can be significantly reduced, it may bend the national health expenditure curve downwards.

The major contributions of this chapter can be summarized as follows:

1. We implement a flexible stress detection and mitigation system that offers two options to the user: ‘generalized’ and ‘individualized’. In the ‘generalized’ model, the system detects and alleviates stress by using a pre-designed stress model based on data obtained from a population of individuals. An ‘individualized’ model is designed based on the individual’s stress response. The ‘generalized’ model becomes active just after turning on the system, whereas the ‘individualized’ model requires training data from the user for modeling purposes. On the other hand, the ‘individualized’ model is more accurate in discerning if the user is stressed since it is trained on WMS data obtained from the user.
2. We do an extensive search to extract the appropriate set of features from physiological signals to improve the stress detection accuracy.

3. We reduce the need for manual input about stress from the user by relying on data obtained from the WMSs. If stress is detected, the system either employs a fixed (‘generalized’) or an adaptable (‘individualized’) sequence of stress reduction techniques.

4. We improve system accuracy by employing unsupervised dimensionality reduction in conjunction with supervised feature selection.

5. We demonstrate the performance, feasibility, and user-friendliness of our system through extensive evaluations.

The remainder of this chapter is organized as follows. Section 4.2 discusses the methodologies for data collection, experimental procedure, processing of physiological signals, classification, and stress alleviation. Section 4.3 provides the experimental results. Finally, Section 4.4 concludes the chapter with a brief summary.

4.2 Methodology

In this section, we describe the data collection, experimental procedure, data processing, and classification stages of SoDA.

4.2.1 Data Collection

We collect physiological signals through five WMSs: ECG [198], blood pressure monitor [199], GSR [198], respiration monitor [198], and blood oximeter [200]. ECG, GSR, and respiration monitor have a sampling rate of 100 Hz, whereas blood oximeter has a sampling rate of 1 Hz. Blood pressure is not measured continuously. When the individual performs a stress-inducing task, blood pressure measurements are taken in
the beginning, middle, and end. However, for the baseline and individual-under-rest parts of the experiments, blood pressure is measured in the beginning and at the end. The body placement of the chosen WMSs takes into account both the comfort of the participant and the accuracy of the measurements. On-body positions of WMSs are shown in Fig. 4.2.

![Figure 4.2: On-body positions of WMSs.](image)

A total of 33 subjects (8 female, 25 male), with ages between 20 and 28 years, participated in the experiment. All the participants were informed about the experiment and signed a consent form. The experimental procedure was approved by the Institutional Review Board of Princeton University. None of the participants reported mental, cardiac or endocrine disorders. Due to the low quality of physiological signals obtained, the data of one of the participants were excluded from the analysis.

### 4.2.2 Experimental Procedure

For each participant, the laboratory session took approximately 90 minutes. Fig. 4.3 summarizes the experimental procedure. The session starts by welcoming and asking
the participant to sit on a comfortable chair. Then, the experimental procedure and on-body positions of WMSs are explained to the participant in detail. After signing the consent form and wearing the sensors, the participant is reminded of the importance of silence and correct body posture. The participant is encouraged to ask questions. Upon ensuring that the participant is comfortable with the experimental setup and procedure, the experiment commences.

**Baseline:** This is the first stage of the experiment. It is performed to obtain the original levels of the physiological signals. In this stage, the participant is asked to look at the black screen and relax.

**Rest:** A rest period is introduced in between two tests to calm the participant down. A stressful task pushes the physiological signals to deviate from their original levels, thus requiring a rest period to recover. As in the baseline stage, the participant is asked to look at the black screen and relax.

**Memory game:** This game is played on a computer. The participant is shown 40 cards that are flipped back. Two cards are selected in every round. If the cards match, they remain in the face-up position. If they do not match, then both cards
are flipped back and another round commences [201]. The participant is given two minutes to complete this task.

**Fly sound:** In this stage, the participant is asked to listen to the sound of a fly buzzing around, with a black screen shown to prevent distraction [202], [203].

**IAPS:** In this task, the participant is shown pictures from the IAPS Database [204]. The pictures are selected based on the affective ratings specified in the IAPS Technical Manual [204]. Before displaying the pictures, an informative slide (“Get Ready for the Next Slide”) is shown for five seconds. Then, the picture is displayed for seven seconds. This procedure is repeated for a total of 10 pictures. Two sets of tests are performed without (T3) and with (T7) stress mitigation techniques. The corresponding picture numbers for the two sets are as follows:

- Set 1: 1304, 3060, 3170, 3266, 6260, 6313, 9040, 9300, 9413, 9635.
- Set 2: 1525, 3053, 3080, 6520, 9220, 9405, 9410, 9570, 9921, 9940.

**Ice test:** In this test, the participant is asked to place the right hand on top of an ice-filled container. In the event of discomfort, the participant is encouraged to raise the hand and place it back on the ice or finish the test. The test is not started until the ice has melted partially.

### 4.2.3 Stress Mitigation Techniques

SoDA offers various stress mitigation techniques: classical music, micro-meditation, warm stone, and good news. In the ‘individualized’ model, an individual-specific order of these techniques is employed, whereas in the ‘generalized’ model, a fixed sequence is used for each individual. In order to obtain the most effective sequence, the four stress-inducing tasks are carried out with and without stress alleviation techniques. In tasks T1-T8, stressors are applied for 0-120 seconds; however, in tasks T5-T8, in
addition to the stressors, the alleviation techniques are also employed in the 50-120 second range.

**Classical music:** During the memory game (T6), classical music is played starting at the 50th second. The composition set includes Benjamin Godard’s “Berceuse” and Frederic Chopin’s “II. Romance” [205].

**Micro-meditation:** Micro-meditation is a short-duration practice for nurturing self-awareness [206]. It can be employed in various forms. In our experiment, in the presence of the fly sound stressor (T5), the participant is asked to close the eyes and relax various body parts starting from the feet to face. As before, this technique is employed starting at the 50th second of the task and instructions related to body parts that need to be relaxed are provided during the meditation.

**Warm stone:** In this stress mitigation technique, using IAPS pictures as stressors (T7), the participant is asked to hold in the palm a warm stone of size approximately $9 \times 8 \times 2$ cm. The stone is warmed up by placing it in boiling water for two minutes. Then, it is taken out, dried, and placed on its side for another two minutes. At the 50th second of the task, the participant is given the warm stone, with continuing display of selected IAPS pictures on the screen.

**Good news:** While the participant is performing the ice test, positive and optimistic news chosen from [207] are displayed on the screen. Task T8 is started with a black screen. Starting at the 50th second, good news accompanied by a corresponding picture is shown for 10 seconds. A total of seven news items are displayed.

### 4.2.4 Preprocessing and Feature Extraction

To obtain the performance measures, the data obtained from 32 participants are analyzed. The dataset for each participant is composed of $24 \times 2$ minutes of measurements collected through the five WMSs. The five paths in Fig. 4.4 show how the data from these WMSs are processed, as described next.
**ECG**: The ECG signal needs to be denoised first. The denoising steps target baseline wander, power-line interference, muscle noise, etc. Baseline wander is a very-low frequency component that can be caused by perspiration, respiration, and body movements [208]. Given that the lowest observed heart rate is approximately 40 bpm (0.67 Hz), we select a cut-off frequency of 0.5 Hz [208]. A Butterworth zero-phase high-pass filter is employed due to its high quality based on this cut-off frequency [209]. In order to remove muscle noise and the aliased components of power-line interference, the FFT of the ECG signal is plotted. When a peak is observed in the absolute FFT, a notch filter is used to remove the noise. The frequency corresponding to the highest amplitude in the peak is selected as the center frequency of the notch filter [210].

Following the de-noising step, the outliers are replaced with the upper/lower thresholds that are derived from the data. Moreover, range normalization shown in Eq. (4.1) is carried out to eliminate the variability in the physiological signal levels of the 32 participants.

\[
d^*_i = \frac{d_i - \min(d)}{\max(d) - \min(d)}
\]  

(4.1)

After signal preprocessing is complete, a total of 57 ECG features are extracted. This is done by detecting the Q-R-S complex and calculating the corresponding features (e.g., mean, variance, quartile deviation, 80th percentile, etc.) using code im-
plemented in MatLab. For heart rate variability related features, Kubios HRV [211] is utilized. In line with the guidance provided by Task Force [212], intervals for frequency domain computations are determined as very low frequency (VLF, 0-0.04 Hz), low frequency (LF, 0.04-0.15 Hz), and high frequency (HF, 0.15-0.4 Hz). Following the computations in MatLab and Kubios, the extracted feature values are combined and stored with the ones obtained through other WMSs.

**GSR:** The data obtained from this sensor are first subjected to range normalization [Eq. (4.1)] as well. Then, mean, median, and standard deviation of the data are calculated using MatLab. Moreover, continuous and discrete decomposition analyses are carried out using Ledalab [100, 213]. A total of 16 features are extracted from GSR data.

**Respiration Monitor:** The outliers of the data obtained from the respiration monitor are replaced with upper or lower thresholds. After removing the data artifacts, range normalization shown in Eq. (4.1) is performed and feature values are calculated using MatLab. From the respiration data, a total of nine features are extracted: mean, median, and quartile deviation of the respiration duration, root mean square (RMS) of the respiration signal, mean of inhalation and exhalation durations, mean and median of the ratio of inhalation-to-exhalation duration, and mean of stretch.

**Blood Oximeter:** The data obtained from the blood oximeter are also first range-normalized by Eq. (4.1). Then, two features are extracted: mean and variance.

**Blood Pressure Monitor:** This monitor measures the systolic/diastolic pressures and derives the mean arterial pressure (MAP). In order to get comparable feature values across the participants, range normalization shown in Eq. (4.1) is used. Then, the corresponding mean and variance are calculated. A total of six features are obtained from the blood pressure measurements.

Various details of the WMSs are summarized in Table 4.1.
Table 4.1: WMSs, their Abbreviations, Units, and Total Number of Features Extracted

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Abbreviation</th>
<th>Unit</th>
<th>Number of Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrocardiogram</td>
<td>ECG</td>
<td>μV</td>
<td>57</td>
</tr>
<tr>
<td>Galvanic Skin Response</td>
<td>GSR</td>
<td>μS</td>
<td>16</td>
</tr>
<tr>
<td>Respiration</td>
<td>RESP</td>
<td>1/min</td>
<td>9</td>
</tr>
<tr>
<td>Blood Oximeter</td>
<td>BO</td>
<td>%</td>
<td>2</td>
</tr>
<tr>
<td>Blood Pressure</td>
<td>BP</td>
<td>mmHg</td>
<td>6</td>
</tr>
</tbody>
</table>

4.2.5 Feature Selection, Thresholding, PCA, and Classification

Fig. 4.5 shows the block diagram of SoDA. After preprocessing, we extract a total of 90 features from the physiological signals collected by the five WMSs. Since some of these features may be correlated and correlations lead to redundancy, which impacts classification performance negatively, not all of them are used for stress detection [130]. Hence, we divide the data into three parts: first part for training, second part for validation of chosen parameters, and third part for testing. We apply supervised attribute selection to the training part using Weka 3.8.0 [214]. It is composed of forward feature selection and best subset evaluation. In forward feature selection, the system starts with an empty set and searches for features in the forward direction in a greedy fashion. In subset evaluation, each feature is analyzed in terms of its individual contribution to accuracy and its redundancy with respect to the other
The output of attribute selection is a set of features that minimizes redundancy while improving accuracy. This procedure is carried out for each of the 32 participants.

The reduced set of features obtained thus far is then subjected to principal component analysis (PCA). PCA transforms the input information into a group of new orthogonal variables that are linearly uncorrelated. These variables are called principal components. The first principal component has the largest variance, hence includes the largest amount of information about the input data. The second principal component is orthogonal to the first one and has the second largest possible variance. Under the condition of orthogonality, the remaining components are then calculated. Since the majority of data can be represented by the first $n$ components, the remaining ones can be ignored. This enables data compression. Thus, PCA is used to extract the most relevant information from the data, and shown to have a positive effect on classification accuracy. In our case, we extract the most important information through supervised feature selection. However, due to the finite size of training data available, the optimal feature set obtained based on the training dataset may not be optimal for the testing dataset. In order to address this problem, the reduced sets of features from all participants’ data are combined and the number of times each feature appears calculated. Only features that appear more than a predefined threshold are selected and provided as input to PCA for dimensionality reduction. This helps to eliminate features that do not carry too much information. However, the combined set may include redundant features and the training data may not be enough to obtain the optimal feature set. Since PCA is an unsupervised method, it does not take into account the labels of the training data. By calculating orthogonal components and choosing the first $n$ that represent majority of the data, the negative impact on supervised feature selection of the finite-sized training data is abated. In other words, supervised feature selection is used as a coarse sieve, then
PCA is employed as a fine sieve. The number of principal components, \( n \), is obtained based on each participant’s data separately. The principal components are added one by one to the training data. At the \((n + 1)\)-th component, if the classification accuracy on the validation data does not increase, then \( n \) is taken to be the optimal number of principal components.

The data obtained after this step are fed to the classification stage for each of the 32 participants. Due to their widespread applicability and excellent performance, we employ support vector machine (SVM) [216] and \( k \)-nearest neighbor (kNN) [217] for binary classification. In SVM, classification is done by finding a hyperplane that separates the \( n \)-dimensional data into two classes and maximizes the margin. However, since the data are not linearly separable in our case, we use a radial basis function (RBF) kernel. The RBF kernel maps the data to a higher-dimensional space where they are linearly separable [216]. In kNN, the \( k \)-nearest neighbors are determined based on a distance metric (e.g., Euclidean, Minkowski, etc.) and classification is performed using majority voting [217]. In this study, we use Euclidean distance as the distance measure. Moreover, since the ‘generalized’ model is obtained by combining each individual’s data, the optimal \( k \) is different for the ‘individualized’ and ‘generalized’ models. Thus, to have comparable results, kNN is performed for a \( k \) value spanning 1 to 4. Analysis based on different \( k \) values demonstrates the consistent performance of SoDA.

4.2.6 Stress Alleviation

Following SVM or kNN, if stress is detected, SoDA offers stress therapy and modifies the stress alleviation protocol based on the data collected. Algorithm 3 shows the stress alleviation protocol. After the user makes use of the suggested stress mitigation technique, the relevant feature values are traced. If these newly obtained feature values have a tendency towards the ‘no stress’ case, SoDA continues to suggest the
current stress mitigation technique for a period of time, otherwise, it suggests the next technique. This period of time can be modified by the user. According to Algorithm 3, when the stress alleviation technique is observed to reduce stress level for 60 s, the system transitions to the stress detection stage, and checks the user’s stress level. If the user is classified as ‘stressed’ again, then the stress alleviation stage continues with the stress reduction technique. To determine whether the proposed technique is working or not, a majority vote based on all the selected feature value trends is carried out. The feature set used in this stage is not the same as the features used for stress detection. This is because not all features behave the same way when a stressor is applied versus when stress mitigation is applied. For example, even when the body begins to relax, the impact on blood pressure is not immediate. Hence, features derived from blood pressure are not used in the stress alleviation step. In this case, feature values should have an immediate response and should be robust to biological differences. Thus, the features that are appropriate for indicating stress alleviation are selected separately using a feature selection process similar to the one used for stress detection. The aim of stress mitigation techniques is to help the user reach a relaxed state faster than when no such therapeutic technique is employed.

4.3 Experimental Results and Discussion

We present experimental results for SoDA next, followed by a discussion.

4.3.1 Feature Selection, Thresholding, and PCA

Recall that we start with 90 features obtained from the physiological data collected by the WMSs and obtain the best subset for each of the participants. The number of features in this subset ranged from 8 to 22. These reduced sets are combined and the number of times each feature appears is counted. Since there are 32 participants, a
Algorithm 3  Stress Alleviation Protocol

Given: therapySet, set of the stress alleviation techniques.

1: therapy ← null, k ← 0, flag ← 0
2: for i = 1, ..., length(therapySet)
3:   therapy ← therapySet(i)
4: Delay (30sec.)
5: Compute selected N feature values
6: Compute k, number of features showing stress relief
7:   if k ≥ N/2
8:     Delay (30sec.)
9:     Compute selected N feature values
10:    Compute k
11:   if k ≥ N/2
12:     flag ← 1
13:    return
14: end
15: end
16: if flag = 0, when none of the stress alleviation techniques is effective
17: Give warning to the user
18: return
19: end

feature may appear at most 32 times. The recurrence count of the features is shown in Fig. 4.6. This figure indicates that all five physiological signals are indicators of stress. However, compared with the others, the recurrence count of the ECG features is smaller. This is because the total number of features extracted from ECG outnumbers the number of features extracted from the other four physiological signals. In other words, since ECG features provide more options to choose from, the result is a more spread distribution. To get around this problem, we use different recurrence count thresholds for ECG and GSR+RESP+BO+BP derived features. Only the features that are above these thresholds are found to be useful for a larger set of participants and selected for the modeling of stress. This step is followed by PCA.

In Fig. 4.7a and 4.7b, the accuracy of the stress detection stage is analyzed for various thresholds of ECG and GSR+RESP+BO+BP, respectively. Since a threshold
of ten for ECG and eight/nine for GSR+RESP+BO+BP lead to the best or near-best accuracy in all the cases, they are chosen as the thresholds. Thus, features with an ECG recurrence count of ten or above and GSR+RESP+BO+BP recurrence count of eight/nine or above are selected. For both ‘generalized’ and ‘individualized’ cases, if kNN2, kNN3 or SVM is selected as the classifier, a threshold of eight should be used due to its superior classification performance; however, if kNN1 or kNN4 is selected as the classifier, a threshold of nine should be used. The corresponding feature sets are shown in Table 4.2.

The selected features are subjected to PCA for dimensionality reduction. In the ‘individualized’ model, dimensionality reduction is carried out for each participant separately, whereas in the ‘generalized’ model, the combined dataset is used. After computing the corresponding principal components, the first $n$ components are kept and classification accuracy on the validation data is computed. If the inclusion of the $(n+1)^{th}$ component does not improve the accuracy, $n$ is taken to be the re-

Figure 4.6: Recurrence count of features extracted from five physiological signals. Analyses are based on the validation set.
Figure 4.7: Validation set accuracy with respect to (a) ECG and (b) GSR+RESP+BO+BP recurrence count limits (thresholds) for different classifiers.

Reduced dimension. The reduced dimensions for the ‘generalized’ model and statistics of reduced dimensions for the 32 ‘individualized’ models are shown in Table 4.3.

The impact of forward feature selection with subset evaluation and PCA on the classification accuracy is demonstrated in Fig. 4.8 for kNN (k = 1-4) and SVM. The first bar for each classifier represents the case when all 90 features were selected, hence, when no processing was done. Forward feature selection with subset evaluation
Table 4.2: Selected Feature Set for Input to the PCA Stage

<table>
<thead>
<tr>
<th>Feature</th>
<th>Sensor</th>
<th>Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  ECG-derived respiration rate</td>
<td>ECG</td>
<td>8, 9</td>
</tr>
<tr>
<td>2  Mean of skin conductance amplitude</td>
<td>GSR</td>
<td>8, 9</td>
</tr>
<tr>
<td>3  Standard deviation of skin conductance amplitude</td>
<td>GSR</td>
<td>8, 9</td>
</tr>
<tr>
<td>4  Sum of amplitudes of skin conductance responses above the threshold</td>
<td>GSR</td>
<td>8</td>
</tr>
<tr>
<td>(continuous decomposition analysis)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5  Mean of tonic activity</td>
<td>GSR</td>
<td>8</td>
</tr>
<tr>
<td>6  Maximum positive deflection</td>
<td>GSR</td>
<td>8</td>
</tr>
<tr>
<td>7  Mean of respiration duration</td>
<td>RESP</td>
<td>8</td>
</tr>
<tr>
<td>8  RMS of respiration signal</td>
<td>RESP</td>
<td>8, 9</td>
</tr>
<tr>
<td>9  Median of respiration duration</td>
<td>RESP</td>
<td>8</td>
</tr>
<tr>
<td>10 Mean of blood oxygen level</td>
<td>BO</td>
<td>8, 9</td>
</tr>
<tr>
<td>11 Mean of systolic blood pressure</td>
<td>BP</td>
<td>8, 9</td>
</tr>
<tr>
<td>12 Variance of systolic blood pressure</td>
<td>BP</td>
<td>8, 9</td>
</tr>
<tr>
<td>13 Mean of diastolic blood pressure</td>
<td>BP</td>
<td>8, 9</td>
</tr>
<tr>
<td>14 Mean of MAP</td>
<td>BP</td>
<td>8, 9</td>
</tr>
<tr>
<td>15 Variance of MAP</td>
<td>BP</td>
<td>8, 9</td>
</tr>
</tbody>
</table>

followed by thresholding and PCA (third and fourth bars) can be seen to have the highest accuracies in all the cases. This is because forward feature selection with subset evaluation and PCA complement each other. Forward feature selection with subset evaluation is a supervised attribute selection method. It takes training data with labels as input and obtains the corresponding feature set. However, due to the finite size of the training dataset, this method may eliminate some features that are indicators of stress [218]. In order to overcome this problem, the advantage offered by unsupervised dimensionality reduction is exploited. After combining the reduced feature sets and selecting the ones above the threshold, PCA is applied. In PCA, the labels are not taken into account. Hence, the method is not adversely impacted by the training dataset size.

Using reduced dimensions, the performance of both the ‘individualized’ and ‘generalized’ stress models are analyzed. The classification accuracy results are shown in boxplots in Fig. 4.9. For the ‘individualized’ model, the top and bottom of the boxplot indicates 75th and 25th percentiles of the classification accuracy, respectively.
Table 4.3: PCA-reduced Dimensions for the Generalized Model and Statistics of Reduced Dimensions for the 32 Individualized Models

<table>
<thead>
<tr>
<th>Generalized Model (≥ 8)</th>
<th>Individualized Model (≥ 8)</th>
<th>Generalized Model (≥ 9)</th>
<th>Individualized Model (≥ 9)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Median</td>
<td>Mean</td>
</tr>
<tr>
<td>kNN1</td>
<td>11.0</td>
<td>9.0</td>
<td>8.4</td>
</tr>
<tr>
<td>kNN2</td>
<td>11.0</td>
<td>9.0</td>
<td>8.5</td>
</tr>
<tr>
<td>kNN3</td>
<td>11.0</td>
<td>9.0</td>
<td>8.5</td>
</tr>
<tr>
<td>kNN4</td>
<td>11.0</td>
<td>9.0</td>
<td>8.0</td>
</tr>
<tr>
<td>SVM</td>
<td>11.0</td>
<td>9.0</td>
<td>6.7</td>
</tr>
</tbody>
</table>

Figure 4.8: Effect of feature selection, thresholding, and PCA on the accuracy of different classifiers.

The whiskers show the maximum and minimum values of the accuracies, without the outliers. The solid and dashed lines depict the median and mean (µ) of the classification accuracies for the ‘individualized’ model, respectively. The dotted line depicts the accuracy value for the ‘generalized’ model. Fig. 4.9a shows the classification results obtained from kNN1 (µ = 94.5%), kNN2 (µ = 93.7%), kNN3 (µ = 93.8%), kNN4 (µ = 94.2%), and SVM (µ = 86.7%) for an ECG threshold of ten and GSR+RESP+BO+BP threshold of eight. Fig. 4.9b shows the classification results obtained from kNN1 (µ = 95.8%), kNN2 (µ = 94.7%), kNN3 (µ = 94.8%), kNN4 (µ = 94.5%), and SVM (µ = 83.2%) for an ECG threshold of ten and
Figure 4.9: Accuracy statistics of the machine learning algorithms for the ‘individualized’ and ‘generalized’ models with GSR+RESP+BO+BP thresholds of (a) 8, (b) 9, and (c) definitions of various boxplot parameters.

GSR+RESP+BO+BP threshold of nine. For the ‘individualized’ model, the mean of kNN classification accuracy varies between 93.7% and 94.5% in Fig. 4.9a and 94.5% and 95.8% in Fig. 4.9b. The maximum accuracy (95.8%) is obtained when \( k = 1 \). For SVM, the mean of the classification accuracy is 86.7% in Fig. 4.9a and 83.2% in Fig. 4.9b. However, for the ‘generalized’ model, the corresponding maximum accuracies are 89.2% and 83.1% in Fig. 4.9a and 89.3% and 84.6% in Fig. 4.9b for kNN and SVM, respectively. Except for the result for SVM in Fig. 4.9b, the ‘individualized’ model is observed to detect stress with higher accuracy relative to the ‘generalized’ model. Considering the negative effect of choosing a GSR+RESP+BO+BP threshold
of nine in Fig. 4.8, this result is expected. In addition to boxplots in Fig. 4.9, we also compute the 95% confidence interval (CI) for classification accuracies of the ‘individualized’ models. With an ECG threshold of ten and GSR+RESP+BO+BP threshold of eight, the corresponding 95% CIs are (92.5%, 96.5%) for kNN1, (91.8%, 95.6%) for kNN2, (91.8%, 95.9%) for kNN3, (92.0%, 96.5%) for kNN4, and (83.6%, 89.7%) for SVM. With an ECG threshold of ten and GSR+RESP+BO+BP threshold of nine, the corresponding 95% CIs are (94.0%, 97.5%) for kNN1, (92.8%, 96.5%) for kNN2, (92.8%, 96.7%) for kNN3, (92.4%, 96.6%) for kNN4, and (79.9%, 86.5%) for SVM. Both Fig. 4.9 and CIs indicate that SoDA provides high stress detection accuracy. The accuracy difference between the two models (‘individualized’ and ‘generalized’) is due to the fact that stress impacts different individuals in slightly different ways. Thus, a model derived from a population of individuals cannot be expected to be as accurate as the model derived from just the individual.

SoDA enables stress detection in real-time. As shown in Table 4.4 on MacBook Pro with 2.5 GHz Intel Core i7 processor, the stress detection stage requires approximately 0.3s for computing the feature values from WMS data. Since WMS data are continuously collected, this enables the system to provide real-time stress tracking.

4.3.2 Stress Alleviation Protocol and Order of Stress Reduction Techniques

After stress is detected, a set of features is selected and their values are traced to evaluate if the stress alleviation technique is having the desired effect. The features that were found to be the most reliable and robust to biological differences at this stage were: R-R interval, heart rate (HR), and ratio of low frequency to high frequency band power (LF/HF) of the ECG signal. Hence, they were chosen for the ‘generalized’ model. However, for the ‘individualized’ model, the system is designed to be more flexible. It allows the choice of additional features to respond to the needs of the
Table 4.4: CPU Time for Feature Extraction

<table>
<thead>
<tr>
<th>Sensor</th>
<th>#Features</th>
<th>Feature Extraction (≥ 8)</th>
<th>Feature Extraction (≥ 9)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECG</td>
<td>1</td>
<td>1</td>
<td>1.5</td>
</tr>
<tr>
<td>GSR</td>
<td>5</td>
<td>2</td>
<td>11.4</td>
</tr>
<tr>
<td>RESP</td>
<td>3</td>
<td>1</td>
<td>46.1</td>
</tr>
<tr>
<td>BO</td>
<td>1</td>
<td>1</td>
<td>35.3</td>
</tr>
<tr>
<td>BP</td>
<td>5</td>
<td>5</td>
<td>205.2</td>
</tr>
<tr>
<td>Total</td>
<td>15</td>
<td>10</td>
<td>299.5</td>
</tr>
</tbody>
</table>

user more effectively. After the application of stress therapy, if SoDA finds the traced values of the selected features indicate recovery from stress, the therapy is continued. Otherwise, the next stress reduction technique is suggested to the user.

Table 4.5 and Table 4.6 show various statistics derived from the physiological signals for the ‘generalized’ model. For the three selected features, the statistics are derived over the 0-50 s (Table 4.5) and 60-120 s (Table 4.6) intervals. In the case of tasks T1-T4, no stress therapy is employed for the duration (120 s) of the stressful task. However, in the case of tasks T5-T8, starting from the 50th second, stress alleviation technique is used, however, without removing the stressor from the environment. Since stress mitigation needs some time to have an impact, the feature values are calculated 10 s after the start of the therapy. First, in order to compare the effectiveness of the stress mitigation techniques, feature values are calculated for the 0-50 s duration for both without and with therapy cases (Table 4.5). Since the therapy does not start until the 50th second, these values can be expected to be approximately equal for the same tasks. In other words, the same tasks should stress the participant to the same degree. Fly sound, IAPS, and ice test were verified to satisfy this condition. However, when the physiological signals were analyzed for task T1 (memory game), it was observed that the participants had an excessive stress response. This may have been because memory game was the first task the participants carried out. Even though the participants wore the sensors for some
Table 4.5: Statistics of Physiological Signals in the Generalized Model for 0-50 sec.

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Median</th>
<th>25th</th>
<th>75th</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T2</td>
<td>T5</td>
<td>T2</td>
<td>T5</td>
</tr>
<tr>
<td>R-R</td>
<td>0.52</td>
<td>0.51</td>
<td>0.51</td>
<td>0.48</td>
</tr>
<tr>
<td>HR</td>
<td>0.18</td>
<td>0.19</td>
<td>0.18</td>
<td>0.18</td>
</tr>
<tr>
<td>LF/HF</td>
<td>0.06</td>
<td>0.06</td>
<td>0.03</td>
<td>0.03</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>T3</th>
<th>T7</th>
<th>T3</th>
<th>T7</th>
<th>T3</th>
<th>T7</th>
<th>T3</th>
<th>T7</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-R</td>
<td>0.54</td>
<td>0.52</td>
<td>0.53</td>
<td>0.51</td>
<td>0.42</td>
<td>0.44</td>
<td>0.64</td>
<td>0.62</td>
</tr>
<tr>
<td>HR</td>
<td>0.17</td>
<td>0.18</td>
<td>0.16</td>
<td>0.17</td>
<td>0.11</td>
<td>0.12</td>
<td>0.22</td>
<td>0.21</td>
</tr>
<tr>
<td>LF/HF</td>
<td>0.06</td>
<td>0.10</td>
<td>0.02</td>
<td>0.03</td>
<td>0.01</td>
<td>0.01</td>
<td>0.06</td>
<td>0.06</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>T4</th>
<th>T8</th>
<th>T4</th>
<th>T8</th>
<th>T4</th>
<th>T8</th>
<th>T4</th>
<th>T8</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-R</td>
<td>0.51</td>
<td>0.55</td>
<td>0.49</td>
<td>0.49</td>
<td>0.41</td>
<td>0.42</td>
<td>0.61</td>
<td>0.58</td>
</tr>
<tr>
<td>HR</td>
<td>0.19</td>
<td>0.18</td>
<td>0.18</td>
<td>0.18</td>
<td>0.13</td>
<td>0.14</td>
<td>0.23</td>
<td>0.23</td>
</tr>
<tr>
<td>LF/HF</td>
<td>0.05</td>
<td>0.07</td>
<td>0.02</td>
<td>0.03</td>
<td>0.01</td>
<td>0.01</td>
<td>0.05</td>
<td>0.07</td>
</tr>
</tbody>
</table>

Table 4.6: Statistics of Physiological Signals in the Generalized Model for 60-120 sec.

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Median</th>
<th>25th</th>
<th>75th</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T2</td>
<td>T5</td>
<td>T2</td>
<td>T5</td>
</tr>
<tr>
<td>R-R</td>
<td>0.51</td>
<td>0.55</td>
<td>0.47</td>
<td>0.53</td>
</tr>
<tr>
<td>HR</td>
<td>0.21</td>
<td>0.19</td>
<td>0.21</td>
<td>0.18</td>
</tr>
<tr>
<td>LF/HF</td>
<td>0.05</td>
<td>0.05</td>
<td>0.02</td>
<td>0.02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>T3</th>
<th>T7</th>
<th>T3</th>
<th>T7</th>
<th>T3</th>
<th>T7</th>
<th>T3</th>
<th>T7</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-R</td>
<td>0.54</td>
<td>0.55</td>
<td>0.53</td>
<td>0.55</td>
<td>0.43</td>
<td>0.46</td>
<td>0.62</td>
<td>0.64</td>
</tr>
<tr>
<td>HR</td>
<td>0.18</td>
<td>0.19</td>
<td>0.18</td>
<td>0.17</td>
<td>0.13</td>
<td>0.12</td>
<td>0.24</td>
<td>0.22</td>
</tr>
<tr>
<td>LF/HF</td>
<td>0.09</td>
<td>0.07</td>
<td>0.03</td>
<td>0.02</td>
<td>0.01</td>
<td>0.01</td>
<td>0.10</td>
<td>0.05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>T4</th>
<th>T8</th>
<th>T4</th>
<th>T8</th>
<th>T4</th>
<th>T8</th>
<th>T4</th>
<th>T8</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-R</td>
<td>0.52</td>
<td>0.55</td>
<td>0.50</td>
<td>0.51</td>
<td>0.40</td>
<td>0.43</td>
<td>0.63</td>
<td>0.68</td>
</tr>
<tr>
<td>HR</td>
<td>0.21</td>
<td>0.19</td>
<td>0.20</td>
<td>0.19</td>
<td>0.13</td>
<td>0.11</td>
<td>0.26</td>
<td>0.25</td>
</tr>
<tr>
<td>LF/HF</td>
<td>0.05</td>
<td>0.05</td>
<td>0.03</td>
<td>0.03</td>
<td>0.02</td>
<td>0.01</td>
<td>0.07</td>
<td>0.07</td>
</tr>
</tbody>
</table>

time to become comfortable with them and were given practice tests, their stress levels were different for the practice and real tests. Thus, their stress responses were different the first time they were asked to play the memory game (task T1) and the second time (task T6). Due to this reason, the effectiveness of the corresponding stress alleviation technique (classical music) could not be analyzed.
For the remaining therapies, when the stress mitigation techniques provide better results than the ‘no therapy’ case, the data are shown in non-bold form, else in bold, in Table 4.6. Increase in the R-R interval and decrease in HR and LF/HF indicate a more relaxed state. Within the same time interval, if the stress mitigation techniques bring these feature values to a more relaxed state, then the corresponding techniques can be concluded to be more effective than the ‘no therapy’ case. In general, the remaining three stress mitigation techniques were verified to be effective. Next, we analyzed how we can order them in terms of effectiveness. Let us consider the mean of the feature values. The table shows that both micro-meditation (T5) and good news (T8) are effective. In the case of warm stone therapy, the heart rate is not observed to indicate fast relief compared to ‘no therapy’ even though the other two feature values do. Hence, we rate warm stone (at least for the ‘generalized’ model) the lowest. Between micro-meditation and good news, since micro-meditation increases the R-R interval more, it is ranked higher.

A similar analysis is performed for the ‘individualized’ model. The results for one of the participants is shown in Table 4.7 and Table 4.8. As in the ‘generalized’ model case, in Table 4.7, the corresponding feature values are expected to be close to each other for 0-50 s (in columns T2-T5, T3-T7, and T4-T8). However, due to the smaller amount of data available for a sole individual in the ‘individualized’ model case, the corresponding feature values are not as close as in the case of the ‘generalized’ model. However, still, since the corresponding values are comparable, the analysis of stress reduction techniques is carried out. In the ‘individualized’ model, as the system collects more data, it models the stress characteristics of the user more precisely. The ‘individualized’ model also provides an opportunity to include more features that are suitable to the individual in question. This also means that the therapy order can now be tailored to the individual. For example, inhalation duration was an additional feature that was found to be useful for this individual. Increase in the
Table 4.7: Statistics of Physiological Signals in the Individualized Model for 0-50 sec.

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Median</th>
<th>25th</th>
<th>75th</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T2</td>
<td>T5</td>
<td>T2</td>
<td>T5</td>
</tr>
<tr>
<td>R-R</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
</tr>
<tr>
<td>HR</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
</tr>
<tr>
<td>LF/HF</td>
<td>0.00</td>
<td>0.28</td>
<td>0.00</td>
<td>0.28</td>
</tr>
<tr>
<td>Inh</td>
<td>0.34</td>
<td>0.24</td>
<td>0.34</td>
<td>0.24</td>
</tr>
<tr>
<td></td>
<td>T3</td>
<td>T7</td>
<td>T3</td>
<td>T7</td>
</tr>
<tr>
<td>R-R</td>
<td>0.98</td>
<td>1.03</td>
<td>0.98</td>
<td>1.03</td>
</tr>
<tr>
<td>HR</td>
<td>0.02</td>
<td>-0.02</td>
<td>0.02</td>
<td>-0.02</td>
</tr>
<tr>
<td>LF/HF</td>
<td>0.84</td>
<td>0.13</td>
<td>0.84</td>
<td>0.13</td>
</tr>
<tr>
<td>Inh</td>
<td>0.70</td>
<td>0.55</td>
<td>0.70</td>
<td>0.55</td>
</tr>
<tr>
<td></td>
<td>T4</td>
<td>T8</td>
<td>T4</td>
<td>T8</td>
</tr>
<tr>
<td>R-R</td>
<td>0.82</td>
<td>1.02</td>
<td>0.82</td>
<td>1.02</td>
</tr>
<tr>
<td>HR</td>
<td>0.11</td>
<td>-0.01</td>
<td>0.11</td>
<td>-0.01</td>
</tr>
<tr>
<td>LF/HF</td>
<td>0.05</td>
<td>0.16</td>
<td>0.05</td>
<td>0.16</td>
</tr>
<tr>
<td>Inh</td>
<td>0.04</td>
<td>0.74</td>
<td>0.04</td>
<td>0.74</td>
</tr>
</tbody>
</table>

Inhalation duration indicates reduced stress. Based on the bold values in Table 4.8, the best order for this individual can be concluded to be: warm stone (T7), good news (T8), and micro-meditation (T5). As shown in Table 4.9, since the order is different from the ‘generalized’ case, the advantage of personalized stress therapy is evident. In other words, if the user selects the ‘individualized’ model, the recovery becomes faster than if the ‘generalized’ model is selected. However, as can be seen from both Tables 4.6 and 4.8, either of the options provides faster relief than the ‘no therapy’ alternative.

4.4 Chapter Summary

In this chapter, we introduced SoDA, an automatic stress detection and alleviation system. SoDA modeled the stress characteristics of the user with the help of four different stressors. With a distinctive feature selection performed based on WMS
Table 4.8: Statistics of Physiological Signals in the Individualized Model for 60-120 sec.

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Median</th>
<th>25th</th>
<th>75th</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T2</td>
<td>T5</td>
<td>T2</td>
<td>T5</td>
</tr>
<tr>
<td>R-R</td>
<td>0.93</td>
<td>0.98</td>
<td>0.93</td>
<td>0.98</td>
</tr>
<tr>
<td>HR</td>
<td>0.04</td>
<td>0.01</td>
<td>0.04</td>
<td>0.01</td>
</tr>
<tr>
<td>LF/HF</td>
<td>0.12</td>
<td>0.31</td>
<td>0.12</td>
<td>0.31</td>
</tr>
<tr>
<td>Inh</td>
<td>0.55</td>
<td>0.28</td>
<td>0.55</td>
<td>0.28</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Median</th>
<th>25th</th>
<th>75th</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T3</td>
<td>T7</td>
<td>T3</td>
<td>T7</td>
</tr>
<tr>
<td>R-R</td>
<td>0.97</td>
<td>1.21</td>
<td>0.97</td>
<td>1.21</td>
</tr>
<tr>
<td>HR</td>
<td>0.02</td>
<td>-0.11</td>
<td>0.02</td>
<td>-0.11</td>
</tr>
<tr>
<td>LF/HF</td>
<td>0.19</td>
<td>0.06</td>
<td>0.19</td>
<td>0.06</td>
</tr>
<tr>
<td>Inh</td>
<td>0.52</td>
<td>1.19</td>
<td>0.52</td>
<td>1.19</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Median</th>
<th>25th</th>
<th>75th</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T4</td>
<td>T8</td>
<td>T4</td>
<td>T8</td>
</tr>
<tr>
<td>R-R</td>
<td>0.93</td>
<td>1.01</td>
<td>0.93</td>
<td>1.01</td>
</tr>
<tr>
<td>HR</td>
<td>0.04</td>
<td>0.00</td>
<td>0.04</td>
<td>0.00</td>
</tr>
<tr>
<td>LF/HF</td>
<td>0.10</td>
<td>0.14</td>
<td>0.10</td>
<td>0.14</td>
</tr>
<tr>
<td>Inh</td>
<td>0.27</td>
<td>0.71</td>
<td>0.27</td>
<td>0.71</td>
</tr>
</tbody>
</table>

Table 4.9: Order of Stress Reduction Techniques

<table>
<thead>
<tr>
<th>Generalized Model</th>
<th>Individualized Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Micro-meditation</td>
<td>Warm Stone</td>
</tr>
<tr>
<td>2 Good News</td>
<td>Good News</td>
</tr>
<tr>
<td>3 Warm Stone</td>
<td>Micro-meditation</td>
</tr>
</tbody>
</table>

data, when the individual is subjected to these stressors, high accuracy in stress detection is obtained. All four stressors are evaluated with and without the stress alleviation techniques. Their impact is evaluated after the participants stress level attains approximately the same level as just before the application of stress alleviation. While performing the task, the stressor is not removed in the alleviation stage. This enables more realistic and reliable comparisons of the stress alleviation techniques. Moreover, since different individuals may respond in different degrees to various stress alleviation techniques, the system responds to users needs adaptively and quickly by selecting the best sequence of such techniques.
Chapter 5

YSUY: Your Smartphone Understands You – Using Machine Learning to Address Fundamental Human Needs

In this chapter, we present machine learning models that can be used to understand us. We call this system YSUY. YSUY uses wearable medical sensors and understands our physical, mental, and 4-class (2-class) emotional states with 90.0%, 90.3%, and 98.4% (99.5%) accuracy, respectively. We show that YSUY is a promising candidate for adapting ML models to human-centric needs [219].

5.1 Introduction

Artificial intelligence (AI) technologies have begun to have an impact on a wide range of application areas, such as natural language processing, healthcare, transportation, entertainment, education, safety, and security, thus enhancing the quality of life of
their users. Owing to their success in being able to tackle complex tasks, these technologies are experiencing rapid growth. According to a report by McKinsey & Co., AI is expected to have a $3.5T to $5.8T impact on the global economy across 19 industries and nine business functions [220]. As AI makes greater inroads into our day-to-day lives, it becomes important to evaluate human-AI interactions. These interactions are expected to become stronger and more personalized [221].

Currently, it is up to the users to assess their human needs and think of how AI technologies can be used to address those needs. This places a burden on the users to assess the needs in an accurate and timely fashion. According to Garry Kasparov, the chess grandmaster, humans spend approximately 99% of their resources on understanding and the remaining on computation; however, AI technologies do the opposite: they utilize 99% of their resources on computation and the remaining on understanding [222]. While these percentages can be debated, the complementary emphasis on understanding and computation seems to be true. Due to this difference, current AI technologies need to depend on the users for decision-making. Once the users make a decision regarding which task should be executed next, the corresponding AI technologies execute it with high accuracy and efficiency. Since AI technologies do not focus on understanding the users, they require the users to assess the situation and their needs. This leads to a gap between the users and the technology; therefore, it weakens the human-AI interaction.

In order to close this gap and enhance the incorporation and effectiveness of AI technologies, we propose an ML-based system: YSUY. YSUY uses wearable medical sensors (WMSs), typically present in small form-factor devices, such as a smartwatch or smartphone, to gauge the physical, mental, and emotional states of the users. After performing an extensive analysis of YSUY, we provide a detailed discussion of how it can be used to begin to address various human needs. Max-Neef presents an elegant
36-cell matrix that delineates nine human needs in four dimensions \[^{223}\]. We discuss how these cells can be addressed by YSUY.

The main contributions of this chapter are as follows:

1. We present an ML-based system, YSUY, that attempts to understand the physical, mental, and emotional states of the users and their needs. YSUY serves as a bridge between the human and the AI technology.

2. We advocate the need for understanding to flow from AI technologies, such as ML models, towards human needs so that these needs can be more efficiently addressed.

3. We built a smartphone app for YSUY and perform experiments with it in real-life situations, without limiting the users to specific experimental protocols.

4. We provide detailed classification analyses of the physical, mental, and emotional states and discuss the role YSUY can play by providing the state information in the context of four dimensions (i.e., ‘being,’ ‘doing,’ ‘having,’ and ‘interacting.’) of Max-Neef’s fundamental human needs matrix.

The remainder of this chapter is organized as follows. Section \[^{5.2}\] introduces the methodologies for data collection, experimental procedure, data processing, feature extraction, ML hyperparameter tuning, and decision making. Section \[^{5.3}\] provides experimental results and a discussion on physical, mental, and emotional state detection. Section \[^{5.4}\] analyzes the proposed system from the fundamental human needs perspective. Finally, Section \[^{5.5}\] concludes the chapter with a brief summary.

### 5.2 Methodology

In this section, we describe the data collection, experimental flow, data processing, and feature extraction phases of YSUY.
Table 5.1: Sensors, Their Abbreviations, Types, and Sampling Rates

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Abbreviation</th>
<th>Type (WMS/Smartphone)</th>
<th>Sampling rate (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Respiration</td>
<td>RESP</td>
<td>WMS</td>
<td>10</td>
</tr>
<tr>
<td>Galvanic skin response</td>
<td>GSR</td>
<td>WMS</td>
<td>10</td>
</tr>
<tr>
<td>Blood volume pulse</td>
<td>BVP</td>
<td>WMS</td>
<td>64</td>
</tr>
<tr>
<td>Temperature</td>
<td>TEMP</td>
<td>WMS</td>
<td>4</td>
</tr>
<tr>
<td>Accelerometer</td>
<td>ACC-w</td>
<td>WMS</td>
<td>32</td>
</tr>
<tr>
<td>Accelerometer</td>
<td>ACC-s</td>
<td>Smartphone</td>
<td>4-5</td>
</tr>
<tr>
<td>Gyroscope</td>
<td>Gyr</td>
<td>Smartphone</td>
<td>4-5</td>
</tr>
<tr>
<td>Blood pressure</td>
<td>BP</td>
<td>WMS</td>
<td>-</td>
</tr>
<tr>
<td>Blood oximeter</td>
<td>BO</td>
<td>WMS</td>
<td>-</td>
</tr>
</tbody>
</table>

5.2.1 Sensors and Data Collection

YSUY bases its analyses of human states on data obtained from various angles by utilizing multiple sensors. It collects physiological [respiration, Galvanic skin response (measures the electrical activity of the skin), blood volume pulse (measures the cardiovascular activity), skin temperature, systolic and diastolic blood pressure, and blood oxygen level] and electromechanical (three-axis accelerometer and three-axis gyroscope) signals through WMSs \[224, 225, 226\] and a smartphone \[227\]. Table 5.1 shows the physiological and electromechanical sensors, their abbreviations, whether they are based on WMS or smartphone, and their sampling rates. The sampling rate of smartphone-based data (accelerometer and gyroscope sensors) is in between 4 Hz and 5 Hz. An Android API for the smartphone has a delay of 200 ms (corresponding to a maximum sampling rate of 5 Hz) and the status of the smartphone affects this delay. Therefore, the sampling rate of the smartphone based data varies between 4
Hz and 5 Hz. In order not to distract the participant, we only collect blood pressure and blood oxygen levels once or twice in each session.

We collected data from seven participants (two females and five males: ages between 21 and 27) for the physical and mental state experiments and eleven participants (four females and seven males: ages 18 to 27) for the emotional state experiments. The experimental flow, content, smartphone app, and the consent form were approved by the Institutional Review Board of Princeton University. Before collecting the data, all participants were clearly informed about the experiment and they all signed a consent form. None of the participants reported any physical or mental disorder or any kind of health condition that would require continuous treatment or use of medication. The data obtained from one of the participants were not used for emotional state analyses due to the low quality of the acquired physiological signals.

5.2.2 YSUY Smartphone Application

We developed the YSUY Android application using Android Studio from Google [228] to collect the physical state, mental state, blood pressure, and blood oxygen level information. The version of the Android software development kit (SDK) and Java versions are 21 and 1.8.0, respectively. We capture the sensor data from an embedded sensor event listener. It returns a sensor data stream at a maximum frequency of 5Hz. In addition, we obtain the users’ input (e.g., physical/mental state labels, etc.) with a click event listener and editable text boxes. We test our YSUY Android application on both Samsung Galaxy S4 smartphone and a virtual Google Pixel smartphone. Fig. 5.1 shows the user interface of the YSUY application. We explain how the participants logged in their physical and mental states next.
5.2.3 Experimental Procedure

We obtain data from the participants for the YSUY experiments for analyzing their three states (physical, mental, and emotional). The participants take part in the physical and mental state experiments while performing their daily activities in their own environments, without being limited to specific experimental protocols. For the emotional state experiments, we carry out a separate experimental flow to avoid biased results as the participant’s emotional status reports can be affected by the participant characteristics and environmental factors [229]. We explain the experimental procedures for obtaining the physical, mental, and emotional states in detail next.

Physical and Mental State Experiments

In the physical and mental state experiments, the WMSs (BP, BO, GSR, RESP, and a wristband that includes BVP, TEMP, and ACC-w), smartphone, and YSUY
app (Fig. 5.1) are used. Data are collected while the participants experience daily life and fulfill their responsibilities in real-life situations. We aim at capturing the natural response of the participant and minimizing distractions by not specifying the frequency and duration of the session. Also, we do not ask the participants to undergo all possible physical and mental state options presented in the YSUY menu (Fig. 5.1a and Fig. 5.1c). We carry out the physical and mental state experiments with the following steps:

1. The participants wear the wristband. The wristband continuously collects data without requiring any intervention.

2. The participants describe their physical and mental states through the YSUY app. From this app, the participants select either one (thirsty, hungry, etc.) or multiple states (sleepy-thirsty, sleepy-full, etc.) from the given options (Fig. 5.1a and Fig. 5.1c).

3. The participants submit the blood pressure [systolic (sys) and diastolic (dia)] and blood oxygen level information at that time instant (Fig. 5.1b). Data collection starts.

4. Data collection continues until the participants stop the session through the YSUY app (‘Stop Collection’ button shown in Fig. 5.1c).

During the experiments, the participants provide their physical and mental state information. Although the total number of data logins for the two states are equal, the total number of different states might not be the same. For example, in the first data collection session, let us assume that the participant is in the ‘walking’ physical state and ‘hungry’ mental state, and in the second session, the participant is in the ‘running’ and ‘hungry’ physical and mental states, respectively. Overall, the participant has two data logins, but two different physical state reports and one mental state report.
Emotional State Experiments

In the emotional state experiments, we follow an experimental procedure that avoids biased reports. While distinguishing between the ‘walking’ and ‘sleeping’ physical states is straightforward for the participants, the emotional status reports can be affected by their characteristics and environmental factors [229]. In order to get around this problem, we stimulate the participants with pictures from the International Affective Picture System (IAPS) database [204]. This database includes pictures and their corresponding arousal-valence ratings that are validated through detailed experiments and analyses. As shown in the two-dimensional emotion chart (Fig. 5.2), each arousal and valence combination points to a particular emotion [230]. In order not to distract the participant and avoid situations that may affect the reporting of emotions, for each IAPS picture, we do not ask the users about their emotions, but use the arousal-valence ratings associated with the picture. We focus on four regions in the emotion chart: high arousal-high valence, low arousal-high valence, low arousal-low valence, and high arousal-low valence, corresponding to Quadrants I, II, III, and IV in Fig. 5.2 respectively. Moreover, using the ratings for females and males in the
<table>
<thead>
<tr>
<th>Emotional State</th>
<th>Picture Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>High Arousal - High Valence</strong></td>
<td>1340, 1463, 1710, 1722, 1811, 1999, 2045, 2050, 2058, 2071, 2075, 2150, 2155, 2158, 2160, 2165, 2208, 2209, 2216, 2224, 2300, 2344, 2345, 2347, 2352, 2352.1, 2550, 4532, 4542, 4572, 4575, 4599, 4610, 4614, 4623, 4624, 4626, 4628, 4640, 4641, 5260, 5270, 5460, 5470, 5480, 5623, 5700, 5825, 5833, 5910, 7230, 7260, 7270, 7282, 7330, 7400, 7405, 7502, 8033, 8041, 8080, 8090, 8162, 8170, 8190, 8200, 8210, 8350, 8380, 8420, 8460, 8470, 8496, 8499, 8500, 8501, 8502, 8503, 8531, 8540</td>
</tr>
<tr>
<td><strong>Low Arousal - High Valence</strong></td>
<td>2095, 2345.1, 2375.1, 2703, 2751, 2800, 2900, 3001, 3005.1, 3015, 3016, 3053, 3059, 3061, 3062, 3063, 3064, 3068, 3069, 3100, 3101, 3102, 3103, 3110, 3130, 3131, 3140, 3168, 3170, 3180, 3181, 3191, 3195, 3225, 3230, 3261, 3266, 3301, 3350, 3350, 3530, 6021, 6022, 6315, 6415, 6563, 9040, 9075, 9140, 9181, 9183, 9185, 9187, 9252, 9253, 9254, 9300, 9301, 9302, 9322, 9325, 9326, 9332, 9405, 9410, 9412, 9420, 9421, 9432, 9433, 9435, 9570, 9571, 9635.1, 9800, 9902, 9903, 9910, 9911, 9921, 9940</td>
</tr>
<tr>
<td><strong>Low Arousal - Low Valence</strong></td>
<td>1505, 2205, 2278, 2301, 2312, 2399, 2455, 2490, 2525, 2590, 2682, 2695, 2700, 2715, 2716, 2718, 2722, 2750, 2752, 2753, 2795, 2900.1, 3190, 3300, 4001, 4142, 4210, 4230, 4235, 4290, 4300, 4302, 4635, 6000, 6010, 6241, 6800, 7023, 7079, 7092, 7136, 7137, 7520, 7521, 9000, 9001, 9002, 9008, 9041, 9045, 9046, 9080, 9101, 9171, 9180, 9190, 9220, 9265, 9270, 9280, 9290, 9291, 9330, 9331, 9341, 9342, 9390, 9395, 9404, 9440, 9445, 9469, 9471, 9635.2, 9830, 9831, 9832, 9912, 9913, 9926</td>
</tr>
<tr>
<td><strong>High Arousal - Low Valence</strong></td>
<td>1440, 1441, 1460, 1500, 1540, 1600, 1620, 1630, 1721, 1731, 1750, 1920, 2035, 2040, 2057, 2070, 2080, 2091, 2151, 2154, 2156, 2170, 2222, 2250, 2260, 2274, 2299, 2304, 2306, 2310, 2311, 2314, 2331, 2332, 2340, 2341, 2360, 2387, 2388, 2395, 2398, 2530, 2540, 2598, 2630, 2650, 2660, 4574, 4616, 4622, 5001, 5199, 5201, 5202, 5210, 5551, 5594, 5600, 5660, 5760, 5779, 5780, 5811, 5820, 5829, 5830, 5831, 5836, 5982, 7200, 7280, 7325, 7470, 7492, 7570, 7580, 8032, 8120, 8461, 8497</td>
</tr>
</tbody>
</table>

IAPS report [204], we select 80 pictures for each quadrant, as shown in Table 5.2 and Table 5.3. Then, to take the stochastic nature of real-life situations into account, we randomly choose 20 pictures from each quadrant for each participant. The flow of the emotional state experiment (Fig. 5.3) is as follows:
### Table 5.3: IAPS Database Picture Numbers for Male Participants

<table>
<thead>
<tr>
<th>Emotional State</th>
<th>Picture Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Arousal - High Valence</td>
<td>1710, 1811, 2030, 2152, 2160, 2209, 2216, 2340, 2346, 2391, 4006, 4008, 4130, 4150, 4250, 4255, 4275, 4279, 4310, 4320, 4599, 4601, 4607, 4608, 4626, 4641, 4650, 4652, 4653, 4660, 4670, 4680, 4690, 5260, 5270, 5450, 5460, 5470, 5480, 5490, 5600, 5623, 5660, 5700, 5825, 5833, 5910, 5982, 7230, 7270, 7350, 7405, 7460, 7480, 7492, 7502, 7508, 7580, 8080, 8116, 8120, 8170, 8180, 8190, 8210, 8300, 8340, 8370, 8371, 8380, 8420, 8470, 8499, 8501, 8502, 8503, 8510, 8531</td>
</tr>
<tr>
<td>Low Arousal - High Valence</td>
<td>2345.1, 2352.2, 2703, 3001, 3005.1, 3015, 3016, 3030, 3051, 3053, 3059, 3060, 3061, 3062, 3063, 3064, 3069, 3071, 3080, 3100, 3101, 3102, 3103, 3110, 3120, 3130, 3131, 3140, 3150, 3160, 3168, 3170, 3180, 3191, 3215, 3220, 3225, 3261, 3266, 3350, 3530, 6021, 6022, 6212, 6360, 6520, 6560, 6570, 7380, 9006, 9040, 9075, 9180, 9183, 9252, 9253, 9254, 9322, 9325, 9326, 9405, 9410, 9412, 9413, 9414, 9419, 9428, 9520, 9560, 9570, 9635.1, 9800, 9810, 9901, 9903, 9904, 9910, 9911, 9920, 9921</td>
</tr>
<tr>
<td>Low Arousal - Low Valence</td>
<td>1111, 1275, 2053, 2055.1, 2095, 2120, 2141, 2205, 2276, 2278, 2301, 2375.1, 2455, 2456, 2457, 2590, 2700, 2750, 2751, 2799, 2800, 2900, 2900.1, 3017, 3181, 3300, 3301, 4621, 6311, 6561, 7079, 7135, 7361, 9000, 9002, 9007, 9010, 9031, 9041, 9043, 9090, 9102, 9140, 9145, 9181, 9182, 9185, 9220, 9265, 9280, 9290, 9295, 9301, 9302, 9320, 9330, 9331, 9340, 9341, 9342, 9395, 9415, 9417, 9421, 9430, 9432, 9435, 9452, 9470, 9471, 9530, 9561, 9571, 9584, 9596, 9610, 9830, 9831, 9832, 9912</td>
</tr>
<tr>
<td>High Arousal - Low Valence</td>
<td>1410, 1440, 1441, 1460, 1463, 1500, 1510, 1540, 1600, 1660, 1721, 1722, 1731, 1740, 1750, 1920, 1999, 2040, 2045, 2050, 2057, 2058, 2070, 2071, 2080, 2091, 2150, 2153, 2154, 2158, 2170, 2208, 2224, 2260, 2306, 2311, 2332, 2345, 2347, 2352, 2373, 2530, 2540, 2550, 2650, 2660, 4603, 5210, 5220, 5300, 5594, 5611, 5631, 5725, 5760, 5780, 5781, 5814, 5820, 5829, 5830, 5831, 5836, 7200, 7250, 7260, 7280, 7330, 7390, 7400, 7410, 7430, 7450, 7470, 7505, 7530, 8350, 8461, 8540</td>
</tr>
</tbody>
</table>

1. We start the experiment by welcoming the participant, explaining the experiment, and placing the WMSs (BP, BO, GSR, RESP, TEMP, and ACC-w).

2. We display 20 pictures from each quadrant of the emotion chart. We show each IAPS picture for 30 s, leading to a 10-min session duration, as shown in red color in Fig. 5.3.
3. Since the pictures from different quadrants stimulate different emotions in the participants, in line with the experimental procedure described in [33], we give breaks in between the sessions to enable the participant to recover. During the breaks, we ask the participant to close their eyes and relax.

4. We end the experiment by taking out the WMSs and thanking the participant.

5.2.4 Data Processing and Feature Extraction

The data processing and feature extraction stages depend on the available data. Hence, the set of sensors used in the physical and mental state experiments is different than the set used for the emotional state experiments. We explain how YSU Y processes the collected data for the two sets of experiments next.

Physical and Mental States

Fig. 5.4 shows the data processing and decision making stages for physical and mental state experiments. The overall process includes ‘Data alignment,’ ‘Data windowing,’ ‘Feature Extraction,’ ‘Data normalization,’ ‘Hyperparameter tuning,’ ‘ML training,’ and ‘Inference’ stages.

Data alignment: YSU Y obtains data from two sources: WMSs and smartphone. The participant starts and ends data collection through the YSU Y app while wearing the sensors. WMSs collect data continuously; however, inputs from the participant through the YSU Y app determine the start time, end time, and labels (physical state and mental state) in the corresponding epoch. As the data are obtained from both the
Figure 5.4: The YSU Y data processing and decision-making stages for physical and mental state experiments. Subscripts SP and WMS stand for smartphone and wearable medical sensors, respectively.
WMSs and smartphone, they need to be aligned in time. We perform data alignment before further data processing and feature extraction, as shown in Fig. 5.4. The data alignment stage targets the available WMS and smartphone data. It finds the start time \((t_{\text{start}})\) and end time \((t_{\text{end}})\) that lead to the largest common time span. As the WMSs collect data continuously and the participant starts and ends the epoch, Case 1 in Fig. 5.4 refers to the expected circumstance in terms of signal duration. In Case 1, smartphone data collection starts and ends earlier than that of WMSs. However, we encounter situations where the participant forgets to turn on the WMSs (Case 2 and Case 3), WMSs lose their connection, or their battery dies (Case 3 and Case 4). After performing data alignment, \(t_{\text{start}}\) and \(t_{\text{end}}\) are provided as input to the next stage.

**Data windowing:** Both data sources (WMSs and smartphone) generate time series data. Each data instance has temporal correlation with the former and subsequent instances. Taking this relationship into account, we divide the collected data into 20 s windows with 5 s shifts in between. We call this stage data windowing. As shown in Fig. 5.4, data windowing is preceded by data length checking. If the data length \((t_{\text{end}} - t_{\text{start}})\) is shorter than the window size \((d_{\text{window}})\), we discard the data as they have inadequate information. Once the data are filtered through the data length checking stage, they are divided into windows using \(t_{\text{start}}\) and \(t_{\text{end}}\) information.

**Feature extraction:** The collected data in the physical and mental state experiments include three-axis accelerometer and three-axis GYR electromechanical signals, and GSR, BVP, TEMP, BO, and BP physiological signals. Since the participant submits the measured values of the blood oxygen level and blood pressure once per session, we directly use these values as features. For the remaining data, we calculate both linear and nonlinear features as WMSs and smartphones exhibit linear and nonlinear characteristics that need to be captured in the feature extraction
Figure 5.5: The YSUY feature extraction stages for smartphone and WMS data in the physical and mental state experiments.

A previous study [231] demonstrates the effectiveness of using the Wavelet and Fourier transform features for human activity recognition. In line with this study, as shown in Fig. 5.5 in the first step of feature extraction, we calculate the 4th-order Daubechies Wavelet transform (for both the approximation and detail coefficients) and Fast Fourier transform of the data in each window of wearable and smartphone based sensors. For the Wavelet transform, we apply single-level decomposition to each data window. Since YSUY also obtains frequency-based information from the Fourier transform, application of single-level decomposition leads to satisfactory results for human state recognition.

From each transform and the original data, we compute the mean, median, standard deviation (Std. dev.), maximum, minimum, and the difference between maximum and minimum values. The original signal and its Fourier transform each leads to six features. On the other hand, the Wavelet transform leads to 12 features: six for approximation coefficients and six for detail coefficients. Therefore, the feature extraction block outputs 24 features extracted from a physiological/electromechanical signal. After obtaining the set of linear and nonlinear features, we proceed to the data normalization stage.

Data normalization: The data normalization stage ensures that the feature values are in the same range, thus avoiding dominance by one feature over another due to the difference in their range. This stage rescales the feature values using Eq. (5.1) to the $[-1, 1]$ range. In Eq. (5.1), the maximum and minimum values are...
calculated using training data \((d)\) and applied to both both training and test data instances \((d_i)\), leading to normalized data instances \((d'_i)\).

\[
d'_i = 2 \times \frac{d_i - \min(d)}{\max(d) - \min(d)} - 1 \tag{5.1}
\]

**Hyperparameter tuning:** The hyperparameter tuning stage takes normalized training data and corresponding labels as input and outputs the best set of hyperparameter values that maximize the classification performance. We implement this stage using the framework introduced in [232]. It uses Bayesian optimization with Gaussian process (GP). It starts with a prior belief over the possible hyperparameter values and their classification performance. In each iteration, it updates GP using the classification performance and chosen hyperparameter values. With the help of an upper confidence bound as the acquisition function, the next set of hyperparameter values is determined. Then, the same loop is executed to obtain classification performance and GP update. This process continues until the maximum number of iterations is reached.

**ML training and inference:** After obtaining the best set of hyperparameter values, YSUUY trains the ML algorithm with the normalized training data and labels. This stage yields ML models for the physical and mental states of the users. YSUUY then proceeds to the inference stage, where it obtains the state information corresponding to an unlabeled data instance.

**Emotional State**

Emotional state experiments do not depend on participant reports to avoid unintentional biases introduced due to environmental factors and participant characteristics. Thus, the participants do not use the smartphone to log their emotional state information. Instead, the participants are shown a set of IAPS pictures, whose arousal-valence ratings and, hence, the induced emotions are known. Moreover, we use the respira-
Figure 5.6: The YSUY data processing and decision-making stages for the emotional state experiment.

Data windowing: Data processing of the emotional state experiments starts with windowing of the WMS data. In line with the physical and mental state experiments, collected data are divided into 20 s windows with 5 s shifts in between. Although consecutive windows have overlaps, none of the windows corresponding to the training set overlaps with the ones in the test set. We discard the test set windows that overlap with those of the training set, when dividing the data into training and test sets.

Feature extraction: Fig. 5.7 shows the feature extraction procedure and the total number of features for the emotional state data. We extract the 24 linear and nonlinear features described in Section 5.2.4 from the GSR, BVP, TEMP, and three-axis ACC-w signals. We extract three features (mean, median, and standard deviation) from blood oxygen level and systolic and diastolic blood pressure for each measurement, resulting in nine features per window. For RESP, we use both the signal and its peaks and valleys that represent inhalation and exhalation, respectively. Fig. 5.8 shows the peak and valley information on the RESP signal. The RESP signal, its Fourier transform, and Wavelet transform lead to 24 features. The total number of peaks-valleys, and mean, median, and standard deviation of the index and magnitude of the peaks-valleys account for 14 features.
Data normalization, hyperparameter tuning, ML training, and inference: We perform the same data normalization, hyperparameter tuning, ML training, and inference, as described in Section 5.2.4 to assess the emotional state of the user.

5.3 Experimental Results

Next, we present and discuss experimental results for the physical, mental, and emotional state detection modules of YSU-Y.
Table 5.4: Declared Physical and Mental States for Each Participant

<table>
<thead>
<tr>
<th>Participant #</th>
<th>Declared Physical States</th>
<th>Declared Mental States</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Driving, Stationary, Typing, Walking</td>
<td>Full, Normal, Preoccupied, Sleepy-Full</td>
</tr>
<tr>
<td>2</td>
<td>Driving, Sleeping, Stationary, Walking</td>
<td>Full, Normal, Sleepy, Thirsty</td>
</tr>
<tr>
<td>3</td>
<td>Driving, Stationary, Walking</td>
<td>Full, Hungry, Normal, Sleepy, Tired</td>
</tr>
<tr>
<td>4</td>
<td>Sleeping, Stationary, Walking</td>
<td>Anxious, Normal, Sleepy</td>
</tr>
<tr>
<td>5</td>
<td>Stationary, Typing, Walking</td>
<td>Excited-Preoccupied, Full, Preoccupied, Sleepy-Thirsty, Sleepy-Tired-Full</td>
</tr>
<tr>
<td>6</td>
<td>Stationary, Typing, Walking</td>
<td>Full, Hungry, Normal, Sleepy, Preoccupied, Tired</td>
</tr>
<tr>
<td>7</td>
<td>Driving, Stationary, Typing, Walking</td>
<td>Bored, Normal, Preoccupied, Tired</td>
</tr>
</tbody>
</table>

5.3.1 Data Collection

Recall that we collect physical and mental state data while the participants undergo their normal daily routine. Although we tracked data collection and kept in touch with the participants to ensure data consistency and provided help whenever needed, we did not specify the physical/mental states that need to be logged through the YSUY app or the overall data collection duration. The goal was to capture participant responses and assess the YSUY state detection capability in real-life situations. Table 5.4 shows the physical and mental states declared by various participants. The participants declare three to four different physical states and three to five different mental states during data collection. Although the participants kept the WMSs and smartphone for a minimum of two days and a maximum of four days, they were not required to start and finish data collection at specific time instances. There-
fore, the total data collection duration is less than the duration that the participant kept the sensors. On the other hand, the emotional state experiments follow an experimental protocol (Fig. 5.3) based on IAPS pictures, but incorporate randomness encountered in real-life through random picture selection from each quadrant in the two-dimensional emotion chart (Fig. 5.2). IAPS pictures have associated with them validated arousal-valence ratings. Therefore, the effect of each picture on the participant and the corresponding emotion are known. This eliminates the need for the participants to report their emotional state, which may include unintentional biases. As a result, each participant has the same set of emotional state labels: high arousal-high valence, low arousal-high valence, low arousal-low valence, and high arousal-low valence.

5.3.2 Data Processing and Feature Extraction

Next, we process the data and extract features from the signals collected through the WMSs and smartphone. Since some of the participants did not provide or only partially provided their blood oxygen level and blood pressure measurements, we did not include these signals in the data processing stage. Moreover, during data login, some participants provided their physical state, but not mental state or vice versa. If physical (mental) state was not provided, we did not use the data while building the physical (mental) state detection module, but did so while building the mental (physical) state detection module.

The feature extraction stages of YSUY, Fig. 5.5 and Fig. 5.7 include time-series, Fourier transform, and Wavelet transform features. In order to assess the significance of each feature set on classification performance, we carried out analyses on the validation set with time-series (Time), Fourier transform (FFT), and Wavelet transform (DWT) features, and their combinations. Fig. 5.9a and Fig. 5.9b show the average accuracy and F1 scores of the seven participants for the physical state recognition
task, respectively. We observe that the combination of time series, Fourier transform, and Wavelet transform features results in high accuracy and F1 score values more consistently as compared with the other sets of features. The validation set analyses for the mental and emotional states shown in Fig. 5.10 and Fig. 5.11 support this observation. In Fig. 5.10 while FFT (DWT) features improve the classification performance for SVM (MLP-1 and kNN), MLP-1 and kNN (SVM) performance deteriorates. In Fig. 5.11 while Time+FFT and Time+DWT features provide comparable classification performance to Time+FFT+DWT features for MLP-1 and SVM, their performance decreases significantly for the kNN classifier. Moreover, as the study in [233] emphasizes, the choice of Fourier transform or Wavelet transform depends on the data type. Since our input data are composed of various data types (RESP, GSR, BVP, TEMP, BP, BO, accelerometer, and GYR), extracting all time-series, Fourier transform, and Wavelet transform features and then feeding them to the machine
After data processing and feature extraction stages, the total time span and number of samples for physical and mental state experiments are shown in Table 5.5 and Table 5.6, respectively. The time span for each participant is different as the physical and mental state experiments do not follow a specific experimental protocol. While undergoing normal daily routine, the participants log in their physical and mental state information through the YSUY app and start/end the data collection. Although the YSUY app enables both the physical and mental states to be recorded for the corresponding time duration, some participants only provided either the physical or mental state information for some of the time instances. This results in different total time span values for the physical and mental state experiments for the same participant. The emotional state experiments follow an experimental protocol (Fig. 5.3) to avoid the unintentional biases reported in [229]. Therefore, each participant has
Figure 5.11: Emotional state validation set: (a) accuracy and (b) F1 score with time, FFT, and DWT-based features.

Table 5.5: Total Time Span and Number of Samples for Each Participant in the Physical State Experiments

<table>
<thead>
<tr>
<th>Participant</th>
<th>Total time span (min)</th>
<th>Number of samples</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training and validation set</td>
<td>Test set</td>
</tr>
<tr>
<td>1</td>
<td>92.4</td>
<td>885</td>
</tr>
<tr>
<td>2</td>
<td>92.7</td>
<td>889</td>
</tr>
<tr>
<td>3</td>
<td>158.4</td>
<td>1520</td>
</tr>
<tr>
<td>4</td>
<td>78.7</td>
<td>754</td>
</tr>
<tr>
<td>5</td>
<td>98.3</td>
<td>942</td>
</tr>
<tr>
<td>6</td>
<td>176.9</td>
<td>1697</td>
</tr>
<tr>
<td>7</td>
<td>110.5</td>
<td>1059</td>
</tr>
</tbody>
</table>

the same time span and number of samples in the training/validation set and test set. Each experiment takes 40 min., which corresponds to 356 and 80 training/validation set and test set samples, respectively. It is important to note that the number of samples depends on the window size and the shifts in between. The reported number of samples shows the case when the window size is 20s with 5s shifts in between.
Table 5.6: Total Time Span and Number of Samples for Each Participant in the Mental State Experiments

<table>
<thead>
<tr>
<th>Participant #</th>
<th>Total time span (min)</th>
<th>Number of samples</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Training set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Test set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total set</td>
</tr>
<tr>
<td>1</td>
<td>92.4</td>
<td>885</td>
</tr>
<tr>
<td></td>
<td></td>
<td>212</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1097</td>
</tr>
<tr>
<td>2</td>
<td>92.3</td>
<td>883</td>
</tr>
<tr>
<td></td>
<td></td>
<td>212</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1095</td>
</tr>
<tr>
<td>3</td>
<td>158.4</td>
<td>1519</td>
</tr>
<tr>
<td></td>
<td></td>
<td>367</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1886</td>
</tr>
<tr>
<td>4</td>
<td>85.7</td>
<td>821</td>
</tr>
<tr>
<td></td>
<td></td>
<td>198</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1019</td>
</tr>
<tr>
<td>5</td>
<td>98.3</td>
<td>940</td>
</tr>
<tr>
<td></td>
<td></td>
<td>225</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1165</td>
</tr>
<tr>
<td>6</td>
<td>176.9</td>
<td>1696</td>
</tr>
<tr>
<td></td>
<td></td>
<td>409</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2105</td>
</tr>
<tr>
<td>7</td>
<td>130.2</td>
<td>1247</td>
</tr>
<tr>
<td></td>
<td></td>
<td>303</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1550</td>
</tr>
</tbody>
</table>

5.3.3 ML Hyperparameter Tuning and Decision Making

The signals, hence the extracted features, form a time series. Therefore, we use the first 64% of the data instances as training data, the next 16% as validation data, and the remaining nonoverlapping part as test data. Since the set of declared states and its cardinality are not uniform across the participants, we build user-specific physical and mental state classifiers based on MLP (with one hidden layer), SVM with RBF kernel, and kNN algorithms. We optimize the hyperparameters of these algorithms using the training and validation data of the corresponding participant and a Bayesian optimization framework [232]. As shown in Table 5.7, the hyperparameter for MLP is the number of neurons in the single hidden layer. $C$ and $\gamma$ are the hyperparameters of the SVM algorithm. $C$ is the regularization (penalty) parameter for the cost function and $\gamma$ is the RBF parameter for the similarity measure. Larger $\gamma$ leads to a decision boundary that is more dependent on (shaped by) individual data points. The number of neighbors is the hyperparameter of the kNN algorithm.

The Bayesian optimization stage starts with the hyperparameters as variables and the range of values they can take. In order to adapt YSUY to data from different participants, we initialize a large range of hyperparameter values. For MLP, the number of neurons in the hidden layer has a range of $[1, 300]$. For SVM, $C$ and $\gamma$ have
Table 5.7: ML Algorithms and Hyperparameters

<table>
<thead>
<tr>
<th>ML Algorithm</th>
<th>Hyperparameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP</td>
<td>n</td>
<td>Number of neurons in the hidden layer</td>
</tr>
<tr>
<td>SVM</td>
<td>C</td>
<td>Penalty parameter</td>
</tr>
<tr>
<td></td>
<td>$\gamma$</td>
<td>RBF parameter</td>
</tr>
<tr>
<td>kNN</td>
<td>k</td>
<td>Number of neighbors</td>
</tr>
</tbody>
</table>

Table 5.8: Classification Performance of the Physical State Classifiers

<table>
<thead>
<tr>
<th>Participant #</th>
<th>MLP ACC</th>
<th>MLP F1</th>
<th>SVM ACC</th>
<th>SVM F1</th>
<th>kNN ACC</th>
<th>kNN F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.717</td>
<td>0.744</td>
<td>0.726</td>
<td>0.733</td>
<td>0.906</td>
<td>0.904</td>
</tr>
<tr>
<td>2</td>
<td>0.948</td>
<td>0.875</td>
<td>0.957</td>
<td>0.915</td>
<td>0.867</td>
<td>0.814</td>
</tr>
<tr>
<td>3</td>
<td>0.884</td>
<td>0.829</td>
<td>0.871</td>
<td>0.808</td>
<td>0.648</td>
<td>0.610</td>
</tr>
<tr>
<td>4</td>
<td>0.945</td>
<td>0.884</td>
<td>0.983</td>
<td>0.987</td>
<td>0.972</td>
<td>0.979</td>
</tr>
<tr>
<td>5</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>6</td>
<td>0.875</td>
<td>0.774</td>
<td>0.890</td>
<td>0.636</td>
<td>0.930</td>
<td>0.858</td>
</tr>
<tr>
<td>7</td>
<td>0.737</td>
<td>0.716</td>
<td>0.875</td>
<td>0.829</td>
<td>0.859</td>
<td>0.810</td>
</tr>
<tr>
<td>Average</td>
<td>0.872</td>
<td>0.832</td>
<td>0.900</td>
<td>0.844</td>
<td>0.883</td>
<td>0.854</td>
</tr>
</tbody>
</table>

ranges $[0.001, 100]$ and $[0.0001, 0.1]$, respectively. For kNN, the number of neighbors has the range $[1, 50]$. Following initialization, the stages described in Section 5.2.4 are executed. The best hyperparameter values that maximize validation set performance are selected and used throughout the experiments.

Table 5.8 and Table 5.9 show the physical and mental state accuracy and F1 scores for each participant, respectively. Accuracy (ACC) is calculated by dividing the total number of true positives (TP) and true negatives (TN) with the total number of instances in the test set [i.e., TP, TN, false positives (FP), and false negatives (FN)]. The F1 score (F1) is calculated as the unweighted mean of the F1 score of each class. The F1 score is itself the harmonic mean of precision (PREC) and recall (REC). Eq. (5.2) and Eq. (5.3), respectively, show the equations for deriving ACC and F1.

\[
ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.2}
\]
Table 5.9: Classification Performance of the Mental State Classifiers

<table>
<thead>
<tr>
<th>Participant #</th>
<th>MLP</th>
<th>SVM</th>
<th>kNN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACC  F1</td>
<td>ACC  F1</td>
<td>ACC  F1</td>
</tr>
<tr>
<td>1</td>
<td>0.939 0.946</td>
<td>0.717 0.597</td>
<td>0.958 0.950</td>
</tr>
<tr>
<td>2</td>
<td>0.675 0.767</td>
<td>0.976 0.979</td>
<td>0.840 0.862</td>
</tr>
<tr>
<td>3</td>
<td>0.823 0.816</td>
<td>0.812 0.805</td>
<td>0.823 0.802</td>
</tr>
<tr>
<td>4</td>
<td>0.995 0.996</td>
<td>0.965 0.933</td>
<td>0.990 0.993</td>
</tr>
<tr>
<td>5</td>
<td>0.938 0.936</td>
<td>0.964 0.964</td>
<td>0.991 0.991</td>
</tr>
<tr>
<td>6</td>
<td>0.857 0.847</td>
<td>0.869 0.884</td>
<td>0.803 0.844</td>
</tr>
<tr>
<td>7</td>
<td>0.937 0.924</td>
<td>0.944 0.945</td>
<td>0.914 0.923</td>
</tr>
<tr>
<td>Average</td>
<td>0.881 0.890</td>
<td>0.892 0.872</td>
<td>0.903 0.909</td>
</tr>
</tbody>
</table>

\[
F_1 = \frac{\sum_{i=1}^{n_{\text{class}}} 2 \cdot \text{PRECI}_i \cdot \text{RECI}_i \left( \text{PRECI}_i + \text{RECI}_i \right)}{n_{\text{class}}} 
\]

where \( \text{PRECI} = \frac{TP}{TP + FP} \), \( \text{RECI} = \frac{TP}{TP + FN} \), and

\(n_{\text{class}}\) is the total number of classes within the dataset.

As we can see, for the physical state, the SVM algorithm has the highest average classification accuracy and F1 score. All three algorithms achieve a perfect accuracy and F1 score of 1.0 for Participant 5. For the remaining participants, MLP and SVM exhibit comparable accuracy and F1 score values except for Participant 7. kNN has a similar trend with SVM except for Participant 3 and Participant 6. For Participant 1, while kNN achieves a classification performance above 0.90, MLP and SVM obtain the lowest accuracy and F1 score values relative to other participants. This may be due to the fact that the amount of data is not sufficient for MLP and SVM algorithms for this participant to distinguish among the four states (driving, stationary, typing, and walking) accurately. Although Participant 1 has the same four declared physical states as Participant 7, data collection duration is lower. This points out the importance of the amount of training data and the choice of the ML algorithm and supports our personalized state detection approach. For the mental state as well, YSUY achieves
Figure 5.12: Links between the physical and mental states. Thicker line corresponds to more frequent physical-mental state reporting.

a classification performance around 0.90 for SVM and kNN, with kNN having a small edge over them.

Overall, YSUY determines the physical and mental states of its users with high classification performance (approximately 0.90) even though the physical and mental state data are collected while the participants go about their daily routines with no restrictions imposed on their actions and data collection duration. This bodes well for the use of YSUY in real-world situations.

Next, we analyze the relationship between the physical and mental states. Fig. 5.12 shows the links between the physical and mental state reports from the participants. Line thickness indicates the frequency of occurrence of the connections (thicker the line, more frequent the occurrence). The participants reported walking (a physical state) and normal (a mental state) in 13% of the YSUY app logs. They reported stationary-normal, typing-preoccupied, and walking-full state combinations in 7%, driving-normal and stationary-sleepy in 6%, and stationary-sleepy-full, typing-full, sleeping-sleepy, stationary-anxious, typing-sleepy, typing-normal, typing-tired, and walking-preoccupied in 4% of the logs. Each of the remaining reports accounted for 2% of the logs. Recall that the features used in these classifications are extracted from both physiological and electromechanical signals. Since participants experience
Table 5.10: Classification Performance of the Emotional State Classifiers (Four Classes: High Arousal-High Valence, Low Arousal-High Valence, Low Arousal-Low Valence, and High Arousal-Low Valence)

<table>
<thead>
<tr>
<th>Participant #</th>
<th>MLP ACC</th>
<th>MLP F1</th>
<th>SVM ACC</th>
<th>SVM F1</th>
<th>kNN ACC</th>
<th>kNN F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>9</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>10</td>
<td>0.950</td>
<td>0.949</td>
<td>0.950</td>
<td>0.949</td>
<td>0.950</td>
<td>0.949</td>
</tr>
<tr>
<td>11</td>
<td>1.000</td>
<td>1.000</td>
<td>0.963</td>
<td>0.962</td>
<td>0.925</td>
<td>0.925</td>
</tr>
<tr>
<td>12</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>13</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>14</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>0.963</td>
<td>0.962</td>
</tr>
<tr>
<td>15</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>16</td>
<td>0.888</td>
<td>0.882</td>
<td>0.913</td>
<td>0.910</td>
<td>0.925</td>
<td>0.923</td>
</tr>
<tr>
<td>17</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>Average</td>
<td>0.984</td>
<td>0.983</td>
<td>0.983</td>
<td>0.982</td>
<td>0.976</td>
<td>0.976</td>
</tr>
</tbody>
</table>

physical and mental states simultaneously and both states have an effect on physiological and electromechanical signals, there is a relationship between the physical and mental states. In Fig. 5.12, we indicate how frequently the links occur. However, the absence of a link may also carry a lot of information. For example, the walking physical state is never linked with the anxious mental state. However, the stationary physical state is linked with the anxious mental state in 4% of the logs. This might indicate that walking may be helpful in avoiding a negative mental state like anxiety. Similarly, the sleeping physical state is not linked with the bored or anxious mental states, but linked with the normal mental state. The reason for this is self-evident.

Table 5.10 shows the classification performance of YSUYY’s emotion detection module. As discussed in Section 5.2.3, YSUYY focuses on the four quadrants of the emotion chart: high arousal-high valence, low arousal-high valence, low arousal-low valence, and high arousal-low valence. The table shows that YSUYY can distinguish among these four quadrants quite accurately, with a perfect 1.00 accuracy and F1 score in many cases.
Table 5.11: Classification Performance of the Emotional State Classifiers (Two Classes: High Valence, Low Valence)

<table>
<thead>
<tr>
<th>Participant #</th>
<th>MLP ACC</th>
<th>F1</th>
<th>SVM ACC</th>
<th>F1</th>
<th>kNN ACC</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>9</td>
<td>1.000</td>
<td>1.000</td>
<td>0.950</td>
<td>0.950</td>
<td>0.988</td>
<td>0.987</td>
</tr>
<tr>
<td>10</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>11</td>
<td>0.963</td>
<td>0.962</td>
<td>1.000</td>
<td>1.000</td>
<td>0.950</td>
<td>0.950</td>
</tr>
<tr>
<td>12</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>13</td>
<td>0.988</td>
<td>0.987</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>14</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>15</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>16</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>17</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>Average</td>
<td>0.995</td>
<td>0.995</td>
<td>0.995</td>
<td>0.995</td>
<td>0.994</td>
<td>0.994</td>
</tr>
</tbody>
</table>

For applications that require classification of emotions into just two classes: positive (high valence) and negative (low valence), we present results in Table 5.11. YSUY has perfect classification performance from at least two algorithms, except for Participant 11.

5.4 Application of YSUY to Human Needs

In this section, we first explain the basic human needs and discuss how YSUY can address these needs in four different dimensions: ‘being,’ ‘doing,’ ‘having,’ and ‘interacting.’

5.4.1 Basic Human Needs

Max-Neef introduced nine basic human needs (subsistence, protection, affection, understanding, participation, leisure, creation, identity, and freedom) viewed from four different perspectives (‘being,’ ‘doing,’ ‘having,’ and ‘interacting’) and placed them in a 36-cell matrix [223]. He eschewed hierarchy and acknowledged that the needs may
be satisfied in different ways based on time, location, and situation. In other words, his human needs model is independent of cultural variations and time periods. However, the mechanisms for satisfying the corresponding needs may change over time, culture, and economical/political circumstances. This makes it a suitable model for YSU to pursue. Moreover, according to Max-Neef’s model, if one of the needs is not satisfied, deprivation occurs, placing a barrier to satisfying the remaining needs.

5.4.2 Being

The ‘being’ aspect of subsistence requires the physical and mental health needs to be satisfied. YSU addresses the physical health need by continuously tracking the physical activity of the users through physiological and electromechanical signals obtained from the WMSs and smartphone, respectively. Physical activities are known to have a positive effect on overall well-being of the users [231]. YSU can also help shed light on the mental health of the users and help discover various anomalies, e.g., eating disorders, negative feelings, reduced physical activity, mood swing, etc. [235], through the monitoring of the physical, mental, and emotional states of the users. Continuous classification of these states also reveals deviations from regular habits of the users and provides an opportunity for early diagnosis.

YSUY satisfies the protection need by broadly targeting users and various circumstances they find themselves in. As opposed to previously introduced care systems that focus on specific age groups, e.g., infant/elderly care [236], [237], [238], gender, or health condition, e.g., cancer [239], diabetes [240], hypertension [241], etc., YSU’s physical/mental/emotional state analysis is applicable to everyone. Thus, YSU could be used as a basis for a broader set of care systems in the future.

YSUY targets affection, participation, leisure, and identity through the emotional state classifier since respect, sense of humour, generosity, sensuality, receptiveness, dedication, sense of belonging, etc., require a positive attitude on part of the indi-
Experimental results presented in Section 5.3.3 show that YSUY is able to distinguish among the four quadrants of the emotion chart with MLP, SVM, and kNN algorithms exhibiting an average accuracy of 0.984, 0.983, and 0.976, respectively. These emotional state classifications can be used to activate biofeedback mechanisms to direct the users towards positive emotions.

YSUY directly targets the understanding need through continuous reporting of the physical/mental/emotional states of the users along with the time information. Continuous reporting and availability of past results can be used to automatically trigger critical thinking, resembling logic-based therapy (LBT) employed in psychiatry. In LBT, critical thinking techniques are practised to identify and show patients the irrational or contradictory aspects of their behavior [242].

The physical/mental/emotional state classifications provided by YSUY create self-awareness in the users about their abilities, skills, and responses. This targets the creation need.

The ‘being’ dimension of freedom includes autonomy, passion, self-esteem, and open-mindedness. Self-esteem is tackled through self-awareness [243], as described earlier. YSUY enhances open-mindedness by enabling the users to look at data points from different angles. In other words, YSUY provides multiple perspectives on a single data epoch. However, in order to enhance open-mindedness further, in the future, YSUY could be augmented beyond state classification to suggestions for a variety of user responses through an app. Since open-mindedness requires embracing, accepting, or cultivating different points of views [244], the updated YSUY may be able to bring users out of their comfort zone and enable them to benefit from a variety of solutions.
5.4.3 Having

The ‘having’ dimension of subsistence targets food, shelter, and work. YSUY identifies hunger through mental state analyses. It determines the mental state of the user with a classification accuracy and F1 score of 0.903 and 0.909, respectively. Moreover, the time information associated with the classification reveals the duration of hunger and its frequency, which can be used to assess the significance of the problem. YSUY can provide hints for shelter- and work-related issues with the help of physical activity, emotion tracking, and mental state analysis components. For example, an unemployed person may exhibit less physical activity or higher mental stress, which may increase blood pressure [245], depression, anxiety [246], and tendency towards an unhealthy diet [247]. Similarly, shelter problems may lead to increased alcohol usage, negative feelings, and sleep problems [248], [249]. YSUY targets the type, duration, and frequency of negative feelings, blood pressure level, alcohol usage, and eating habits through detailed physical, emotional, and mental state analyses. YSUY classification results need not just be for personal consumption. With user permission, if the collective YSUY results from a large number of individuals are shared with accredited organizations without revealing user identities, YSUY could become influential on a large scale. Collective YSUY results could be used to assess the states/regions of a country that require action to satisfy the ‘having’ dimension of subsistence, e.g., food, shelter, work, etc. This could lead to a better allocation of limited resources. The collective YSUY results may also be helpful for the protection and understanding needs since social security, health systems, work, literature, policies, and educational tools are best tackled at governmental and legislative levels. Population-level well-being results could guide related organizations on where the most resources are needed, e.g., for social security, health systems, employment, protection plans, retirement policies, educational tools, etc. Currently, the main means of assessing the well-being of the population is through surveys and questionnaires [250], [133]. While YSUY does not
eliminate the need to collect data in this fashion, it provides a continuous source of rich data that can complement surveys/questionnaires, so that a more accurate view of the state of the population can be arrived at.

In the ‘having’ dimension of affection, friendships and relationship with nature are listed. YSUY addresses this need directly by trying to understand the users and providing feedback on their physical, mental, and emotional states, as if it were their friend. However, unlike a real-world friend, YSUY is available around the clock. It may also be helpful to the users in terms of navigating the environment in the sense that the YSUY classification results may help users reason about the link between various environmental factors and their states.

YSUY targets some elements of the participation need (responsibilities and duties) through the biofeedback mechanism [251], [252].

The leisure need requires low-intensity positive emotions. These emotions reside in the low arousal-high valence quadrant. YSUY targets this need through the emotional state classifier and gives feedback to users about the quality of their leisure. Users could also be exposed to positive emotion-inducing inputs, such as good news and pictures, positive quotes, classical music, and micro-meditation [33].

YSUY targets some aspects of the creation need through the mechanism discussed in Section 5.4.2. The self-awareness created by YSUY can encourage users to participate in related activities and monitor how they respond to this participation.

Features of the identity need are language, religion, work, customs, and values. Currently, YSUY does not address these features. However, in future work, we would like to integrate user preferences into decision-making by retraining the models at predefined intervals. Moreover, we would like to provide YSUY outputs in different languages in order to enhance its cultural coverage.

In the ‘having’ dimension of freedom, equal rights is listed. YSUY does not do much in this regard.
5.4.4 Doing

YSUY targets the ‘doing’ dimension of subsistence and creation with the approach discussed in Section 5.4.3.

For protection, understanding, and identity, the physical/mental/emotional state classification YSUY provides to the users can be used to activate a biofeedback mechanism, whose effectiveness has been verified in various studies [251], [252]. Since YSUY possesses historical data on the users, the trend lines of how useful an action is can be drawn. Feedback from YSUY also provides an opportunity to satisfy the participation need of users by verifying their state and encouraging them to share opinions in real-life situations. Feedback can also be provided in the reverse direction: from the users to YSUY through the smartphone app. This feedback can be on whether the users agree with the state classification YSUY provides. This can be taken advantage of when YSUY is retrained. Thus, cooperation and expression of opinion from the users can lead to better personalization of YSUY to user needs. Moreover, for the affection need, the biofeedback mechanism triggered through emotional analysis can enable greater sharing, caring, and emotional expression.

The leisure need requires the following features to be satisfied: day-dreaming, remembering, relaxing, and having fun. These features require positive emotions with low intensities, which reside in the low arousal-high valence quadrant. YSUY can thus give feedback to the users about the quality of their leisure.

Freedom requires dissenting, choosing, running risks, and developing awareness. These can be addressed in future work by developing modules that provide separate solutions for each of the states: physical, mental, and emotional.

5.4.5 Interacting

The ‘interacting’ dimension refers to the environments (e.g., living environments, social settings, schools, families, universities, communities, private and intimate spaces
of togetherness, associations, parties, churches, neighborhoods, and workshops) corresponding to each of the fundamental human needs. YSUY targets the ‘interacting’ dimension of these needs through both the smartphone and WMSs. The smartphone provides location information through a global positioning system (GPS) sensor or assisted GPS or differential GPS [253] and smartphone/WMSs indicate the physical, mental, and emotional states of the users. Thus, the state information can be correlated with user location or environment, offering the possibility for ML applications to provide a location- and state-specific response. Moreover, the combination of state and location information can enable the suitability of the corresponding social environment to be assessed.

5.5 Chapter Summary

We described YSUY, ML-based system targeting understanding the user and his/her needs. YSUY begins by collecting data through WMSs and the smartphone. Then, it processes the data, extracts features, and carries out a classification to determine the state of the users from three different perspectives: physical, mental, and emotional. The states at a given time instance indicate whether the human need is being met. Following obtaining the physical, mental, and four-class (two-class) emotional state classification accuracy values of 90.0%, 90.3%, and 98.4% (99.5%), respectively, we discussed how YSUY can be used to address basic human needs. The results verified the potential of YSUY on bridging the gap between human need and AI.
Chapter 6

Smart, Secure, yet Energy-efficient, Internet-of-Things Sensors

In this chapter, we present simultaneously smart, secure, and energy-efficient Internet-of-Things (IoT) sensor architecture. Our proposed approach employs signal compression, machine learning inference, and cryptographic techniques on the IoT sensor node with two different data transmission scenarios: alert notification and continuous notification. Compared to traditional sense-and-transmit IoT sensor, our approach achieves $57.1 \times$ to $912.6 \times$ energy reduction [254], [255].

6.1 Introduction

Technological advancements have led to a proliferation of Internet-of-Things (IoT) targeted at various applications. IoT ranges from a small sensor that reports on the health information of the user to an array of sensors and devices that covers the whole city to regulate traffic, ensure security, and monitor weather [256]. The total number of IoT devices is expected to reach 80 billion by 2025 and generate 180 zettabytes (ZB), or $180 \times 10^{21}$ B, of data, in total [257]. Thus, ensuring data security and
reducing energy consumption required for signal processing and data transmission are prominent challenges faced by IoT designers.

The first step in an IoT application is to collect data through IoT sensors. These sensors generate raw data that need to be processed before any action can be taken. Typically, the collected data are transmitted to the base station for processing. However, often, base stations have limited processing and storage resources. In such cases, they can only carry out simple operations on the data, such as reformatting, compression/expansion, aggregation, etc. [258]. Following these operations, data are transmitted to cloud servers for further processing and decision making (i.e., to distill intelligence and, hence, impart smartness to the system). Although cloud servers have the required computational resources for signal processing and information extraction, data transmission from IoT sensors to cloud servers throws up serious design challenges, such as security concerns, insufficient energy, and limited bandwidth. To get around these obstacles, previous studies have suggested pushing data processing towards the edge of the IoT devices and implementing cryptographic techniques (i.e., encryption and hashing) on the collected data. However, although edge-side computing enables decision-making without the use of cloud resources and encryption and hashing strengthen security, the overall computational cost, and hence energy consumption, increases significantly. Therefore, simultaneously achieving smartness, security, and energy efficiency continues to be a paramount design challenge.

We propose a novel approach to addressing this challenge by employing signal compression and machine learning inference on the IoT sensor node. We use compressed-domain inference based on concepts like compressive sensing [51, 52] and compressed signal processing (CSP) [53]. Depending on the inference outcome, the IoT sensor transmits the data or provides an alert whenever necessary. Since compressed-domain inference significantly reduces the amount of data that needs to be transmitted, we obtain a very large energy bonus. This enables the sensor to also carry out encryption
and hashing to ensure data confidentiality and integrity. Our IoT sensor architecture achieves smartness through decision-making inferences, security through encryption and hashing, and energy efficiency through both compression and decision-making inferences.

The main contributions of this chapter are as follows:

1. We propose a novel IoT sensor architecture that simultaneously achieves smartness, security, and energy efficiency. It utilizes compression and decision-making inference to obtain a significant energy bonus relative to a traditional sense-and-transmit IoT sensor and uses a part of this bonus to also incorporate encryption/hashing on the sensor.

2. We demonstrate that the architecture can be operated in multiple modes that easily adapt to various design objectives.

3. We evaluate the architecture for various IoT sensors and applications. We verify the effectiveness of our approach with detailed classification performance and energy consumption analyses. We assess the amount of saved energy by comparing our results with the traditional sense-and-transmit approach.

4. We discuss future research directions for our work.

The remainder of this chapter is organized as follows. Section 6.2 discusses the methodologies for system design and energy modeling. Section 6.3 provides experimental results for six different IoT applications. Finally, Section 6.4 concludes the chapter.
6.2 Design and Analysis Methodology

In this section, we describe our smart, secure, and energy-efficient IoT sensor design. Then, we explain the energy model that we use to analyze the proposed sensor architecture.

6.2.1 System Design

As described earlier, smartness, security, and energy efficiency are vital objectives of IoT sensor design. Previous studies typically target these objectives individually due to the trade-offs involved in navigating between security and energy efficiency or smartness and energy efficiency. All three objectives have not been targeted simultaneously previously.

We introduce a new IoT sensor architecture to address the above challenge. Fig. 6.1a and 6.1b show the architectures for the traditional sense-and-transmit IoT sensor and the proposed smart, secure, and energy-efficient IoT sensor, respectively. As opposed to the traditional IoT sensor (Fig. 6.1a), our design also includes the various stages depicted on the right (red) part in Fig. 6.1b (i.e., compression, feature extraction, inference, and encryption/hashing). This red part provides two data compression options to the users. One of these options compressively senses the data and carries out the remaining operations in the compressed domain, whereas the other option employs signal processing in the Nyquist domain and then carries out the
compression. Following data compression and processing, the architecture performs classification for both continuous and alert notification scenarios. In the alert notification scenario, the system carries out classification to assess whether an alert needs to be issued (e.g., arrhythmia detected by a smart electrocardiogram sensor or seizure detected by a smart electroencephalogram sensor). In case of an alert, the encrypted and hashed data are transmitted to the base station. However, in the no-alert condition and continuous notification scenario, the classification results are accumulated, stored in the memory, and sent to the base station at specific time intervals.

The architecture allows flexible deployment of the blocks in Fig. 6.1b. Depending on the application and its objectives, a relevant subset of the blocks can be chosen. Fig. 6.2 and 6.3 show the paths through the architecture that correspond to the traditional sense-and-transmit and sense-compress-transmit approaches, respectively, with an option to employ cryptographic techniques. Fig. 6.4 shows the paths corresponding to compression and data processing in the compressed-domain with options to carry out classification and encryption/hashing. The path in Fig. 6.4a extracts features, but does not carry out classification. This path transmits the features extracted from the input data. Since it does not do classification, it cannot raise an alert. Due to the computational load for signal processing and transmission of the feature vector corresponding to each input vector, Fig. 6.4a does not offer much energy efficiency. Thus, we analyze the energy consumption of the path shown in Fig. 6.4b and Fig. 6.4c.

Figure 6.2: IoT sensor architecture for the traditional sense-and-transmit approach.

Figure 6.3: IoT sensor architecture for the sense-compress-transmit approach.
Figure 6.4: IoT sensor architecture that employs direct computations on compressively-sensed data: (a) without classification (b) with classification for alert notification, and (c) with classification for continuous notification.

instead for alert and continuous notification scenarios, respectively. By doing more (i.e., classification), they counter-intuitively require less energy since the output of the classification stage only has few bits per input vector. This dramatically cuts down on the amount of data transmission to the base station. As we will see later, since transmission energy dominates sensor energy, this provides a huge energy benefit.

Fig. 6.5 shows various architectural paths that can be used in a reduced architecture based on compressed signal processing. As in the case of Fig. 6.4a, Fig. 6.5a and 6.5d perform signal processing without utilizing classification. This results in increased energy consumption compared to the traditional sense-and-transmit approach. Since these parts are only useful when energy is not a concern, we do not focus on them. The path shown in Fig. 6.5b and Fig. 6.5c corresponding to alert and continuous
Figure 6.5: IoT sensor architecture based on compressed signal processing with (a) direct transmission, (b) machine learning inference for alert notification, (c) machine learning inference for continuous notification, (d) compression, (e) compression and machine learning inference for alert notification, and (f) compression and machine learning inference for continuous notification.
6.2.2 Energy Modeling

Our IoT sensor architecture includes blocks (i.e., the red part in Fig. 6.1b) that are useful for energy-constrained sensor nodes. Since performing more operations on the sensor node, yet claiming energy efficiency, is counter-intuitive, we need to demonstrate that this is a viable claim through detailed energy modeling. Thus, energy bonus/overhead analyses of each block in Fig. 6.1b is important for assessing the applicability of the proposed architecture. With the inclusion of new blocks in Fig. 6.1b:

- the total number of multiply-accumulate (MAC) operations and static random access memory (SRAM) accesses are impacted,
- the amount of data that requires encryption/hashing and transmission to the base station is altered, and
- classification incurs extra energy.

In order to make a fair comparison between the traditional sense-and-transmit approach and various versions of our proposed architecture, we model the energy of all these versions as shown in Eq. 6.1.

\[
E_{Total} = E_{MAC} + E_{SRAM} + E_{Cl} + E_{Enc} \\
+ E_{Hash} + E_{Tr}
\] (6.1)

\(E_{MAC}\) and \(E_{SRAM}\), respectively, refer to MAC operation and SRAM access energy consumed in the compression and feature extraction blocks. \(E_{MAC}\) and \(E_{SRAM}\) do not
include MAC/SRAM energy consumption in the remaining blocks (i.e., classification, encryption/hashing, and transmission). The MAC/SRAM energy in these blocks are accounted for in their respective energy models: $E_{Cl}$, $E_{Enc}$, $E_{Hash}$, and $E_{Tr}$, which represent classification, encryption, hashing, and transmission energy, respectively. We ignore the energy for analog-to-digital conversion since it is insignificant relative to the other energy components.

We base our analysis on the 130 nm technology. However, the analysis is valid for any other CMOS technology as well. Measurements from a 130 nm CMOS IC indicate 11.8 pJ and 34.6 pJ energy consumption, respectively, for a 32-bit MAC operation and access of data from a 32 kB SRAM [53], [259]. We compute $E_{MAC}$ by multiplying the unit MAC operation energy with the total number of MAC operations in the compression and feature extraction blocks. For example, the multiplication of $M \times N$ and $N \times K$ matrices with 32-bit entries requires $(M \cdot N \cdot K)$ MAC operations, hence, a total of $(11.8 \cdot M \cdot N \cdot K)$ pJ of energy. Likewise, we obtain $E_{SRAM}$ by multiplying the energy of a single SRAM access with the total number of SRAM accesses. In the matrix multiplication example, the SRAM is accessed $(2 \cdot M \cdot N \cdot K)$ times. This leads to an energy consumption of $(34.6 \cdot 2 \cdot M \cdot N \cdot K)$ pJ.

To obtain $E_{Cl}$, we model the classification energy for the random forest, AdaBoosted decision tree, and the K-means algorithms. The resulting classifier of the random forest algorithm includes a large number of decision trees, each of which employs thresholds on feature values for branching. For this computation, we start with a unit thresholding energy of 4.09 fJ obtained based on an 8-bit binary tree comparator in 180 nm CMOS technology [260]. Since this does not match the technology or bit-width assumed for the other blocks, we use four 8-bit comparators to design a 32-bit comparator and Dennard scaling [261] to scale results to the 130 nm CMOS technology. Dennard scaling from 180 nm to 130 nm decreases capacitance
and voltage by a factor of 180/130. Thus, it entails dividing the overall classification energy by \((180/130)^3\). Each tree node in a random forest consumes energy for unit thresholding and two associated SRAM accesses (one for the threshold value and the other for the pointer to the next node) for a single comparison. We obtain the overall \(E_{Cl}\) for the random forest model by multiplying this energy per tree node comparison, the maximum tree depth, and the total number of trees. This is a conservative estimate since not every tree is traversed to its full depth for a given data instance.

The classifier derived from the AdaBoosted decision tree algorithm relies on comparisons (at tree nodes and final output) and multiplications (between tree outputs and their weights) to make a prediction. We model the energy for these two parts separately, and accumulate the results to obtain the final \(E_{Cl}\) for the AdaBoosted decision tree model.

The classifier derived from the K-means algorithm involves the inner product calculations between each incoming instance and the cluster center vectors for similarity analysis. Thus, \(K - 1\) comparisons are needed to obtain the best prediction among the \(K\) cluster centers. We accumulate the incurred energy in all these steps to obtain the final \(E_{Cl}\) for the K-means model.

To obtain \(E_{Enc}\) and \(E_{Hash}\), we use a gate-level implementation of the encryption (AES-128) and hashing (SHA-3) algorithms in 65 \(nm\) CMOS technology [71]. We scale the energy by a factor of \((130/65)^3\) to make it compatible with the 130 \(nm\) CMOS technology.

To obtain the transmission energy, \(E_{Tr}\), we focus on the amount of data that needs to be transmitted to the base station. In case of an alert, we transmit the compressed data immediately; whereas under the no-alert condition, we accumulate the classification results and then transmit. With a BLE connection that sends up to six packets, each containing 20 B of data, we calculate the required number of packets and connection intervals needed for transmission [262]. Then, in line with the study in [263],
using the current and timing measurements of various BLE stages, i.e., wake-up, pre-processing, pre-listening, listening, pre-transmission, transmission, post-processing, pre-sleep, and sleep, we obtain $E_{Tr}$ for BLE based on TI CC2650 module measurements [264]. Since BLE has in-built encryption/hasing [265], encryption/hasing energy is not separately taken into account when using BLE. To obtain $E_{Tr}$ for MICS, we use the 0.51nJ/bit transmission energy based on the transceiver design proposed in [57]. This transceiver accommodates the signal loss due to the shadowing effects from human bodies. It achieves very high modulation accuracy, sensitivity, and interference robustness. However, in the MICS case, encryption/hasing energy needs to be separately added.

### 6.3 Experimental Results

In this section, we present the experimental results for the smart, secure, and energy-efficient IoT sensor architecture when applied to various datasets.

#### 6.3.1 Datasets

Our sensor architecture is versatile and applicable across IoT applications. We evaluate its effectiveness based on six IoT datasets. As shown in Fig. 6.6, we divide these datasets into two groups based on their requirements: alert notification or continuous notification. Arrhythmia, freezing of gait, and seizure detection applications require immediate action to minimize/eliminate unwanted consequences (e.g., injury, brain damage, heart attack, death, etc.). These are called alert conditions. The alert notification system informs the base station/server when the event of interest occurs. For non-alert conditions, the proposed system sends information to the base station/server at specific time intervals to certify that the system is up and running, without the need to trigger alert notification. We introduce the datasets next.
Smart, secure, and energy-efficient IoT sensor response

Alert notification
- ECG based arrhythmia detection
- Parkinson’s disease freezing of gait detection
- EEG based seizure detection

Continuous notification
- Neural prosthesis spike sorting
- Human activity classification
- Chemical gas classification

Figure 6.6: IoT sensor applications for alert and continuous notification.

The MIT-BIH arrhythmia database [266], [267] is composed of ECG measurements that are utilized for arrhythmia detection. The UCI Daphnet Freezing of Gait Dataset [8] is based on acceleration sensor readings that are used to assess motor blocks in patients with Parkinson’s disease. The neural prosthesis dataset [268] is used for spike sorting. The UCI Daily and Sport Activities Dataset [269], [270], [271] is composed of accelerometer, gyroscope, and magnetometer measurements that are used for human activity detection. The CHB-MIT Scalp EEG Database [266], [272] includes EEG measurements that are used for epileptic seizure detection. The UCI Gas Sensor Array Drift Dataset [273] includes metal-oxide gas sensor measurements for chemical gas classification.

We present the experimental results for each of the six datasets next. The first five datasets contain linear features. These features can be extracted via matrix-vector multiplications, where both the compressed-domain feature extraction technique [51, 52] and the CSP technique [53] can be directly applied to cut down on computation energy. We achieve 57.1-379.8× energy reduction for these datasets. Then, we present our results for the chemical gas classification dataset that involves nonlinear features. We show that even though the nonlinear features cannot be extracted via matrix-
vector multiplication, the nonlinearity can be easily handled by existing circuitry in sensor nodes. We achieve $912.6 \times$ energy reduction for this dataset.

### 6.3.2 Alert Notification

The alert conditions in ECG-based arrhythmia detection, freezing of gait detection for Parkinson’s disease, and EEG-based epileptic seizure detection applications are an irregular heart rhythm (both supraventricular and ventricular), freezing event, and epileptic seizure, respectively. For arrhythmia, the irregular supraventricular and ventricular heartbeats are detected through the use of classifiers. In order to be in line with the previous approaches, we use the heartbeat annotation of The Association for the Advancement of Medical Instrumentation (AAMI) [274], [275]. For freezing of gait, the freezing event is detected through the energy of the accelerometer signal in specific frequency bands with the help of classifiers. Similarly, for EEG-based epileptic seizure detection, the seizure episode is detected by computing the EEG energy in eight different frequency bands and employing classifiers. Overall, the alert notification systems detect irregular events.

We provide a detailed explanation of each application next.

**ECG based Arrhythmia Detection**

ECG is a physiological signal that provides information on the electrical activity of the heart [98]. It is used to detect irregularities in the cardiovascular system, such as arrhythmia. Arrhythmia [276] is an irregularity of the heart rhythm. If the symptoms are not detected at an early stage, it can lead to cardiac arrest, heart attack, or even death. Continuous ECG monitoring is one of the methods to avoid these severe consequences and enhance patient wellness. Our IoT sensor architecture is directly applicable to the continuous ECG monitoring application. It significantly improves
battery lifetime based on on-sensor data compression and classification, and improves security based on encrypted/hashed data transmission.

We use the MIT-BIH Arrhythmia Database [266], [267] to analyze our architecture in terms of accuracy and energy consumption. This database includes 48 ECG data sections collected from 47 participants [267]. Each data section includes 30 minutes of ECG measurements based on a 360 Hz sampling rate. The ECG measurements are composed of various heartbeat waveforms. AAMI [274], [275] advocates grouping these independently annotated beats into five different classes: normal (N), supraventricular ectopic (S), ventricular ectopic (V), fusion (F), and unknown beat (Q). We follow the AAMI standard in our evaluation. Moreover, due to the reduced signal quality issues discussed in AAMI-recommended practice [274], [275], we discard four data sections, which contain paced heartbeat signals, from our analyses. Since the goal of our architecture is to detect arrhythmia from ECG heartbeat signals, we design two different binary classifiers to identify the S and V beats.

We design the sensor node using the first lead of the ECG signal. We expect the arrhythmia classification accuracy to improve further when both leads are used. However, our aim is to show the applicability of our sensor architecture to various ECG sensors, including single-lead and multiple-lead ones. Therefore, we use the ECG data from the first lead in the MIT-BIH Arrhythmia Database. Also, we assume that the ECG sensor provides R-peak positions. This is a reasonable assumption since several on-market sensors are capable of detecting R-peaks. With advancing technology, more sensors are expected to provide this information. However, if this information is not available, then Discrete Wavelet Transform (DWT) is one of the methods for detecting R-peaks. Sasikala et al. [277] show that DWT detects R-peaks with 99.89% sensitivity. Since DWT can be represented in matrix form [51], although it is out of the scope of this chapter, our architecture is capable of performing R-peak detection.
Following the acquisition of R-peak positions, we divide the ECG signal into 256 sample epochs that cover approximately 0.3 s interval before and 0.4 s after the R-peak. Since these are time-series data, we use the first 80% of the epochs as the training set and the remaining as the test set. From each of these epochs, we extract bandpass filter (BPF) and DWT features. To obtain the BPF features, we pass the corresponding ECG signal through 30 BPFs (0.5-2.0 Hz, 2.0-3.5 Hz, 3.5-5.0 Hz, ..., 44.0-45.5 Hz). This involves the convolution operation. We execute this operation by obtaining the row-shifted version of BPF coefficients [51]. At the end of this row-shift, an $M \times M$ convolution matrix is obtained from the array of BPF coefficients ($1 \times M$). Then, we multiply the convolution matrix with the ECG data of the corresponding epoch. At the final step, we compute the inner product of the resulting array and store it in the memory for use in the classification stage. We repeat this process for each BPF. To obtain DWT features, we use a 6-level DWT to capture arrhythmia information. We implement DWT with a filter bank (consecutive high-pass and low-pass filters) [51]. To pass ECG data through high-pass and low-pass filters, we employ the above-mentioned procedure. After each filtering stage, we down-sample the output by 2×. Since the filters have half the bandwidth of the original signal, aliasing due to down-sampling is avoided [51]. Coefficients obtained through DWT are stored in the memory for use in the classification stage.

Due to its simplicity and fast inference, we use random forest with 100 decision trees having unlimited maximum tree depth for binary classification of S and V beats in the ECG signal (we use the Weka 3.8.0 [281] platform for deriving the machine learning models in our work). Table 6.1 and Table 6.2 show the classification accuracy (ACC), true positive rate (TPR), and true negative rate (TNR) for results from the relevant work in the upper section, and results based on our implementations in the lower section for S-beat and V-beat, respectively. Our results cover the following cases: no compression, 5×, 10×, 15×, and 20× compression based on direct compu-
Table 6.1: Supraventricular (S) Ectopic Beat Detection Performance

<table>
<thead>
<tr>
<th>Case (S-beat)</th>
<th>ACC (%)</th>
<th>TPR (%)</th>
<th>TNR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chazal et al. [278] (non-adaptive)</td>
<td>94.6</td>
<td>75.9</td>
<td>95.3</td>
</tr>
<tr>
<td>Chazal et al. [278] (adaptive-500 beats)</td>
<td>95.9</td>
<td>87.7</td>
<td>96.2</td>
</tr>
<tr>
<td>Hu et al. [279]</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ince et al. [280]</td>
<td>97.4</td>
<td>63.5</td>
<td>99.0</td>
</tr>
<tr>
<td>No compression (Fig. 6.5b)</td>
<td>92.6</td>
<td>85.9</td>
<td>92.8</td>
</tr>
<tr>
<td>5× with Method I (Fig. 6.4b)</td>
<td>94.0</td>
<td>85.0</td>
<td>94.2</td>
</tr>
<tr>
<td>5× with Method II (Fig. 6.5e)</td>
<td>89.7</td>
<td>86.5</td>
<td>89.8</td>
</tr>
<tr>
<td>10× with Method I (Fig. 6.4b)</td>
<td>93.2</td>
<td>83.1</td>
<td>93.4</td>
</tr>
<tr>
<td>10× with Method II (Fig. 6.5e)</td>
<td>91.3</td>
<td>86.3</td>
<td>91.4</td>
</tr>
<tr>
<td>15× with Method I (Fig. 6.4b)</td>
<td>97.6</td>
<td>72.4</td>
<td>98.3</td>
</tr>
<tr>
<td>15× with Method II (Fig. 6.5e)</td>
<td>90.2</td>
<td>87.7</td>
<td>90.3</td>
</tr>
<tr>
<td>20× with Method I (Fig. 6.4b)</td>
<td>94.2</td>
<td>81.4</td>
<td>94.6</td>
</tr>
<tr>
<td>20× with Method II (Fig. 6.5e)</td>
<td>90.8</td>
<td>87.0</td>
<td>90.9</td>
</tr>
</tbody>
</table>

Table 6.2: Ventricular (V) Ectopic Beat Detection Performance

<table>
<thead>
<tr>
<th>Case (V-beat)</th>
<th>ACC (%)</th>
<th>TPR (%)</th>
<th>TNR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chazal et al. [278] (non-adaptive)</td>
<td>97.4</td>
<td>77.7</td>
<td>98.8</td>
</tr>
<tr>
<td>Chazal et al. [278] (adaptive-500 beats)</td>
<td>99.4</td>
<td>94.3</td>
<td>99.7</td>
</tr>
<tr>
<td>Hu et al. [279]</td>
<td>94.0</td>
<td>82.6</td>
<td>97.1</td>
</tr>
<tr>
<td>Ince et al. [280]</td>
<td>98.3</td>
<td>84.6</td>
<td>98.7</td>
</tr>
<tr>
<td>No compression (Fig. 6.5b)</td>
<td>98.7</td>
<td>85.7</td>
<td>99.7</td>
</tr>
<tr>
<td>5× with Method I (Fig. 6.4b)</td>
<td>97.8</td>
<td>83.7</td>
<td>98.9</td>
</tr>
<tr>
<td>5× with Method II (Fig. 6.5e)</td>
<td>98.0</td>
<td>87.3</td>
<td>98.8</td>
</tr>
<tr>
<td>10× with Method I (Fig. 6.4b)</td>
<td>97.3</td>
<td>84.1</td>
<td>98.3</td>
</tr>
<tr>
<td>10× with Method II (Fig. 6.5e)</td>
<td>98.6</td>
<td>86.9</td>
<td>99.5</td>
</tr>
<tr>
<td>15× with Method I (Fig. 6.4b)</td>
<td>97.1</td>
<td>84.7</td>
<td>98.0</td>
</tr>
<tr>
<td>15× with Method II (Fig. 6.5e)</td>
<td>98.3</td>
<td>86.2</td>
<td>99.2</td>
</tr>
<tr>
<td>20× with Method I (Fig. 6.4b)</td>
<td>96.4</td>
<td>77.0</td>
<td>97.9</td>
</tr>
<tr>
<td>20× with Method II (Fig. 6.5e)</td>
<td>98.7</td>
<td>84.6</td>
<td>99.7</td>
</tr>
</tbody>
</table>

...ations on compressively-sensed data (Method I) and CSP (Method II). We achieve a similar performance to those by Chazal et al [278]. Their adaptive approach uses a combination of global and local classifiers. It uses the first 500 beats of the training data to train the local classifier. The non-adaptive approach only employs a global classifier [278]. Since the adaptive approach learns more information on the heartbeat...
Figure 6.7: Total energy consumption of S-beat and V-beat detecting architectural paths (i.e., those shown in Fig. 6.2, 6.3, 6.4b, and 6.5e) with 1×, 5×, 10×, 15×, 20× compression factors.

types, it results in a higher classification performance. Moreover, compared to results by Hu et al. and Ince et al., we achieve a higher TPR. Since TPR indicates the percentage of correctly classified S and V beats, even with significantly compressed data, our architecture detects arrhythmia more accurately. Based on results from our implementations, we observe similar performance even when the data are compressed.

Classification performance is important. However, we also need to evaluate the energy consumption to assess the feasibility of our IoT sensor architecture. We compute the total energy consumption for the different architectural paths, which include encryption and hashing, shown in Fig. 6.2, 6.3, 6.4b, and 6.5e. The result for the architectural path depicted in Fig. 6.5b is shown under the ‘No compression’ case of Fig. 6.5e, as they are equivalent. We compare the total energy consumption of these paths in Fig. 6.7 for S-beat and V-beat detection. Overall, we observe a continuing decrease in energy consumption (an increase in energy bonus) with increasing compression factors. This result is as expected. As we compress the data/features more, the amount of signal processing and transmission packet size both decrease. Machine
learning inference decreases the energy consumption by reducing the raw data to a few bits of inference. Since our system sends the compressed data only when an alert is raised (i.e., when arrhythmia has been detected) and accumulates the inference results for the no-alert case for a more regular data transmission, significant data transmission energy is saved in the no-alert situation. We can see that up to $57.1\times$ (for the S-beat and V-beat) lower energy is needed to obtain performance comparable to the traditional sense-and-transmit approach.

**Freezing of Gait Detection for Parkinson’s Disease**

Parkinson’s disease affects motor abilities of the patient negatively [282]. One of its consequences is freezing of gait (FoG) due to which the patient loses the ability to move his/her leg temporarily. Sometimes, arms and eyelids are also frozen temporarily [282], [283]. FoG causes injuries to patients, since it arises abruptly and leads to falls [282], [284]. In order to reduce FoG periods and prevent possible falls, Bachlin et al. [8] propose a wearable system that provides rhythmic sounds in case a FoG period is detected. This FoG detection application is also directly amenable to implementation with our sensor architecture, since it employs wearable sensors and requires online FoG detection. Use of wearable sensors necessitates long battery lifetimes, since frequent battery charging or replacement negatively impacts practicality and system adoption.

We use the UCI Freezing of Gait Dataset [8] to evaluate our architecture in terms of accuracy and energy consumption. The dataset includes accelerometer measurements from ten patients: eight with and two without FoG periods. We implement personalized FoG detection. Since two patients do not experience FoG, we use data from the remaining eight patients. The dataset contains measurements from three-axis accelerometers obtained using a 64 Hz sampling rate. The accelerometers are positioned at the shank, thigh, and belt of the patients.
In order to classify the FoG periods, rather than applying thresholding as proposed in the study by Bachlin et al. [8], we again use the random forest algorithm due to its simplicity and fast inference [285]. However, to be able to compare classification performance, in line with [8], we use 4 s windows (256 samples) with 0.5 s (32 samples) shifts in between. To provide personalized care, we build the classifier models for each patient separately. Keeping in mind that we have time-series data, we use the first 70% of the windows as the training set and the remaining windows, which do not have any overlap with the training set, as the test set. Based on the study in [8] and [286], we use two BPFs with 0.5-3.0 Hz and 3.0-8.0 Hz bands. Since passing a signal through a BPF requires a convolution operation, we obtain convolution matrices for each filter by shifting the rows of BPF coefficients one by one for each row [51]. We multiply the corresponding vector of the accelerometer data with the two BPF convolution matrices. Then, we compute the inner product of the output array and store it in the memory for use in the classification stage. We repeat the above procedure for each window in the training and test sets. Moreover, we handle the imbalance between classes of the target feature with the SMOTE method [287]. We use random forest with 100 decision trees having unlimited maximum tree depth for the classification of FoG periods. For each of the eight patients, we obtain the classifier model based on the training set and analyze the performance of the corresponding model on the test set. In the classification stage, we use the features obtained from y-axes of the three accelerometer sensors. Table 6.3 shows ACC, TPR, and TNR for the results from relevant work in the upper section, and results based on our implementations in the lower section. Our results cover the following cases: no compression, 5×, and 10× compression using Method I and Method II. We achieve lower performance compared to the study by Mazilu et al. [288]. However, Mazilu et al. choose training and test sets randomly for a 10-fold cross-validation. Random selection discards the time-series nature of the data and results in correlated data points in the training.
Table 6.3: Parkinson’s Disease Freezing of Gait Detection Performance

<table>
<thead>
<tr>
<th>Case</th>
<th>ACC (%)</th>
<th>TPR (%)</th>
<th>TNR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mazilu et al. [288]</td>
<td>-</td>
<td>99.7</td>
<td>100.0</td>
</tr>
<tr>
<td>Bachlin et al. [8]</td>
<td>-</td>
<td>73.1</td>
<td>81.6</td>
</tr>
<tr>
<td>No compression (Fig. 6.5b)</td>
<td>86.2</td>
<td>81.8</td>
<td>87.1</td>
</tr>
<tr>
<td>5× with Method I (Fig. 6.4b)</td>
<td>68.5</td>
<td>64.9</td>
<td>68.7</td>
</tr>
<tr>
<td>5× with Method II (Fig. 6.5e)</td>
<td>84.8</td>
<td>79.9</td>
<td>86.1</td>
</tr>
<tr>
<td>10× with Method I (Fig. 6.4b)</td>
<td>72.1</td>
<td>64.5</td>
<td>73.7</td>
</tr>
<tr>
<td>10× with Method II (Fig. 6.5e)</td>
<td>84.1</td>
<td>79.8</td>
<td>85.0</td>
</tr>
</tbody>
</table>

and test sets. This boosts ACC, TPR, and TNR. In our study, we take into account the time-series nature of data by using the first 70% of the data as the training set and the remaining non-overlapping part as the test set. Thus, we obtain a realistic assessment of classifier performance. Moreover, compared to the more realistic study on online FoG detection by Bachlin et al., we achieve higher TPR and TNR values. Bachlin et al. use thresholding; however, we employ machine learning algorithms to detect FoG. Machine learning algorithms are able to identify more complex patterns and thus result in higher classification performance.

We observe comparable classification performance for no compression and Method II with 5× and 10× compression factors. However, with Method I, we could not achieve as high a performance as with Method II. Method I performs feature extraction in the compressed domain. This introduces a small amount of error in the inner product values. In this case, these errors were consequential whereas they were not in the case of arrhythmia detection.

We also perform energy consumption analyses. Fig. 6.8 shows the total energy consumption of the architectural paths, which include encryption and hashing, shown in Fig. 6.2, 6.3, 6.4b, and 6.5e. We can see that Method II, with compression factors of 5× and 10× (Fig. 6.5e), reduces energy consumption by 217.0× and 379.8×, respectively, relative to the traditional sense-and-transmit case. Thus, our architecture
results in a huge energy bonus. As pointed out earlier, the cryptographic techniques are already integrated into BLE operations.

**EEG based Seizure Detection**

Epilepsy is a neurological disorder that can lead to an abrupt loss of consciousness and body convulsion. It currently affects 4-5% of the world population [289]. Abrupt epileptic seizure onsets pose physical risks to epilepsy patients. A continuous seizure detection system can help mitigate these risks and enhance the quality of healthcare. Such a system also needs to be secure and energy-efficient, making it a suitable candidate for our approach.

Lu et al. conducted a detailed performance comparison of Method I and Method II for EEG epileptic seizure detection in [53]. They used the data collected from three epileptic patients present in the CHB-MIT Scalp EEG Database [266], [272] for performance evaluation. They partitioned the EEG signal streams into three-second signal epochs with a two-second overlap. They used eight BPFs (sequentially covering the 0-24 Hz range with 3 Hz individual pass-band widths) to extract the
spectrum energy features from the data epochs. They used an AdaBoosted decision
tree classifier for seizure detection. There are three classifier performance metrics that
are important for this application: sensitivity (same as TPR), latency, and false alarm
rate [53, 51, 266]. Lu et al. achieved 100% sensitivity at different compression factors. However, Method II outperformed Method I in latency and false alarm rate. Method I is based on data processing in the compressed domain. As in the case for freezing of gait detection, the error introduced in the compressed domain has substantial effect on the classification performance.

We conduct energy consumption analyses for this application based on the experimental setup used in [53]. The EEG signals are captured by an 18-channel EEG sensor front-end. The sampling rate for each channel is $256 \text{ Hz}$. For the energy evaluation of the feature extraction step, we consider two distinct steps that require MAC operations and SRAM accesses. The first step is linear filtering by BPFs that involves matrix-vector multiplication. The second step is energy accumulation based on the inner product of the post-filtered 144-dimensional (8 features per channel × 18 channels) signal vector with itself. We sum the energy from these two steps to model the feature extraction energy. We model the classification energy based on the AdaBoosted decision tree parameter set used in [53] that achieves the best performance. Hence, we set the maximum number of weak classifiers to 200. A weak classifier is a shallow decision tree with a maximum depth of three. As a result, the worst-case classification energy corresponds to a data instance that requires 601 comparisons (200 trees × maximum tree depth of 3 + one final comparison) and 200 multiplications (one per tree).

We summarize the energy results in Fig. 6.9. Our architecture cuts the energy consumption by $5.8 \times$ for the no-compression case (i.e., when signals are in the Nyquist domain). This ratio increases to $68.5 \times$ and $139.7 \times$ for $12 \times$ and $24 \times$ compression factors, respectively.
Figure 6.9: Total energy consumption for EEG seizure detection for various architectural paths (i.e., those shown in Fig. 6.2, 6.3, 6.4b, and 6.5e) with 1×, 12×, and 24× compression factors.

6.3.3 Continuous Notification

Continuous notification sensors provide regular feedback to the user. Their data transmission to the base station does not depend on the classifier output outcome. The human activity classification, neural prosthesis spike sorting, and chemical gas classification applications fall into this category. For human activity classification, the activities are determined through inputting the collected electromechanical signals to the machine learning algorithm. For neural prosthesis, the spikes are classified by inputting the DWT features of neural spikes to a clustering algorithm. Similarly, the chemical gases are determined by inputting the combination of linear and nonlinear features to the corresponding classifiers.

We provide a detailed explanation of the data processing techniques, decision-making mechanisms, and performance analyses for each application next.
Human Activity Classification

State-of-the-art body-wearable sensors enable myriad daily applications, such as daily activity monitoring [269], sleep status analysis [290], stress detection and alleviation [33], and pervasive disease diagnosis [34]. Such a sensor has to be both secure, hence protect user’s privacy, and energy-efficient, hence increase the battery lifetime for user’s convenience and satisfaction. In this section, we target human activity classification with body-worn miniature inertial sensor units. This application enables its users to have a direct, fine-grained visualization of their life logs.

We use the UCI Daily and Sport Activities Dataset [269], [270], [271] to evaluate our architecture for this application. This dataset contains 19 different activities that comprehensively cover a wide range of daily routines: sitting (1), standing (2), lying on back (3) and right side (4), ascending (5) and descending stairs (6), standing still (7) and moving around (8) in an elevator, walking in a parking lot (9), walking on a treadmill with a speed of 4 km/h in flat (10) and 15-degree inclined positions (11), running on a treadmill with a speed of 8 km/h (12), exercising on a stepper (13), exercising on a cross trainer (14), cycling on an exercise bike in horizontal (15) and vertical positions (16), rowing (17), jumping (18), and playing basketball (19). The data are collected from four female and four male participants. Each participant wears five body-worn miniature inertial sensor units [265]. Each tracker contains a tri-axial accelerometer, a tri-axial gyroscope, and a tri-axial magnetometer. The sampling rate per sensor per axial is 25 Hz. Each activity contains five minutes of data per participant, and is divided into 5-second intervals [269]. Therefore, there are 480 data intervals per activity.

We next train the classifiers for this application. In [269], Altun et al. extracted 1170 features from each 5-second data interval, and then reduced the feature dimension to 30 through principal component analysis. They performed a leave-one-out (L1O) analysis for this application. In this analysis, a new machine learning model is
trained in each L1O iteration based on the data from seven participants (7980 training instances = 60 vectors per person per activity × 7 persons × 19 activities) and tested on data from the remaining participant (1140 testing instances = 60 vectors per person per activity × 1 person × 19 activities). Data from each participant are used only once for testing. The average accuracy over the eight iterations denotes the final performance. They achieved the highest accuracy of 87.6% with a support vector machine (SVM) classifier. The system delay is five seconds because of the 5-second data interval required for feature extraction.

In order to decrease the system delay and improve classification performance, we employ a 0.04 s data interval and utilize the first 80% of the participants’ data (7296 training instances = 48 vectors per person per activity × 8 persons × 19 activities) for training and the remaining (1824 testing instances = 12 vectors per person per activity × 8 persons × 19 activities) for testing. We achieve 96.0% accuracy using the random forest algorithm (100 trees with a maximum tree depth of 10). Our design is approximately 125× faster than the design in [269].

In Fig. 6.10, we show the classification accuracy for various compression factors. The classification accuracy drops from 96.0% in the Nyquist domain to 69.7% and 50.1% with 5× and 15× compression factors, respectively. This significant performance degradation may be explained by the fact that the 19-class classification task is very challenging, thus more sensitive to information loss caused by compression.

Due to the significant accuracy degradation in the compressed domain, we perform energy analyses for this application only in the Nyquist domain. We consider worst-case random forest classification energy for an input data instance that requires 1K comparisons (100 trees × a maximum tree depth of 10). The sensor transmission protocol is assumed to be BLE [265]. Thus, we do not model encryption and hashing energy separately as BLE already incorporates them.
Figure 6.10: Compressed-domain classification accuracy for the 19-class daily activity classification task using random forest.

Table 6.4: Energy Breakdown for the 19-class Human Activity Classification without Compression

<table>
<thead>
<tr>
<th>Case</th>
<th>SRAM Access</th>
<th>Classification</th>
<th>Transmission</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>-</td>
<td>-</td>
<td>8.80m.J</td>
<td>8.80m.J</td>
</tr>
<tr>
<td>Ours</td>
<td>0.31μJ</td>
<td>6.93μJ</td>
<td>33.39μJ</td>
<td>40.63μJ</td>
</tr>
</tbody>
</table>

We summarize the results in Table 6.4. Its columns correspond to the method, SRAM access energy for local data instance storage, classification energy, transmission energy, and total energy consumption for this application. The first row shows the energy values for the traditional sense-and-transmit approach (Fig. 6.2). The second row shows the energy values for our proposed approach (Fig. 6.5c). We achieve \(216.6\times\) energy reduction for this application.

**Neural Prosthesis**

Neural prosthetic systems enable external devices to collect, analyze, and respond to the neural activities in human brains through prosthetic implants. These systems can alleviate treatment-resistant depression and chronic pain, Alzheimer’s disease, post-traumatic stress disorder, traumatic brain injury, speech disability, and sustained
spinal cord injury. Due to the high risk of surgery, the lower the energy computation (and hence longer the battery lifetime) of the prosthetic implants, the better.

We evaluate our sensor architecture for neural prostheses that detect and sort neural spikes. To address the energy and bandwidth constraints, the spike analysis process is carried out on biomedical implants. The spike analysis process has five sequential stages: analog-to-digital conversion, spike detection, spike alignment, feature extraction, and spike classification. The classification results are transmitted to an external controller via low-power transceivers that operate in the MICS band.

For neural prostheses, Lu et al. conducted a detailed performance comparison of Method I and Method II in [53]. They used spike records E1, E2, D1, and D2 from [268] for performance evaluation. They simulated the records for three neural spike classes under 18 different neural noise levels. They extracted DWT features and used a K-means classifier for spike sorting. According to their results, Method I (Method II) can correctly classify 87.8% (87.8%) of the spikes without compression with a K-means classifier. This accuracy slightly drops to 83.9% (85.6%) and 82.5% (83.2%) with 4× and 8× compression factors, respectively.

We conduct energy consumption analyses for this application based on the experimental setup used in [53]. The analog-to-digital converter front-end samples signals at 24,000 Hz with 32 bits per sample. The system uses an amplitude threshold method (a 4δ rule based on standard deviation of the background noise [268]) for spike detection. We consider the associated comparisons and SRAM accesses for the spike detection energy calculation. Whenever a spike is detected, the system keeps track of 64 samples in one data segment for further analysis. The average frequency of spike occurrence is 51.9 neural spikes per second. These spikes incur subsequent feature extraction, classification, and transmission energy consumption. In the Nyquist domain, 64 DWT coefficient features are extracted from each spike data segment.
for subsequent classification (the number can be reduced in the compressed domain using Method I and Method II to reduce computation energy). We consider both the MAC operations and SRAM accesses due to the matrix-vector multiplications for DWT feature extraction energy calculation. Since this application needs a three-class classification, we model the energy for a K-means classifier that contains three cluster centers. Data transmission is based on MICS transceivers [268]. For security purpose, we use AES-128 for encryption and SHA-3 for hashing, and also model their energy.

We summarize our results in Fig. 6.11. This figure shows the total energy consumption of various architectural paths, which include encryption and hashing. Without compression (i.e., when Nyquist-domain feature extraction and classification are done in the path shown in Fig. 6.5f or equivalently Fig. 6.5c), our proposed sensor architecture is able to cut the energy consumption by $22.8 \times$ against a conventional sense-and-transmit approach. This ratio further increases to $86.7 \times$ and $162.8 \times$ with $4 \times$ and $8 \times$ compression factors, respectively. This shows that our method can provide a huge energy benefit, while at the same time adding security and smartness bonuses.

**Chemical Gas Classification with Nonlinear Features**

Next, we analyze the impact of our sensor architecture on an application that requires nonlinear features. We show that even though such features cannot be extracted via matrix-vector multiplications, the nonlinearity may be easily handled by existing sensor circuitry.

Specifically, we focus on the classification of air flow chemical composition using chemical sensors. Through this application, we aim to evaluate the feasibility of our approach for industrial automation/monitoring. Industrial IoT sensors are better than humans at capturing data consistently and accurately. However, transmitting all the raw sensor data to the cloud consumes substantial sensor energy, server stor-
Figure 6.11: Total energy consumption for neural prosthesis for various architectural paths (i.e., those shown in Fig. 6.2, 6.5d, 6.4c, and 6.5f) with 1×, 4×, and 8× compression factors. The path shown in Fig. 6.5d transmits compressed neural spikes after encryption.

We use the UCI Gas Sensor Array Drift Dataset to evaluate our architecture in terms of accuracy and energy consumption. This dataset is targeted at six chemical gases: ammonia, acetaldehyde, acetone, ethylene, ethanol, and toluene. The data are collected by 16 commercially available chemical sensors in a controllable test sensing chamber. Each sensor yields a time series of measurements. Eight features are extracted from each time series. The feature vectors and their labels are placed in ten data batches prior to upload to UCI. We use batch 1 for training (first two months of data) and batch 2 for testing (next five months of data) for two reasons. First, we want to avoid the sensor degradation phenomena in later data batches. Also, we need both the training and testing datasets to contain labels for all six target gases to enable a more comprehensive analysis. One handicap we have in dealing with this dataset is that the raw data are not available, just the feature vectors.
We first analyze this problem in the Nyquist domain. We achieve a classification accuracy of 75.6% using random forest (100 trees with a maximum tree depth of 10). This is 1.2% higher than the 74.4% accuracy reported in [293].

Next, we evaluate the feasibility of applying Method I and Method II to solve this problem in the compressed domain. There are two types of features stored in the dataset: (i) steady-state, and (ii) exponential moving average. A steady-state feature is derived from the maximum and minimum values of the sensor data streams, and is thus nonlinear [293]. Two major steps are needed to extract an exponential moving average feature: the exponential moving average transform and minimum/maximum value extraction [293]. The first step is linear, whereas the second step is not. In the first step, the exponential moving average transform maps an incoming data stream $\vec{r}$ to its exponential moving average series $\vec{y}$. An instance of $\vec{y}$ at time $t$, denoted by $y(t)$, is linearly dependent on $y(t - 1)$ and two data instances $r(t)$ and $r(t - 1)$, which are instances of $\vec{r}$ at time $t$ and $t - 1$, respectively [293]. Hence, each $y(t)$ can be expanded as a linear combination of $r(k)$, $k \leq t$, through iterative expansion of $y(t')$, $t' < t$. However, the second step, which extracts the maximum and minimum values from $\vec{y}$, is nonlinear. This makes it impossible to derive the exponential moving average feature through linear transformations.

We see that both the steady-state and exponential moving average features are nonlinear. However, we can see that nonlinearity is only introduced due to the need to acquire the maximum and minimum values. These values can be easily captured through a series of comparisons.

To handle the nonlinear features, we add a new path to our architecture, as shown in Fig. 6.12. It contains both linear and nonlinear transformation blocks in its feature extraction stage. Data flow between these two blocks is governed by the required feature extraction for the target application. For example, in this application, the nonlinear transformation block in Fig. 6.12 computes the maximum/minimum values
Figure 6.12: A new architectural path that contains the nonlinear transformation block.

Table 6.5: Energy Breakdown for the Six-class Chemical Gas Classification in the No-compression Case

<table>
<thead>
<tr>
<th>Case</th>
<th>Processing and SRAM access</th>
<th>Transmission</th>
<th>Total Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.03(mJ) -</td>
<td>0.94(J)</td>
<td>0.94(J)</td>
</tr>
<tr>
<td>Ours</td>
<td>0.22(mJ)</td>
<td>69.28(nJ)</td>
<td>0.81(mJ)</td>
</tr>
</tbody>
</table>

of its input. It accepts raw data inputs from the IoT sensors as well as the series of exponential moving averages from the linear transformation block. As a result, this path enables the derivation of both the steady-state and exponential moving average features for this application. Since the data transmission protocol is BLE, we do not model the encryption and hashing energy separately.

We summarize the results in Table 6.5. Its columns depict the case, processing and SRAM access, classification energy, transmission energy, and total energy consumption for this application. The first row presents the energy values for the sense-and-transmit approach (Fig. 6.2). The second row presents the energy values for our proposed approach (Fig. 6.12). In the Nyquist domain, our architecture achieves a 912.6\(\times\) energy reduction relative to the baseline (Fig. 6.2). We could not conduct a compressed-domain analysis due to the unavailability of \(\vec{r}\) and \(\vec{y}\).

We observe that the extraction of the maximum/minimum values incurs very little energy overhead: it only consumes 23.64\(nJ\) extra energy on top of the 50.98\(\mu J\)
of MAC and $0.17mJ$ of SRAM energy needed for the linear transformation. This demonstrates our method can also tackle nonlinear features.

### 6.4 Chapter Summary

In this chapter, we proposed a novel IoT sensor architecture that is simultaneously smart, secure, and energy-efficient. We evaluated the architecture on six different IoT applications: arrhythmia detection, Parkinson’s disease freezing of gait detection, epileptic seizure detection, neural prosthesis spike sorting, human activity detection, and chemical gas detection. While the classification accuracies for each of these applications were comparable to the state-of-the-art ones, we obtained $57.1 \times$ to $912.6 \times$ energy reduction compared to traditional sense-and-transmit IoT sensor.
Chapter 7

Conclusion and Future Work

In this chapter, we summarize our findings and discuss future work.

7.1 Summary of Findings

In Chapter 3 we introduced a new dimension (semantic space) to the feature space based decision-making employed in ML algorithms and encapsulated it in a dual-space classification approach called SECRET. As opposed to traditional approaches, SECRET maps data to labels while integrating meaning-based relationships among labels. We analyzed SECRET’s classification performance on ten datasets that represent different real-world applications. Compared to traditional supervised learning, SECRET achieved up to 14.0% accuracy and 13.1% F1 score improvements. Compared to ensemble methods, SECRET achieved up to 12.7% accuracy and 13.3% F1 score improvements. We also took a step toward understanding how SECRET builds the semantic space component and its impact on overall classification performance.

In Chapter 4 we introduced SoDA, an automatic stress detection and alleviation system that is adaptive and requires minimum user involvement. We designed, implemented, and analyzed the system with multiple options and stress mitigation techniques. The system was shown to be capable of responding to and reducing the
stress level of its user more effectively than when the ‘no therapy’ option is used. SoDA provides two options to its user: ‘individualized’ and ‘generalized’. The ‘individualized’ model is more accurate (95.8% as opposed to 89.3% for the ‘generalized’ model). However, it requires physiological training data to be collected from the user for the derivation of the model. The ‘generalized’ model can be used as is.

In Chapter 5 we described a user-centered operational flow for AI technologies. Rather than directly focusing on the task, as traditional approaches do, our flow targets understanding the users first and then completing the task. The traditional approach has been shown to be effective for various ML applications. However, understanding the users can augment the efficacy of such approaches. We introduced YSUY, an ML-based system for strengthening human-AI interactions and enhancing the quality and timing of various processes. YSUY places major emphasis on understanding the various states (physical, mental, and emotional) of the users before completing the task. This is aimed at closing the human need/AI gap. We have implemented YSUY with WMSs and a smartphone, then experimented with it in the context of the natural flow of participants’ lives. We obtained physical, mental, and four-class (two-class) emotional state classification accuracy values of 90.0%, 90.3%, and 98.4% (99.5%), respectively. We discussed how YSUY can be used to address basic human needs, as enunciated in Max-Neef’s 36-cell matrix. Since YSUY has the potential to address most of these cells, it can be further developed to bridge the human need/AI gap.

In Chapter 6 we proposed a novel IoT sensor architecture that is smart, secure, yet energy-efficient. The IoT sensor designer can choose one from among many paths through this architecture based on which one is the most suitable for the targeted IoT application. We evaluated the architecture on IoT applications picked from different domains: arrhythmia detection, Parkinson’s disease freezing of gait detection, epileptic seizure detection, neural prosthesis spike sorting, human activity detection, and
chemical gas detection. We investigated both the classification accuracy and energy consumption of the architecture. We showed that the classification accuracies were comparable to the state-of-the-art for these applications, yet with energy consumption up to three orders of magnitude lower than IoT sensors based on the traditional sense-and-transmit approach.

7.2 Future Work

Next, we discuss future work under the various research categories we targeted in this thesis.

SECRET: Semantically Enhanced Classification of Real-world Tasks

Further improvements in SECRET’s overall classification performance and feature/semantic space characteristics can be made as follows. First, further analyses of different datasets are needed to support extensive applicability of SECRET. Second, although MLP and RF are well-known supervised ML algorithms, other ML algorithms need to be analyzed in this context. Third, semantic vectors could be trained specially for SECRET and the corresponding application of interest, as done in the case of intrinsic and extrinsic analyses in NLP [294, 295]. Fourth, detailed classification analyses need to be carried out for the multilabel classification task, where SECRET can be implemented with multilabel classification and regression algorithms targeted at the feature and semantic spaces, respectively. Finally, in addition to the feature and semantic spaces, other information sources for classification should be explored.

Keep the Stress Away with SoDA: Stress Detection and Alleviation System

The efficacy of the proposed method, SoDA, is likely to improve as more WMSs get integrated into one or a small number of wearable platforms. This would make it more
convenient to wear them and, hence, more likely to be used. This is already beginning to happen in the marketplace. Second, the proposed methodology can be used to evaluate the impact of other stressors and stress mitigation techniques to augment the menu of options available to the user. Third, the experiments could be carried out for more than 24 hours (when the sensors become more comfortable to wear) to assess the relationship between the circadian rhythm and stress response of the user. Fourth, analysis must be done on a larger population with diverse biological traits to make the ‘generalized’ model more universally applicable. Fifth, rather than just carrying out binary classification (stress or no stress), stress response could be calibrated at a finer granularity by classifying it into multiple levels through multi-class classification. Finally, in addition to stress, a similar methodology can be followed to detect and mitigate other psychological conditions.

**YSUY: Your Smartphone Understands You – Using Machine Learning to Address Fundamental Human Needs**

Possible future research directions for YSUY are as follows. First, the states of close associates of the users or environmental factors, such as weather, air quality, water quality, etc., can be integrated into the YSUY models. Second, further analyses of a larger more diverse population needs to be performed to broaden YSUY’s applicability. For physical and mental state analyses, we collected data from seven individuals, and for the emotional state analysis, from 10 individuals. They were either undergraduate or graduate students at Princeton University. However, the applicability of YSUY is not limited to a particular age group, gender, ethnicity, or health condition. Third, the effect of YSUY’s state reports on the user’s actions and emotions can be used to update the current YSUY model and implement a suggestion mechanism. Once YSUY determines the user state, the user can take advantage of a biofeedback mechanism, whose effectiveness has been verified earlier [251], [252]. However, in the future, without waiting for the user to take action, YSUY can offer
multiple personalized suggestions, depending on the current state of the user. Finally, the number of classes in physical, mental, and emotional states can be increased to cover a larger set of conditions.

**Smart, Secure, yet Energy-efficient, Internet-of-Things Sensors**

In the future, the applicability of the proposed IoT sensor architecture can be expected to be reinforced with analyses of more datasets for diverse IoT applications. Second, energy modeling can be extended to more machine learning algorithms and short-range data transmission protocols to further buttress the versatility of the architecture. Third, matrix representations for other feature extraction algorithms would help generalize Methods I and II further. Fourth, it needs to be shown how existing (legacy) IoT sensors can be augmented with another device based on our architecture in order to reap some of the benefits of the latter. However, in this case, the energy reduction cannot be expected to be that dramatic. Fifth, a scalability analysis of our architecture needs to be performed for IoT sensor platforms that accommodate a varying number of sensors targeted at a given application or sets of applications. Finally, detailed IC implementations of our sensor architecture are needed.
Bibliography


[99] N. Burns, “Cardiovascular physiology,” School of Medicine, Trinity College, Dublin, 2013.


“These smart TVs were apparently spying on their owners,” [https://www.washingtonpost.com/news/the-switch/wp/2017/02/06/these-smart-tvs-were-apparently-spying-on-their-owners/?utm_term=.62239da4016a](https://www.washingtonpost.com/news/the-switch/wp/2017/02/06/these-smart-tvs-were-apparently-spying-on-their-owners/?utm_term=.62239da4016a), accessed: 08-29-2017.


195


