Semi-Supervised Learning and Data Augmentation for Wearable-based Health Monitoring System in the Wild

Anonymous Author(s)
Affiliation
Address
email

Abstract

Physiological and behavioral data collected from wearable or mobile sensors have been used to detect human health conditions. Sometimes the health-related annotation relies on self-reported surveys during the study, thus a limited amount of labeled data can be an obstacle in developing accurate and generalized predicting models. On the other hand, the sensors can continuously capture signals without labels. This work investigates leveraging unlabeled wearable sensor data for health condition detection. We first applied data augmentation techniques to increase the amount of training data by adding noise to the original physiological and behavioral sensor data and improving the robustness of supervised stress detection models. Second, to leverage the information learned from unlabeled samples, we pre-trained the supervised model structure using an auto-encoder and actively selected unlabeled sequences to filter noisy data. Then, we combined data augmentation techniques with consistency regularization, which enforces the consistency of prediction output based on augmented and original unlabeled data. We validated these methods in sensor-based in wild stress detection tasks using 3 wearable/mobile sensor datasets collected in the wild. Our results showed that the proposed methods improved stress classification performance by 5.3% to 13.8%, compared to the baseline supervised learning models. In addition, our method showed competitive performances compared to state-of-the-art semi-supervised learning methods in the literature.

1 Introduction

Physiological and behavioral sensors have played essential roles in helping measure and improve human health conditions. Researchers developed deep time-series learning methods for various health applications using multimodal sensor data. For example, Swapna *et al.* applied convolutional neural network (CNN) and long short-term memory network (LSTM) to diagnose the diabetes using heart rate signal [24]; Michelli *et al.* designed a cascaded LSTM structure to help detection the human sleep stages based on electroencephalogram (EEG) signals [14]. Moreover, in terms of mental health, Umematsu *et al.* leveraged LSTM to forecasting the daily stress levels of college students [28].

Although researchers achieved promising accomplishments in health-related works using time-series data, the number of annotations limits the model performance in applications. In the health-centered datasets, ground truth labels are usually based on experts (health professionals)' annotations or patients' self-reports. On the other hand, sensors can continuously collect millions of data samples throughout the study. However, even without labels, we cannot ignore the value of information in the collected data. By leveraging these unlabeled sequential data, semi-supervised deep learning has been proven to contribute to applications such as cardiovascular risk detection [1] with an LSTM auto-encoder-based pre-training method originally proposed in [6]. Nevertheless, one drawback of

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

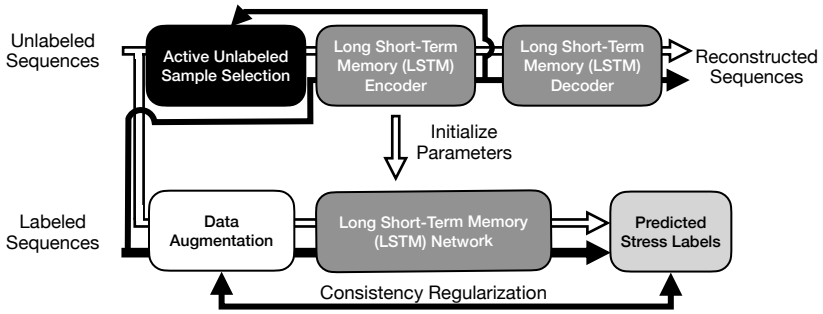

Figure 1: The overall structure of the designed semi-supervised sequence learning framework for stress estimation. The framework contains the components of data augmentation, LSTM auto-encoder pre-training with active unlabeled sample selection, and consistency regularization.

the approach above with wearable data can be that they use all of the unlabeled data. Considering the distribution shifts and noisy samples in human sensing data, leveraging all the unlabeled samples collected by sensors can be unnecessary, especially for these data collected in the wild.

Thus, in this work, we propose novel hinges for effectively learning from unlabeled samples by designing an active sampling approach to effectively select unlabeled data based on the distributions of the labeled samples. Besides learning representations from the unlabeled data, our framework is also integrated with a consistency regularization with data augmentation techniques for time-series data to help the robustness of the model. To evaluate the performances of the proposed methods, we tested the proposed methods using three mental health-related datasets for momentary stress detection in the wild. We observed the improvements in model performances using the proposed methods. The evaluation results also showed competitive performances compared to the reproduced state-of-the-art baseline methods.

2 Methods

This section introduces our proposed semi-supervised learning method for leveraging both labeled and unlabeled sequences in stress estimation. Figure 1 shows the overall framework of the designed approach including data augmentation (DA) on physiological and behavioral sequences, unsupervised LSTM auto-encoder pre-training with active sampling, and consistency regularization. We used LSTM as our backbone model to extract temporal learning representation from the input sequences, and the detailed information on our implemented LSTM can be found in Appendix A.1. Also, we developed sequence-centered DA methods inspired by [27]. The implemented DA methods include jitter, scaling, rotation, permutation, time warping, and magnitude warping. We also described the detailed information on DA in Appendix B.1. The following of this section introduces the approaches of semi-supervised sequence learning with active sampling and consistency regularization.

2.1 Semi-Supervised Sequence Learning

Semi-supervised sequence learning, which uses a sequence-to-sequence auto-encoder in learning representations from unlabeled data, has been shown to improve the model performance when there is a large amount of unlabeled training data[6]. Inspired by the previous work, we construct a sequence-to-sequence LSTM auto-encoder (LSTM-AE). The input X of LSTM-AE is the time-series wearable feature sequence, then the output of the decoder returned the reconstruct sequences $\hat{X}$. The loss function is the mean square loss between the original sequence X and the reconstructed output $\hat{X}$:

$$L_{AE} = \|X - \hat{X}\|^2 \tag{1}$$

After the LSTM-AE was trained, we use the parameters of the LSTM-AE encoder layers as the initial parameters for the corresponding layers in the supervised architecture.

2.1.1 Active Unlabeled Sample Selection

To reduce the influence of noise and unlabeled samples with distribution shifts on the LSTM-AE pre-trained parameters of the model, we propose an active unlabeled sample selection method. We first train an LSTM-AE with labeled data only, then cluster all labeled samples in latent space low-dimension representation using a Gaussian mixture model (GMM). After analyzing the elbow points of both the Akaike and Bayesian information criterion, we fix the number of Gaussian components as K. Then, we use the trained encoder to infer the latent representations of all the unlabeled samples as $h(x_u)$. The negative log-likelihood (NLL) of each unlabeled samples, which is the probability of the observed data under the trained GMM model, can be calculated via:

$$\ell(\mathbf{x_u}) = -\log\left(\sum_{m=1}^{K} \alpha_m \phi(\mathbf{h}(\mathbf{x_u})|\mu_m, \Sigma_m)\right) \tag{2}$$

where α represents the weight mixture component, μ and Σ are the learned mean value and co-variance of the corresponding Gaussian component. Then, we select the unlabeled samples based on the calculated NLL values. The smaller the NLL, the more similar the sample distributes as labeled data. Under this scenario, the pre-trained model can focus on the information learned from samples with a similar distribution as the labeled data.

2.2 Consistency Regularization (CR)

Inspired by [32], we conduct consistency training combined with the augmented data. We generate M new augmented sequences using each labeled/unlabeled time window. If there are any differences between the LSTM model outputs based on the original labeled data and their augmented input data, the consistency losses will be regularized to the supervised loss function. For example, since our task is to estimate stress status in binary classification, the supervised loss will be a cross-entropy loss. We apply the Kullback-Leibler divergence loss as our designed consistency loss. To present the method in a formula, the final loss function with the consistency regularization method is:

$$L = L_{CE}(X_l, y) + \frac{1}{M}\sum_{m=1}^{M}[\alpha \cdot L_{KL}(p(y_l|X_l), p(y_l|\bar{X_l^m})) + \lambda \cdot L_{KL}(p(y_u|X_u), p(y_u|\bar{X_u}^m))] \tag{3}$$

where X_l and y represent the labeled data and ground truth labels, X_u is the unlabeled sequence, $\bar{X_l}$ is the augmented labeled sequence, and $\bar{X_u}$ is the augmented unlabeled sequence. y_l and y_u represent the output from the model using the given input data sequence X_l and X_u, respectively. In our case of classification, $p(y|x)$ is the sigmoid outputs for binary classification. α and λ control the weights of the consistency regularization.

3 Evaluation

We conducted experiments on 3 datasets, including the SMILE [21], the TILES [16], and the CrossCheck [7] datasets. The detailed information on the dataset description and hyper-parameter settings can be found in Appendix C. We conducted experiments in a participant-independent setting, where we conducted a 5-fold cross-validation for each dataset by splitting data from 80% of the participants as training sets and the rest as validation sets. For example, in the SMILE dataset with 45 participants, we selected data from 9 participants as a validation set for each cross-validation.

3.1 LSTM-AE Pre-Training with Active Sampling

To test the effectiveness of the proposed active sampling method for semi-supervised learning, we controlled using different volumes of unlabeled samples to pre-train the model. We compared the model performances based on samples that were selected using the active sampling method with different NLL thresholds versus the same volumes of randomly sampled data points.

We found that the f1 scores of stress prediction models with active sampling as well as random sampling pre-training conditions showed an increasing trend versus the growing amount of pre-training unlabeled data. The performance of active sampling converged with fewer pre-trained

Table 1: Model Performances of 10-fold cross-validation using different methods (macro f1 score). DA: data augmentation, LSTM-AE: LSTM auto-encoder in pretraining, CR: consistency regularization.

	SMILE	TILES	CrossCheck
Baseline: Random	0.46 ± 0.01	0.47 ± 0.01	0.48 ± 0.01
Baseline: LSTM	0.53 ± 0.03	0.51 ± 0.05	0.57 ± 0.04
LSTM + DA	0.56 ± 0.04	0.52 ± 0.03	0.56 ± 0.03
Π-model [13]	0.57 ± 0.02	0.53 ± 0.03	0.56 ± 0.03
VAT [15]	0.58 ± 0.04	0.56 ± 0.02	0.58 ± 0.03
ICT [29]	0.56 ± 0.02	0.55 ± 0.03	0.54 ± 0.04
MixMatch [2]	$\mathbf{0.59 \pm 0.03}$	0.56 ± 0.03	0.52 ± 0.05
DA + LSTM-AE	0.57 ± 0.03	0.55 ± 0.03	$\mathbf{0.59 \pm 0.02}$
DA + CR	$\mathbf{0.59 \pm 0.03}$	0.56 ± 0.04	0.57 ± 0.03
DA + LSTM-AE + CR	$\mathbf{0.59 \pm 0.02}$	$\mathbf{0.58 \pm 0.03}$	$\mathbf{0.60 \pm 0.03}$

samples relative to random sampling. For example, with an NLL score of 10^{-1} - which sampled 30.2%, 34.7% and 28.4% of the unlabeled sequences from the SMILE, TILES, and CrossCheck datasets, respectively - the performances of using semi-supervised learning with actively sampled data were all significantly higher than the performances with randomly sampled data.

3.2 Semi-Supervised Learning

To evaluate the respective contributions of the proposed method, we conducted experiments of evaluating the performances of the supervised LSTM and the proposed methods. We also compared the performances of the random baseline, which assigns labels to test instances according to the class probabilities in the training set [3]. We also compared our proposed method with four state-of-the-art (SOTA) methods including Π-model [13], virtual adversarial training (VAT) [15], interpolation consistency training (ICT) [29] and MixMatch [2]. The reproduction details can be found in Appendix A.

Table 1 shows the model performance comparison using different semi-supervised learning methods. The baseline LSTM method on all 3 datasets outperformed the baseline (random) on the test set (paired t-test, $p < 0.01$). On the SMILE and TILES datasets, the model with DA showed statistically significantly higher f1 scores than the baseline models (ANOVA, Tukey, $P < 0.01$). In contrast, we did not observe significant improvement of applying DA on the CrossCheck dataset. CR improved the model performances on the SMILE and TILES dataset (paired t-test, $P < 0.01$); whereas the statistical test did not verify the performance improvement via CR on the CrossCheck dataset. On all the 3 datasets, the combination of DA, LSTM-AE, and CR showed the best performance (ANOVA, Tukey, $P < 0.05$). When comparing our method with SOTA methods, we observed that our method (LSTM-AE + CR) outperformed all the reproduced SOTA methods on the TILES and CrossCheck datasets (ANOVA, Tukey, $P < 0.05$). On the SMILE dataset, the reproduced MixMatch method performed equivalently with our LSTM-AE and LSTM-AE + CR methods, and showed statistically significant differences with the baseline and DA methods (ANOVA, Tukey, $P < 0.05$).

4 Conclusion

In this work, we proposed a semi-supervised learning framework - which contained components of DA, LSTM-AE pretraining with active sampling, and CR - to help human stress estimation by leveraging massive unlabeled physiological and behavioral data collected in wild. We evaluated our proposed methods using 3 datasets with a small amount of labeled data but a large amount of unlabeled data. We demonstrated that our proposed active sampling approach for LSTM-AE pretraining outperformed the random sampling method and helped the model achieve better performances with less unlabeled samples. Furthermore, our results showed that the combination of DA, LSTM-AE pretraining, and CR provided the best results in f1 scores. In the future, we will continue developing our method and apply it to a broad range of health-related applications.

References

[1] Brandon Ballinger, Johnson Hsieh, Avesh Singh, Nimit Sohoni, Jack Wang, Geoffrey H Tison, Gregory M Marcus, Jose M Sanchez, Carol Maguire, Jeffrey E Olgin, et al. Deepheart: semi-supervised sequence learning for cardiovascular risk prediction. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[2] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[3] Christopher M Bishop. Pattern recognition. *Machine learning*, 128(9), 2006.

[4] Robin Champseix. Heart rate variability analysis. `https://github.com/Aura-healthcare/hrv-analysis`, 2018.

[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[6] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pages 3079–3087, 2015.

[7] Amr Gaballah, Abhishek Tiwari, Shrikanth Narayanan, and Tiago H Falk. Context-aware speech stress detection in hospital workers using bi-lstm classifiers. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8348–8352. IEEE, 2021.

[8] Denise Harrison, Suzanne Boyce, Peter Loughnan, Peter Dargaville, Hanne Storm, and Linda Johnston. Skin conductance as a measure of pain and stress in hospitalised infants. *Early human development*, 82(9):603–608, 2006.

[9] Katherine A Herborn, James L Graves, Paul Jerem, Neil P Evans, Ruedi Nager, Dominic J McCafferty, and Dorothy EF McKeegan. Skin temperature reveals the intensity of acute stress. *Physiology & behavior*, 152:225–230, 2015.

[10] Sepp Hochreiter and Schmidhuber Jurgen. *Long short term memory*. Inst. fur Informatik, 1995.

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456. JMLR.org, 2015.

[12] Hye-Geum Kim, Eun-Jin Cheon, Dai-Seg Bai, Young Hwan Lee, and Bon-Hoon Koo. Stress and heart rate variability: a meta-analysis and review of the literature. *Psychiatry investigation*, 15(3):235, 2018.

[13] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[14] Nicola Michielli, U Rajendra Acharya, and Filippo Molinari. Cascaded lstm recurrent neural network for automated sleep stage classification using single-channel eeg signals. *Computers in biology and medicine*, 106:71–81, 2019.

[15] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

[16] Karel Mundnich, Brandon M Booth, Michelle l'Hommedieu, Tiantian Feng, Benjamin Girault, Justin L'hommedieu, Mackenzie Wildman, Sophia Skaaden, Amrutha Nadarajan, Jennifer L Villatte, et al. Tiles-2018, a longitudinal physiologic and behavioral data set of hospital workers. *Scientific Data*, 7(1):1–26, 2020.

[17] Arthur Pimentel, Abhishek Tiwari, and Tiago H Falk. Human mental state monitoring in the wild: Are we better off with deeperneural networks or improved input features? *CMBES Proceedings*, 44, 2021.

[18] Akane Sano, Sara Taylor, Andrew W McHill, Andrew JK Phillips, Laura K Barger, Elizabeth Klerman, Rosalind Picard, et al. Identifying objective physiological markers and modifiable behaviors for self-reported stress and mental health status using wearable sensors and mobile phones: observational study. *Journal of medical Internet research*, 20(6):e9410, 2018.

[19] Sreelekshmy Selvin, R Vinayakumar, EA Gopalakrishnan, Vijay Krishna Menon, and KP Soman. Stock price prediction using lstm, rnn and cnn-sliding window model. In *2017 international conference on advances in computing, communications and informatics (icacci)*, pages 1643–1647. IEEE, 2017.

[20] Nandita Sharma and Tom Gedeon. Objective measures, sensors and computational techniques for stress recognition and classification: A survey. *Computer methods and programs in biomedicine*, 108(3):1287–1301, 2012.

[21] Elena Smets. Towards large-scale physiological stress detection in an ambulant environment. 2018.

[22] Dimitris Spathis, Sandra Servia-Rodríguez, Katayoun Farrahi, Cecilia Mascolo, and Jason Rentfrow. Sequence multi-task learning to forecast mental wellbeing from sparse self-reported data. 05 2019.

[23] Matthew A Stults-Kolehmainen and Rajita Sinha. The effects of stress on physical activity and exercise. *Sports medicine*, 44(1):81–121, 2014.

[24] Goutham Swapna, Soman Kp, and Ravi Vinayakumar. Automated detection of diabetes using cnn and cnn-lstm network and heart rate signals. *Procedia computer science*, 132:1253–1262, 2018.

[25] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

[26] Vincent W-S Tseng, Akane Sano, Dror Ben-Zeev, Rachel Brian, Andrew T Campbell, Marta Hauser, John M Kane, Emily A Scherer, Rui Wang, Weichen Wang, et al. Using behavioral rhythms and multi-task learning to predict fine-grained symptoms of schizophrenia. *Scientific reports*, 10(1):1–17, 2020.

[27] Terry T Um, Franz MJ Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, pages 216–220, 2017.

[28] Terumi Umematsu, Akane Sano, Sara Taylor, and Rosalind W Picard. Improving students' daily life stress forecasting using lstm neural networks. In *2019 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, pages 1–4. IEEE, 2019.

[29] Vikas Verma, Kenji Kawaguchi, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. *arXiv preprint arXiv:1903.03825*, 2019.

[30] Rui Wang, Min SH Aung, Saeed Abdullah, Rachel Brian, Andrew T Campbell, Tanzeem Choudhury, Marta Hauser, John Kane, Michael Merrill, Emily A Scherer, et al. Crosscheck: toward passive sensing and detection of mental health changes in people with schizophrenia. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 886–897, 2016.

[31] Rui Wang, Weichen Wang, Min SH Aung, Dror Ben-Zeev, Rachel Brian, Andrew T Campbell, Tanzeem Choudhury, Marta Hauser, John Kane, Emily A Scherer, et al. Predicting symptom trajectories of schizophrenia using mobile sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):1–24, 2017.

[32] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. 2019.

[33] Han Yu, Elizabeth B. Klerman, Rosalind W. Picard, and Akane Sano. Personalized wellbeing prediction using behavioral,physiological and weather data. *IEEE International Conference on Biomedical and Health Informatics*, 2019.

A Implemented Models

A.1 Baseline Model: Long Short-Time Memory (LSTM) network

Long short-term memory (LSTM) networks [10], as an extended type of recurrent neural network has been used in time-series applications [19, 5]. In some previous studies, LSTM models provided promising results in stress regression with time-series sensor data[33, 22].

Considering that people's current stress status might be affected by previous short-term physiology or behavior changes, we applied each participant's previous time steps of the data to a multi-layer LSTM for sequential learning. Moreover, we found the distributions of the training data might vary among participants. For example, different participants might have different average heart rates, which introduced the internal covariate shifts among samples. Thus, we also applied a batch normalization (BN) layer[11] after LSTM so that the high-level temporal features extracted by LSTM would be scaled and shifted into the same distribution.

A.2 Π-model [13]:

The Π-model operates two different transformation for an unlabeled input x_u, to form x'_u and x''_u so that the model predicts y'_u and y''_u. Then the model constrains the consistency of the two results. We implemented the Π-model with two different DA approaches in section B.1 randomly to form different input data transformations for each training sample. The mean squared error was used as the consistency loss.

A.3 VAT [15]:

This algorithm constrains the consistency of a signal and its transformation with additive noise, the trainable adversarial perturbation r. The perturbation r is trainable, which was constrained by coefficient $\xi = 1 \times 10^{-6}$ to avoid gradients explosion in our implementation. We allowed 5 iterations for each sample in a single epoch to update the parameter of r.

A.4 ICT [29]:

The ICT algorithm used the mixup method, which summed the original unlabeled data to generate the augmented samples. In our implementation, the mix up coefficient was set as 0.2, which means we summed up $0.8 \cdot x_u^1$ and $0.2 \cdot x_u^2$ as a new signal as x'_u. Then, the model optimized the discrepancy between the prediction y'_u and $\{0.8 \cdot y_u^1 + 0.2 \cdot y_u^2\}$. Also, we reproduced the average teacher strategy [25] with an updating factor of 0.999.

A.5 MixMatch [2]:

The MixMatch approach combined multiple prior techniques, such as consistency regularization, entropy minimization and mixup DA approach, to serve as a semi-supervised learning framework. Similarly to ICT, we also reproduced the mixup approach in the MixMatch algorithm with a mixup coefficient of 0.2. In the steps of sharpening prediction and reducing model entropy, we set the averaging bag size as 3 for each sample with a normalizing temperature of 0.5. The ramping up epoch in equation (**??**) was set to 30 with a coefficient c of 100.

B Data Augmentation Methods

B.1 Data Augmentation

Training a deep neural network model requires a large amount of data for learning robust model parameters; however, obtaining labels is expensive and we usually have access to only limited labeled datasets. Data augmentation (DA) synthesizes a comprehensive set of possible data points and could alleviate the limited data problems. Further, DA approaches can generate transformed samples even for unlabeled data, which is the foundation of the consistency regularization method we introduce in section 2.2.

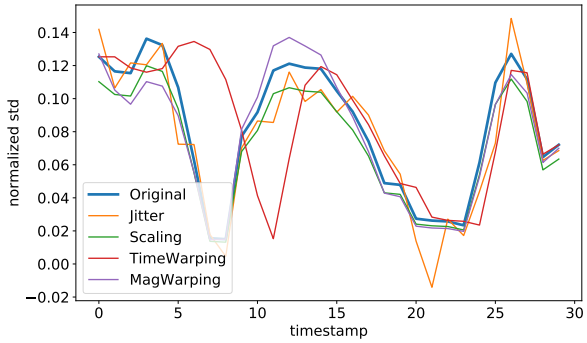

Figure 2: Examples of data augmentation on a sequence of ACC std signal

Table 2: Meta information about datasets used for evaluation

Dataset	SMILE	TILES	CrossCheck
# of Labeled Data	2494	1229	5914
# of Classes	2	2	2
Used Modality	ECG, GSR, ACC, ST	ECG	Smartphone
# of Participants	45	212	75

We extended four DA methods from [27] to physiological and behavioral sequences. These four methods are jittering, scaling, time warping, and magnitude warping. Jittering (J) added tiny Gaussian noise to the original signals. For scaling (S), the original signals are scaled by generated Gaussian random numbers ($\mathcal{N} \sim (1, 0.05)$). Time warping (TW) is a way to perturb the temporal characteristics of the data. The temporal locations of the samples are changed by smoothly distorting the time intervals between samples. Magnitude warping (MW) changes the magnitude of each sample by convoluting the data window with a smooth curve varying around one with a standard deviation of 0.05 ($\mathcal{N} \sim (1, 0.05)$). The essence of these methods is adding a small amount of noise to time-series data so that the trained model will be robust. Figure 2 shows an example of different DA methods on a 30-minute sequence of accelerometer standard deviation data. The blue line is the original signal, and the other lines represent the data generated using four different DA methods. We applied the combination of all these methods in our experiments.

C Experimental Settings

C.1 Datasets

In this section, we describe three datasets we used to evaluate our methods. The meta information of the used dataset can be found in table 2.

C.1.1 Dataset I: SMILE

Wearable sensor and self-report data were collected from 45 healthy participants (39 females and 6 males), in total for 390 days. The average age of participants was 24.5 years old, with a standard deviation of 3.0 years. Participants contributed to an average of 8.7 days of data, with a minimum of 5 days and a maximum of 9 days.

Two types of wearable sensors were used for data collection [21]. One was a wrist-worn device (Chillband, IMEC, Belgium) designed for the measurement of skin conductance (SC), skin temperature (ST), and acceleration data (ACC). The SC was sampled at 256 Hz, ST at 1 Hz, and ACC at 32 Hz. Participants wore the sensor for the entire testing period, but could take it off during the night and while taking a shower or during vigorous activities. The second sensor was a chest patch (Health Patch, IMEC, Belgium) to measure ECG and ACC. It contains a sensor node designed to monitor

ECG at 256 Hz and ACC at 32 Hz continuously throughout the study period. Participants could remove the patch while showering or before doing intense exercises.

Stress Labels: In addition to the physiological data collected by sensors, participants received notifications on their mobile phones to report their momentary stress levels 10 times per day, spaced out roughly 90 minutes apart for eight consecutive days. In total, 2494 stress labels were collected across all participants (80% compliance). The stress scale ranged from 1 ("not at all") to 7 ("Extreme"). In 45% of the cases, participants reported that they were not under stress, while in only 2% of the cases did they report that they were under extreme stress. In this work, we binarized the stress levels by categorizing stress level 1 as a class of "non-stressed" (45%) and level 2-7 as the "stressed" class (55%).

C.1.2 Dataset II: TILES

Tracking Individual Performance with Sensors (TILES) is a multi-modal data set for the analysis of stress, task performance, behavior, and other factors pertaining to professionals engaged in a high-stress workplace environment [16]. The dataset was collected from 212 participants for 10 weeks. In this work, we leveraged the ECG signals, which were not collected in a strictly continuous manner. At 5-minute intervals, the sensor collects ECG signals for fifteen seconds at a sampling rate of 250 Hz for the participants. We extracted features using the ECG signals and estimated the self-reported stress levels.

Gaballah *et al.* leveraged TILES audio and physiological data with a bidirectional LSTM network and inferred stress labels in a binary classification task[7]. They achieved a f1-score of 0.64. With extracted features from ECG signals, Pimentel *et al.* proposed SVM based binary stress detection models with an f1-score of 0.68 [17].

Stress Labels: Participants annotated stress levels through multiple 5-point scale questions. Following the stress label processing procedures in [7], We calculated the z-scores of stress levels for each individual, considering the subjective variability and then divided them into two classes, class 0 (non-stressed, z-score below the average) and class 1 (stressed, z-score above the average). Overall, 600 stressed labels and 629 non-stressed labels were processed.

C.1.3 Dataset III: CrossCheck

We also used the dataset from the CrossCheck project [30], where smartphone data were collected in a clinical trial from 75 patients with schizophrenia for up to a year. The collected phone data include acceleration, light levels, sound levels, GPS, and call/SMS meta data. In addition, the participants filled out the momentary ecological assessment (EMA) up to three times a week to assess their symptoms.

Prior work estimated schizophrenia symptoms including depression, harm, stress, etc using machine learning regression models and behavioral features extracted from the phone data. [30]. The best mean absolute error performance from the non-personalized model was 1.5 out of 0-3 scale. Similar symptom estimation performance was also reported in [26, 31].

Stress Labels: Stress labels were collected via EMA. Participants reported their stress levels using a 4-point scale, where "0" means no stress at all, whereas "3" means extremely stressed. The total number of stress labels was 5914, where 49% was "0" stress label class, 28% was "1", 16% was "2", and 7% was "3". We evaluated the proposed methods in a binary classification task on "0" (non-stressed) versus "1, 2, 3" (stressed).

C.1.4 Features Extraction

Considering categorical event data, such as incoming/outgoing phone call and SMS, were included in the datasets, we decided to extract hand-crafted features, not using deep learning methods such as CNN to extract deep features from raw data. We used features extracted from the period prior to the stress label to develop stress detection models and learn the temporal representations. In the three datasets mentioned in section C.1, ECG (SMILE, TILES), SC (SMILE), ST (SMILE), ACC (SMILE) and smartphone logs (CrossCheck) were used in estimating stress level. All these signals contribute to infer human stress. For example, ECG reflects sympathetic and parasympathetic activity, which has been proven related to stress [12, 20]. Stress has also been proven related to conductance

in human eccrine sweat glands, which can be captured by sensors as SC [8]. ST changes could reveal the intensity of stress [9]. ACC is associated with human physical activity, which has been shown influenced by stress level [23]. Also, previous studies have shown that smartphone usage data contributed to stress prediction [18, 28].

We calculated statistical features from ST and ACC data, such as mean and standard deviation values across periods. For other sensor measurement, we introduce the features extracted from ECG, SC, and smartphone data in this section.

ECG Features: To extract features, based on the data sampling pattern of sensors in different studies, we used 15 seconds of high-resolution ECG data every 5 minutes for the TILES dataset and continuous ECG data for the SMILE dataset. We extracted both time and frequency domain ECG features using a Python library [4]. Also, the outlier removal was performed using the same library with 300 and 2000 as the low and high bound of R-R interval values, respectively. The engineered time-domain features covered subjects' heart rate, heart variability, etc. The frequency-domain features contained the power spectrum information of heart activities in various frequency bands. These features has been proven associated with human stress levels [12, 20]. The detailed list of ECG features is available in Appendix C.1.5.

SC Features: The raw SC signal was cleaned with a Elliptic filter with a order of 4, maximum pass-band ripple of 0.1, and minimum stop-band attenuation of 40 [21]. We computed SC magnitude, the number of SC responses, the response duration, etc, following previous stress studies [12]. See the detailed list of SC features in Appendix C.1.6.

Smartphone Features: We processed the CrossCheck data and extracted features using data collected by smartphones. These features include acceleration intensity, phone application usage, call/sms counts and duration, and the location related features from GPS. See the detailed list of smartphone features in Appendix C.1.7.

C.1.5 ECG features

Time-domain features

- mean_nni: The mean of RR-intervals (time interval from one ECG peak to the next peak).
- sdnn : The standard deviation of the time interval between successive normal heartbeats (i.e., the RR-intervals).
- sdsd: The standard deviation of differences between adjacent RR-intervals
- rmssd: The square root of the mean of the sum of the squares of differences between adjacent NN-intervals. Reflects high frequency (fast or parasympathetic) influences on HRV (i.e., those influencing larger changes from one beat to the next).
- median_nni: Median Absolute values of the successive differences between the RR-intervals.
- nni_50: Number of interval differences of successive RR-intervals greater than 50 ms.
- pnni_50: The proportion derived by dividing nni_50 (The number of interval differences of successive RR-intervals greater than 50 ms) by the total number of RR-intervals.
- nni_20: Number of interval differences of successive RR-intervals greater than 20 ms.
- pnni_20: The proportion derived by dividing nni_20 (The number of interval differences of successive RR-intervals greater than 20 ms) by the total number of RR-intervals.
- range_nni: the difference between the maximum and minimum nn_interval.
- cvsd: Coefficient of variation of successive differences equal to the rmssd divided by mean_nni.
- cvnni: Coefficient of variation equal to the ratio of sdnn divided by mean_nni.
- mean_hr: The mean Heart Rate.
- max_hr: Max heart rate.
- min_hr: Min heart rate.
- std_hr: Standard deviation of heart rate.

Frequency-domain features:

- total_power : Total power density spectral
- vlf : variance (= power) in HRV in the Very low Frequency (.003 to .04 Hz by default). Reflect an intrinsic rhythm produced by the heart which is modulated primarily by sympathetic activity.
- lf : variance (= power) in HRV in the Low Frequency (.04 to .15 Hz). Reflects a mixture of sympathetic and parasympathetic activity, but in long-term recordings, it reflects sympathetic activity and can be reduced by the beta-adrenergic antagonist propranolol.
- hf: variance (= power) in HRV in the High Frequency (.15 to .40 Hz by default). Reflects fast changes in beat-to-beat variability due to parasympathetic (vagal) activity. Sometimes called the respiratory band because it corresponds to HRV changes related to the respiratory cycle and can be increased by slow, deep breathing (about 6 or 7 breaths per minute) and decreased by anticholinergic drugs or vagal blockade.
- lf_hf_ratio : lf/hf ratio is sometimes used by some investigators as a quantitative mirror of the sympathy/vagal balance.
- lfnu : normalized LF power.
- hfnu : normalized HF power.

C.1.6 SC Features

- skin conductance (SC) level: average SC value.
- phasic SC : signal power of the phasic SC signal (0.16-2.1 Hz).
- SC response rate : number of SC responses in window divided by the totally length of the window (i.e. responses per second)
- SC second difference : signal power in second difference from the SC signal
- SC response : number of SC responses
- SC magnitude : the sum of the magnitudes of SC responses
- SC duration : the time duration of SC responses
- SC area : the sum of the area of SC responses in seconds

C.1.7 Smartphone Features

- accel_mean: mean value of 3-axis acceleration data.
- appall : number of APPs used.
- app[com, entertain, product, social, fit] : number of APPs in category of [communication, entertainment, production, social, fitness & health] used
- call_log_count_type[1,2] : number of outgoing-call (1) and incoming-call (2)
- call_log_sum_type[1,2] : total duration (in second) of outgoing-call (1) and incoming-call (2)
- conversation_sum : total duration (in second) of conversation captured by phone microphone
- distances_sum : total distance of movement captured by GPS location data
- screen_sum : total duration (in second) of screen usage
- sms_log_count_type[1,2] : number of outgoing-sms (1) and incoming-sms (2)

C.2 Model Hyper-parameters

For hyperparameters of the baseline LSTM model, we used grid searching through cross-validation. We adopted the following structure and parameters:

- **SMILE**: 3 layers of LSTM with 64 recurrent units were connected, incorporating 0.4 recurrent dropout and 0.3 dropout rates in each LSTM layer. After a BN layer, a fully-connected layer followed with 512 hidden units with a dropout rate of 0.5. We chose Adam as the optimizer with a learning rate of 0.0001.

- **TILES & CrossCheck**: 3 layers of LSTM with 32 recurrent units were connected, incorporating 0.3 dropout rates in each LSTM layer. After a BN layer, a fully-connected layer followed with 256 hidden units with a dropout rate of 0.5. We chose Adam as the optimizer with a learning rate of 0.00005.