EasyBreathe: Data Analysis and Stress Recognition

Mark Zolotas

***Abstract*—To provide therapeutic stress-relief treatment for youthful adults, EasyBreathe must incorporate accurate real-time stress recognition into its platform. In this report, I outline the design requirements and challenges for this learning component, and investigate relevant work in supervised stress classification. Finally, I present both my preliminary findings after implementation and planned future developments.**

I. Introduction

A fundamental characteristic of EasyBreathe is timely detection of elevated psychological stress levels in young adults. In order to achieve this target, our mobile platform aims to incorporate a stress recognition module into its system design that is capable of identifying persistent peaks of distress in users based on incoming wearable sensor data. There is currently a wide selection of available machine learning (ML) algorithms for performing stress recognition, which is often expressed in the form of a classification problem. By employing these ML techniques, the learning component of our system intends to distinguish between user states of relaxation or distress and consequently enable the in-built iPhone application to report on the long-term effects of either outcome. Depending on the output of this stress-prediction model, the device holder can also choose to activate therapeutic stress-relief treatments in accordance with our project hypotheses for inciting relaxation.

For a chronic stress-reduction platform such as EasyBreathe, a key design requirement is an operational ML component that will facilitate real-time identification of severe distress in users. Designing a ML model that fits the criteria of automatic stress detection can be broadly subdivided into four stages of development: data acquisition, data pre-processing, model training, and performance assessment. Whilst my colleague is responsible for acquiring input data streams and designing an interface to feed raw signal measurements from wearable sensors into the iPhone, I will construct a learning framework that spans across the latter three phases of development. My role in the implementation of a stress recognition module therefore consists of performing the dimensionality reduction steps over incoming signal data, training a psychological stress classifier, and then evaluating its performance under experimentation.

II. Related Research & Existing Applications

Pre-processing physiological signal data measured directly from wearable sensors is an essential step in ML models for health monitoring systems. This data mining procedure involves processing raw input signals prior to applying a learning algorithm over these data streams. Typical examples of pre-processing operations are data cleaning by filtering out noise present in raw input signals, or interpolating data elements to account for missing values [1]. However, another extended method of pre-processing in ML systems is dimensionality reduction, whereby high-dimensional data is transformed into a representative form of reduced dimension [2]. High dimensionality is a well-known problem for classification algorithms due to expensive computational costs and memory usage [3].

There are two primary approaches to dimensionality reduction: feature extraction and feature selection. Both of these approaches offer numerous advantages for ML models, such as removal of redundant or irrelevant input data, increased classification accuracy [3][4], compact representation of time-series through data transformations, and separation of signal artifacts from noise [5]. In the domain of mobile healthcare, features correspond to identifiable characteristics and predominant variables present in vital signals. Banaee et al. presented a summarised table of commonly used features for every wearable sensor stream in literature [6], which provides a thorough basis for selecting attributes available in physiological signal data.

A variety of feature extraction methods have been applied over wearable sensor streams. In the Pitale et al. [7] paper observing heart rate variability (HRV) classification, feature extraction of electrocardiogram (ECG) signals was performed to reduce feature dimensionality. A combination of principal component analysis (PCA) and wavelet transforms was utilised in order to extract relevant variables from the ECG signal. Preece et al. [8] conducted a comparison between 14 methods of extracting classification features from accelerometer signals. Their findings indicated that fast Fourier transform (FFT) feature sets outperformed wavelet feature sets by obtaining higher classification accuracies in all tested scenarios. Some examples of the frequency-domain feature sets included measurements of spectral energy and magnitudes of the first five components of FFT analysis.

Recent research also proposes a range of feature selection algorithms for sensor data. In particular, the post-traumatic stress disorder (PTSD) research of Karstoft et al. [9][10] on pre-deployed soldiers applied a feature selection algorithm to generate a set of the most predictive variables of resilience and PTS. The Markov Boundary feature selection algorithm for Generalized Local Learning [11] resulted in this output set by identifying univariate associations between variables and outputs of resilience or PTS, while iteratively discarding predictors that displayed non-significant associations.

With regard to stress recognition, there is currently a competitive selection of ML classification algorithms available in research applications similar to EasyBreathe. For instance, a number of studies used support vector machines (SVMs) to classify participants into groups of resilience or distress [9][10][12]. SVMs are a state-of-the-art supervised learning algorithm [13] for classification problems and have been

broadly applied in stress detection by using modified techniques [14][15], or by integrating with other algorithms in order to form a hybrid system [16]. Other classifiers have also been extensively used for elevated stress identification, such as the Naïve Bayes and Decision Tree (DT) classifiers [17]. It is worth noting that the DT classification method is considered to be of low computational complexity [18], therefore making it an ideal candidate for a mobile system.

Many online frameworks exist to train classifiers and determine the best predictive algorithm to apply in the context of stress inference. A common ML engine used in this research domain is the online WEKA data mining software [19]. In the work of Sun et al. [20] for mental stress classification, various supervised learning models were trained through this WEKA engine. Nevertheless, there are other data analytics tools specifically employed for exploring ML models, such as the MATLAB Classification Learning application [21] and the Orange interactive visualisation toolbox [22].

Despite typical performance assessment of classifiers focusing on classification accuracy, overfitting training examples is a widely known cause of misclassification error over unseen data. To avoid mistaking classifiers with high classification rates as generalised predictors of unseen examples, cross-validation is a state-of-the-art method for partitioning sample data into two sets, one for training and the other for validation. Using smartphone applications for stress detection in campus students [23], one cross-validation approach was to leave one student out and use their instances for testing, while the training set consisted of remaining data. A study measuring perceived stress at work through smartphones [24] also employed a leave-one-day-out variant of cross-validation. Both methodologies resulted in relatively low classification rates in the range 40-60%, which potentially indicate more realistic results in the complex problem of stress recognition using wearable devices.

III. Design Requirements & Challenges

The learning component of EasyBreathe is responsible for performing real-time recognition of elevated stress levels in device holders by implementing three essential data-analysis procedures. First of all, the dimensionality reduction stage involves extracting and selecting the most influential statistical qualities or features present within the input streams of stress-related data. Once a feature dataset is obtained, a range of supervised ML classification models can be trained over this pre-processed information to subsequently generate a classifier capable of producing a binary output that distinguishes between emotional states of distress and calm. Finally, various metrics are used to assess the performance of these ML models with respect to classification accuracy and generalisation beyond examples in the training dataset.

Feature extraction for wearable sensor streams primarily aims to discover a set of new characteristics from the raw signal data, which are identically representative of the original data [6][25]. In the context of stress inference, our system will deal with accelerometer and heart rate (HR) vital signals as measured via in-built sensors of the Apple Watch. Given

Wearable Sensors	Features
Heart-Rate *Time-Domain*	Mean and Std. Dev. HR Mean and Std. Dev. RR
Accelerometer *Time-Domain*	Mean and Std. Dev. of XYZ Activity Index Correlation coefficients
Accelerometer *Frequency-Domain*	Spectral Energy over XYZ Average Energy over Cartesian axes Magnitude of first 5 components of FFT analysis

TABLE I: Core features chosen for extraction from the time-domain and frequency-domain properties of the HR and accelerometer signals. Largely influenced by the work of other research applications [20][27].

the challenge of high-dimensionality in data-driven ML tasks, mapping these physiological variables to a smaller dimension is critical for this stress recognition framework due to the computational intractability and worsened performance in operating prediction algorithms over real-time signal data. Moreover, the consequences of high-dimensional data have significant impacts on the power consumption of mobile applications.

Exploiting the dimension-reducing nature of feature extraction is therefore a key design objective for our system's stress detection module. Since stress-related HR and accelerometer data are continuous time-series readings, the features extracted from these input feeds will predominantly be time-series signal properties [26]. However, spectral-domain analysis of accelerometer coordinate signals is also performed in this pre-processing stage. Table I provides a complete list of time-domain and spectral-domain properties scheduled for extraction in our prototypical stress recognition component. The wearable sensor features presented in this table were largely influenced by similar predictive stress applications [20][27]. MATLAB is the designated tool for implementing feature extraction algorithms that perform time-domain and frequency-domain analysis of incoming signal data.

Another fundamental design requirement lies in the selection of features. Attribute selection involves choosing a subset of features from the original set, such that there is an optimal reduction in dimensionality of the feature space according to a certain criterion [26][28][29]. This procedure renders sets of feature vectors that will further narrow the scope and dimension of the initial stress inference problem. A common wrapper-based model of variable selection is the sequential feature selection function, which is available in the Statistics and ML Toolbox of MATLAB [30]. This function minimises a parameterised criterion, such as misclassification rate, over all feasible feature sets by exercising a sequential search in a single direction through the candidate set. The WEKA ML engine [19] also offers wrapper-based attribute selection methods that evaluate different combinations of features against model performance.

Prior to determining an optimal classifier of distressed and relaxed user states, our system requires consideration for its mobile environment. For instance, the real-time processing of raw input signals and subsequent data analysis will potentially necessitate high computational costs, as well as abundant memory storage requirements. It is therefore in our advantage to apply a model of low computational complexity, such as DT classification, and avoid the significant drawbacks of

power consumption. Furthermore, performance metrics of the model described through a confusion matrix will also play a role in the choice of a classification algorithm. While hardware and complexity overheads are crucial concerns for development, performance evaluation of various classification models will initially act as the deciding factor in designing a stress recognition component.

The iPhone component of EasyBreathe will perform the bulk of processing required for our stress classification model using the C++ OpenCV framework. In order to operate the ML libraries contained within OpenCV through our mobile device, a C bridge will need to be set up to interface this C++ package with the iOS development environment programmed using Swift. Once this connection is established between the two software frameworks, our iOS application can translate inward sensor data into the cv::Mat format for matrices and then train classifiers over these signals. As this stress recognition module is taking a supervised learning stance, our predictive algorithm will also require labelled output classes over the training datasets.

Beyond determining a suitable classification model for our stress recognition application, another critical stage in development involves assessing the performance of a classifier and preventing overfitting. The MATLAB Classification Learner App [21] will be applied in an isolated environment over recorded sensor data before training a classifier in our proposed platform. This will enable us to access confusion matrices for multiple classification algorithms and accordingly nominate a predictive algorithm that fulfils high classification rates. Moreover, a well-known challenge for ML models that needs to be accounted for is overfitting training datasets. This may occur in our system if the preferred classifier is trained to perfectly fit the training examples provided, thereby increasing error in its ability to generalise over unseen data. Our implementation will thus require cross-validation in order to reduce the effects of overfitting.

IV. Implementation

My initial approach to obtaining a working prototype of the EasyBreathe ML component involved exploring healthcare datasets and plotting stress-related time-series readings in MATLAB. This led to my discovery of the online UCI ML repository [31], which possesses both accelerometer and HR mobile healthcare datasets for real-world participants. Although these datasets provided me with a greater understanding of the behaviour of these signals in a continuous-time scenario, they presented a couple of major drawbacks. First and foremost, there was a significant lack of datasets for participants undergoing chronic stress, which consequently prevented the use of these .dat files to label training examples for our classification model. Furthermore, these datasets were derived from the daily monitoring of mobile human subjects, and hence possessed missing data points or noisy artefacts in the results. However, the latter fault is likely to occur for most wearable sensor data streams.

After building on the knowledge obtained from visualising the learning signals through online datasets, my next stage in development revolved around feature processing of mock sensor data. Alongside the .m file used to parse online repository datasets and load them into a MATLAB environment as struct arrays, a separate script was programmed to extract time-series properties from these mock physiological signals. Following from analogous research into activity-based stress detection [20] and chronic stress recognition [27], loaded input signals were segmented into data windows of 5-minute intervals during feature extraction. As a result, daily monitoring would ideally require a total of 288 data windows $w_1 \dots w_{288}$ for full coverage. Each window has a corresponding feature vector $f_i(j)$, where i is the window index and j is a stress-related feature. Currently, these features vectors solely contain time-domain analysis of mean HR, RR, and accelerometer signal data, as well as the Activity Index [27]:

$$Activity\ Index = \sum_{i=1}^{N} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2} \quad (1)$$

The activity index presented in Equation 1 is essentially a comparison of Cartesian coordinate values at successive time segment intervals. According to its founding paper, this index represented a core feature of their stress recognition model as it demonstrated how active a test subject was, which is often disregarded from pure HR analysis. My model therefore incorporates this index into its design, as a stress inference algorithm for university students is likely to encounter substantial activity on a daily basis, thus affecting classification results unless taken into consideration.

Although classifying this mock data is on halt due to the lack of accessible labels for training examples, there are still numerous reflections for implementation in future feature processing. First of all, spectral-domain properties of accelerometer data should be computed within the aforementioned MATLAB script. Categorical features related to the time of day, the location of the user, or even the date should also be potentially integrated into this feature extraction program. Furthermore, real data is likely to face frequent inconsistencies and noise. Our pre-processing stage of development may need to consider applying a set of rules to determine which data segments are reliable throughout the day, as conducted in the perceived stress investigation [27]. Finally, the mock sensor datasets should undergo feature selection using the MATLAB Statistics and ML Toolbox method *sequentialfs* [30].

Once sensor data from the Apple Watch is accessible to the iPhone component of EasyBreathe, training examples will be collected from ten students for a period of 24 hours. Our data collection experiment will rely on students inputting their psychological state into the iPhone application whenever they perceive distress. This will hopefully present our model with the training labels it requires to perform supervised classification learning. Additionally, this real data will then be driven into MATLAB's Classification Learner App after undergoing the aforementioned pre-processing steps. The benefit of using MATLAB as a development environment during initial data-analysis stages is twofold. From a data mining perspective, the Statistics toolbox facilitates feature extraction, variable selection, and other ML techniques. While from a performance point of view, harnessing the Classification Learner App will

play a crucial role in comparing classification models in later stages of development.

In parallel to feature processing, my stress recognition component has also begun to include ML models within a Swift program underlying our iOS development project. From previous research into different classification models, my initial approach considered training a classifier with the online WEKA engine [19] and then importing this trained classifier into an iOS environment. However, after brief research into the subject matter, the likelihood of accessing this classifier at runtime without severe communication overhead seemed improbable. Consequently, I began developing a C bridge to interface between the C++ OpenCV framework and our iOS project code. My intentions for future development are therefore to translate all feature processing implementations from MATLAB into the iOS ML program and to implement a training algorithm for our optimally selected classifier. After this step is successfully accomplished, 10-fold cross validation with varying feature combinations will be employed to deal with overfitting.

V. Conclusion

In conclusion, the stress detection component of Easy-Breathe plays a fundamental role in the proposed chronic stress reduction platform. I have outlined multiple sources of influential stress recognition applications in ML and discussed my current contribution to the development of this classification module. Taking advantage of the encountered challenges and research findings, my aim is to complete the necessary functionality required for accurate prediction results in a chronic stress setting.

References

[1] Sow, D., Turaga, D.S. and Schmidt, M., 2013. *Mining of sensor data in healthcare: a survey.* (pp. 459-504). Managing and Mining Sensor Data Springer US.

[2] Van Der Maaten, L., Postma, E. and Van den Herik, J., 2009. *Dimensionality reduction: a comparative.* (pp. 66-71). J Mach Learn Res,10, 109.1.

[3] Janecek, A., Gansterer, W.N., Demel, M. and Ecker, G., 2008.*September. On the Relationship Between Feature Selection and Classification Accuracy.* (pp. 90-105). FSDM.

[4] Ladha, L. and Deepa, T., 2011. *Feature selection methods and algorithms.* (pp.1787-1797), International journal on computer science and engineering,1(3).

[5] Meyer, F.G., 2011.*Signal Data Mining from Wearable Systems.* (pp. 123-146), Wearable Monitoring Systems Springer US.

[6] Banaee, H., Ahmed, M.U. and Loutfi, A., 2013. *Data mining for wearable sensors in health monitoring systems: a review of recent trends and challenges.* (pp.17472-17500) Sensors,13(12).

[7] Pitale, R., Tajane, K. and Umale, D.J., 2014 *Heart Rate Variability Classification and Feature Extraction Using Support Vector Machine and PCA: An Overview.* International Journal of Engineering Research and Applications,4(1).

[8] Preece, S.J., Goulermas, J.Y., Kenney, L.P. and Howard, D., 2009. *A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data.* (pp.871-879) Biomedical Engineering, IEEE Transactions on,56(3).

[9] Karstoft, K.I., Statnikov, A., Andersen, S.B., Madsen, T. and Galatzer-Levy, I.R., 2015. *Early identification of posttraumatic stress following military deployment: Application of machine learning methods to a prospective study of Danish soldiers.*, (pp.170-175) Journal of affective disorders,184.

[10] Galatzer-Levy, I.R., Karstoft, K.I., Statnikov, A. and Shalev, A.Y., 2014. *Quantitative forecasting of PTSD from early trauma responses: A machine learning application.* (pp.68-76) Journal of Psychiatric Research, 59.

[11] Aliferis, C.F., Statnikov, A., Tsamardinos, I., Mani, S. and Koutsoukos, X.D., 2010. *Local causal and markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation.* (pp.171-234) The Journal of Machine Learning Research, 11.

[12] Zhai, J. and Barreto, A., 2006. *Stress detection in computer users based on digital signal processing of noninvasive physiological variables.* Engineering in Medicine and Biology Society. EMBS'06. 28th Annual International Conference of the IEEE.

[13] Boser, B.E., Guyon, I.M. and Vapnik, V.N., 1992. *A training algorithm for optimal margin classifiers.* (pp. 144-152) Proceedings of the fifth annual workshop on Computational learning theory. ACM.

[14] Hernandez, J., Morris, R.R. and Picard, R.W., 2011. *Call center stress recognition with person-specific models.* (pp. 125-134) Affective computing and intelligent interaction. Springer Berlin Heidelberg.

[15] Sano, A. and Picard, R.W., 2013. . *Stress recognition using wearable sensors and mobile phones.* (pp. 671-676) Affective Computing and Intelligent Interaction (ACII), Humaine Association Conference IEEE.

[16] Sharma, N., Dhall, A., Gedeon, T. and Goecke, R., 2014. *Thermal spatio-temporal data for stress recognition.* (pp. 1-12) EURASIP Journal on Image and Video Processing, 2014(1).

[17] Zhai, J. and Barreto, A., 2006. . *Stress Recognition Using Non-invasive Technology.* (pp. 395-401) FLAIRS Conference.

[18] Lim, T.S., Loh, W.Y. and Shih, Y.S, 2000. *A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms.* (pp.203-228) Machine learning,40(3).

[19] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I.H., 2009. *The WEKA data mining software: an update.* (pp.10-18) .ACM SIGKDD explorations newsletter,11(1).

[20] Sun, F.T., Kuo, C., Cheng, H.T., Buthpitiya, S., Collins, P. and Griss, M., 2010. *Activity-aware mental stress detection using physiological sensors.* (pp. 211-230) Mobile computing, applications, and services. Springer Berlin Heidelberg.

[21] Classification Learner App - Statistics and Machine Learning Toolbox. Classification Learner App - MATLAB & Simulink - MathWorks United Kingdom. http://uk.mathworks.com/help/stats/classification-learner-app.html (Accessed 24 February 2016).

[22] Orange Data Mining. http://orange.biolab.si/ (Accessed 24 February 2016).

[23] Gjoreski, M., Gjoreski, H., Lutrek, M. and Gams, M., 2015. *Automatic detection of perceived stress in campus students using smartphones.* (pp. 132-135) Intelligent Environments (IE), 2015 International Conference. IEEE.

[24] Muaremi, A., Arnrich, B. and Trster, G., 2013. *Towards measuring stress with smartphones and wearable devices during workday and sleep.* (pp.172-183) BioNanoScience,3(2).

[25] Motoda, H. and Liu, H., 2002. *Feature selection, extraction and construction.* (pp. 67-72) Communication of IICM (Institute of Information and Computing Machinery, Taiwan) Vol, 5.

[26] Choi, J., Ahmed, B. and Gutierrez-Osuna, R., 2012. *Development and evaluation of an ambulatory stress monitor based on wearable sensors.* (pp. 279-286) Information Technology in Biomedicine, IEEE Transactions on, 16(2).

[27] Wu, M., Cao, H., Nguyen, H.L., Surmacz, K. and Hargrove, C., 2015, August. *Modeling perceived stress via HRV and accelerometer sensor streams.* (pp. 1625-1628) Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE. IEEE.

[28] Blum, A.L. and Langley, P., 1997. *Selection of relevant features and examples in machine learning.* (pp.245-271) Information Artificial intelligence, 97(1).

[29] Dash, M. and Liu, H., 1997 *Feature selection for classification.* (pp.131-156) Information Intelligent data analysis, 1(3).

[30] Feature Selection - Statistics and Machine Learning Toolbox. Feature Selection - MATLAB & Simulink - MathWorks United Kingdom. http://uk.mathworks.com/help/stats/feature-selection.html?requestedDomain=www.mathworks.com (Accessed 24 February 2016).

[31] Asuncion, A & Newman, D.J. (2007). UCI Machine Learning Repository. http://www.ics.uci.edu/~mlearn/MLRepository.html. Irvine, CA: University of California, Department of Information and Computer Science.